2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

An Adaptive Velocity Obstacle Avoidance Algorithm for Autonomous Surface Vehicles

Daniel Filipe Campos[1], Aníbal Matos[2] and Andry Maykol Pinto[3]

Abstract— **This paper presents a new algorithm for a real-time obstacle avoidance for autonomous surface vehicles (ASV) that is capable of undertaking preemptive actions in complex and challenging scenarios. The algorithm is called adaptive velocity obstacle avoidance (*AVOA*) and takes into consideration the kinematic and dynamic constraints of autonomous vessels along with a protective zone concept to determine the safe crossing distance to obstacles. A configuration space that includes both the position and velocity of static or dynamic elements within the field-of-view of the ASV is supporting a particle swarm optimization procedure that minimizes the risk of harm and the deviation towards a predefined course while generating a navigation path with capabilities to prevent potential collisions.**

Extensive experiments demonstrate the ability of *AVOA* to select a velocity estimative for ASVs that originates a smoother, safer and, at least, two times more effective collision-free path when compared to existing techniques.

I. Introduction

The global trade transportation is mainly assured by shipping, rounding up to 90% of the cargo, existing over 50 000 merchant ships trading internationally. As so the safety of the cargo and the crew is a major concern. More than 75% of incidents in the shipping industry are caused by human error and, therefore, the introduction of autonomous vessel technology can be a way of decreasing those losses [1]. However, some of that technology is immature with several challenges to be solved. An example of that challenge is the ability of an autonomous surface vehicle (ASV) to avoid unexpected obstacles that could appear along its course. This can be classified as a hard task when considering dynamic and unstructured environments and so, a motion planning capability to anticipate potentially dangerous situations considering multiple arbitrary obstacles, constrains of weather conditions as well as the kinematic motion of the vehicle can easily become a NP-hard problem [2].

In this context, the research presented in this article proposes a real-time method for avoiding static and dynamic obstacles that is efficient in terms of computational power, maneuverability and travel time. The method is called adaptive velocity obstacle avoidance (*AVOA*) and estimates a collision-free velocity model based on a protective zone concept that represents the scenario by combining the position and velocity information for all obstacles that are perceived by the vehicle using multi-target tracking methodologies, either with resource to computer vision, radar or both. No external communication between the agent and the environment is required. The safe velocity is estimated through a Particle Swarm Optimization (PSO) approach that minimizes a multicriteria evaluation metric that characterizes the effort required by the ASV to avoid the danger, the positional deviation towards the original course and the crossing distance to static and/or dynamic obstacles.

Therefore, the major contributions of this paper include:

- The *AVOA* method is a real-time obstacle avoidance algorithm for ASVs. The method enables a safe interaction between the ASV and the scenario;
- The introduction of a protective zones concept towards a position and velocity configuration space model of the detected obstacles;
- An optimization procedure to estimate an effective velocity through a minimization of a multicriteria condition that characterize both the collision risk and the course deviation;
- An extensive set of experimental validations for comparing the performance of *AVOA* and a recent obstacle avoidance technique, in terms of smoothness, safeness and effectiveness of the avoidance maneuver.

The paper presents a revision of the work in section II. The representation of the environment is described in section III-A and, subsequently, a multicriteria optimization approach based on the Particle Swarm Optimization (PSO) to estimate the velocity for the ASV is proposed in section III-B. The results of the proposed method are presented in section IV and, at last, a brief conclusion appears in section V.

II. Related Work

Towards an obstacle avoidance technique Fiorini et al. in 1998 [2] proposed a configuration space representation using positional and inertial information, merging the obstacles data to a single map. With this a new velocity obstacle space (VO), which lists the set of allowed velocities, is created. The selection of an avoidance velocity is made by a constructive heuristic method selecting the best solution on the feasible set relative to different evaluation metrics. Later Berg et al. (2008) [3] presented an extension of the VO for multiple agents that takes into account the decision of the other agent and allows the selection of a velocity with collision risk if no other decision is possible,

[1]Daniel Filipe Campos is with the Center for Robotics and Autonomous Systems, INESC TEC and Faculty of Engineering of University of Porto, Rua Dr. Roberto Frias, Portugal daniel.f.campos@inesctec.pt

[2]Aníbal Matos is with the Center for Robotics and Autonomous Systems, INESC TEC and Faculty of Engineering of University of Porto, Rua Dr. Roberto Frias, Portugal anibal@fe.up.pt

[3]Andry Maykol Pinto is with the Center for Robotics and Autonomous Systems, INESC TEC and Faculty of Engineering of University of Porto, Rua Dr. Roberto Frias, Portugal andry.pinto@fe.up.pt

978-1-7281-4004-9/19/$31.00 ©2019 IEEE

presenting the Reciprocal Velocity Obstacle (RVO). Snape et al. (2010) [4] and Berg et al. (2011) [5] presented ORCA (Optimal Reciprocal Collision Avoidance) where a representation of the velocity obstacles space is proposed and tested in differential robots and in crowd simulators. Allawi et al. (2015) [6] presented an implementation of a PSO methodology to improve the RVO approach. This uses a heuristic to minimize the velocity distance to a predefined magnitude and heading to the reach the goal pose and maximizing the time to collision. This techniques can also be applied into maritime applications as shown in the approaches proposed by Kuwata et al. (2014) [7] and Kufoalor et al. (2018) [8] applying velocity obstacles for a COLREGS [9] compliant obstacle avoidance. An ant colony algorithm for collision avoidance in maritime environment was proposed by Tsou et al. (2010) [10] that combines common navigational practices, maritime laws/regulations and real-time navigation information to plan a safe path. Rösmann et al. (2017) [11] proposed a Time-Elastic-Band approach that locally optimizes the robot's trajectory with respect to the compliance with kinodynamic constraints and trajectory execution time while maintaining a collision-free path.

III. Adaptive Velocity Obstacle Avoidance

A. Protective Zones Representation

The *AVOA* defines the protective zones as a set of all velocities that leads to a collision [2] and includes kinematic constraints with the cost of danger for classifying the safeness level of the feasible velocity estimative (magnitude and heading).

1) Representation concept: Assuming that A is an agent moving at a velocity v_A in a plane, with the reference pose p_A, and B an obstacle moving in the same plane with the pose p_B and velocity v_B, as shown in Fig. 1; the protective zone represented by PZ_A^B (yellow cone in Fig. 1) can be defined by the set of all velocities v_A that result in a collision with the moving obstacle at a given time interval. The agent and the obstacles are defined by a circular region delimiting the area of the object, represented in blue and red, respectively. For objects with complex shapes an approximation with several circular regions can be made.

The definition of PZ_A^B can be achieved by the Minkowski sum [12] for an agent A and for the obstacle B with a circular shape, $B \oplus A$ (orange circle).

Moreover, by casting a ray (λ) starting in the agent pose and with the heading of the relative velocity from A to B ($v_A - v_B$), the intersection of the Minkowski sum ($B \oplus A$) provides an evidence of a collision at a specific moment in time. Thus, the PZ_A^B can be defined by (1), where $\lambda = p_A + t*(v_A - v_B)$, $t \geq 0$ and θ is the heading. If v_A does not belong to PZ_A^B, there will be no collision however, when v_A is on the boundary then elements A and B will graze the surface of each other at some moment in time.

$$PZ_A^B = \{[v_A, \theta_A] : \lambda \cap (B \oplus A) \neq \varnothing\} \quad (1)$$

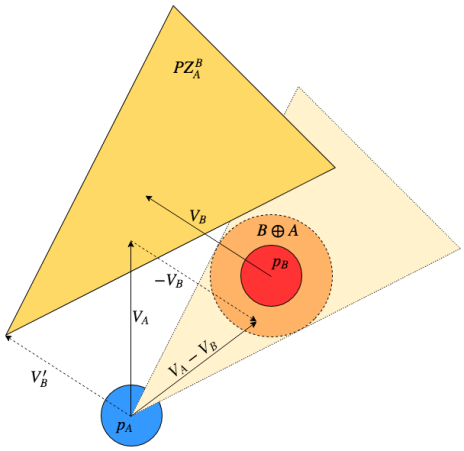


Fig. 1: Protective zone PZ_A^B of an obstacle B relative to agent A.

Nevertheless and despite leading to a collision, a simplification to the protective zone can be made in order to avoid an early route change, which can be obtained by removing all the velocities that would take more than a predefined time (t_H) to lead to a collision from PZ_A^B [2], the green area PZ_{AH}^B of Fig. 2.

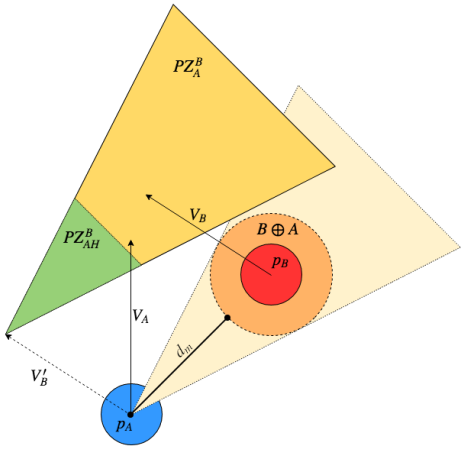


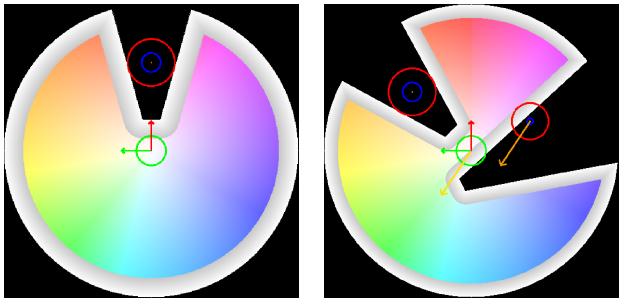
Fig. 2: Protective zone PZ_A^B of an obstacle B relative to agent A considering a non-imminent collision time window (PZ_{AH}^B).

The area PZ_{AH}^B is defined in (2), where d_m is the minimum distance between p_A and $B \oplus A$ and thus, the velocities leading to an imminent collision will be deemed as unfeasible.

$$PZ_{AH}^B = \left\{ [v_A, \theta_A] : v_A \in PZ_A^B, |v_A - v_B| \leq \frac{d_m}{t_H} \right\} \quad (2)$$

The position and velocities of the obstacles are referenced to the agent frame and the velocity space is described by the linear velocity (v) and heading (θ), which means that the tracking velocity information are transposed from $[v_x, v_y] \rightarrow [v, \theta]$. Also, all velocities surpassing a maximum velocity are set as unfeasible, i.e. $v > v_{max} \Rightarrow v \in PZ_R^K$, where PZ_R^K is the protective zone due to the kinematic constraints.

In short, the protective zone defines all velocities that are feasible and do not jeopardize the imminent safety of the vehicle. This concept is improved in AVOA by imposing a safety description defined based on the distance to the nearest PZ_A^B point (e.g., by using the Euclidean distance estimation [13]) to differentiate between tangent crossings and movements in the opposite direction. This will return a decreasing distance gradient from the farther velocity to PZ_A^B. Two examples are presented in Figs. 3a and 3b, depicting a scenario having one static obstacle and, two obstacles (one static and other dynamic), respectively.



(a) Single static obstacle. (b) Static and dynamic obstacles.

Fig. 3: Protective zone representation.

The blue, green and red circles are the obstacle and the agent shape, and the Minkowski sum of both respectively. The orange and yellow arrows are the obstacle velocity in their position and projected to the agent. The black region represents the unfeasible areas, the cone is the $PZ_R^{O_j}$, protective zone relative to j^{th} obstacle, and the image margins are the set of velocities greater then the maximum speed allowed in PZ_R^K. The HSV color space is used to represent the velocity v (saturation) and heading θ (color) for all non-colliding velocities [14], where the softer colors depict velocities with a smaller magnitude and the color transition represents a change in heading relative to the xx axis. The white to dark grey gradient presents the distance to the nearest $PZ_R^{O_j}$ up to a predefined threshold.

2) Kinematic constraints: Usually, an ASV uses a non-holonomic system, normally a differential drive alike kinematics and the thrusters speed are limited and so, they should be formally imposed in the space model defined previously.

Considering an ASV like the Zarco [15], with two thrusters placed side by side, the speed is restricted to a linear (v) and a angular (ω) components and no lateral movement is possible. As so, the speed ($[v_1^t, v_2^t]$) affected to each thruster can be determined from equations (3) and (4) [16].

$$v_1^t = \alpha \cdot v + \beta \cdot \omega \tag{3}$$

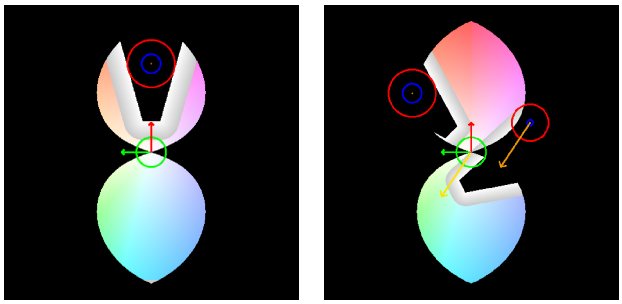
$$v_2^t = \alpha \cdot v - \beta \cdot \omega \tag{4}$$

Where α and β are constants for thrusters placement and configuration within the ASV, and the angular velocity is considered to be proportional to the heading ($\omega \propto \theta$) in the protective zones.

Assuming the maximum velocity of thrusters (v_{max}^t), the kinematic protective zone, PZ_R^K, can be updated by excluding the set of unreachable velocities. Thus, the kinematic protective zone (PZ_R^K) will be defined as (5).

$$PZ_R^K = \left\{ [v_R, \theta_R] : v_R > v_{max} \cup v_1^t > v_{max}^t \cup v_2^t > v_{max}^t \right\} \tag{5}$$

Figs. 4a and 4b demonstrate the visual representation of the protective zones after applying the constraint of (5), see Fig. 3.



(a) Single static obstacle. (b) Static and dynamic obstacles.

Fig. 4: Protective zone representation with kinematics.

B. Avoidance Velocity Selection

After the full representation of the protective zones, it is necessary to estimate a velocity that minimizes the error according to a multicriteria problem formulation. An optimization approach selects a possible velocity, $V_i = [v_i, \theta_i]$, belonging to the feasible set $F = \{V : V \notin PZ_R^K \cap V \notin PZ_R^{O_j}, \forall j = 0, ..., n\}$, where n is the number of obstacles. This solution minimizes the deviation from the predefined speed while maintaining a safe path.

This paper uses the Particle Swarm Optimization (PSO) [17] procedure for searching in the protective zone space for an optimal solution, where a multicriteria objective function is used to evaluate the particles performance.

The following notation is assumed in this paper: the subscript d represents the desired attributes to consider and i represents a solution in the set of all feasible velocities, in the protective zones space.

1) The multicriteria metric: The formulation considered in this paper to measure the quality, ε of the velocity $V_i = [v_i, \theta_i]$, determined by a heuristic procedure, is minimizing the multicriteria objective function defined by (6).

$$\min_{\Delta v_d, \Delta\theta_d, \nabla_i} h(V_i) \tag{6}$$

Where Δv_d is the velocity deviation relative to the expected velocity, $\Delta v_d = v_d - v_i$, $\Delta\theta_d$ is the heading error, $\Delta\theta_d = \theta_d - \theta_i$, ∇_i is the distance cost to the nearest protective zone for a specific solution V_i (cost of danger) that penalizes the proximity of the obstacles protective zone up to a threshold and $h(V_i)$ is a multicriteria heuristic.

This multicriteria objective function is formed by a weighted function (7), where ε is the error vector defined by (8) and ψ is a weight vector with $\sum_{n=0}^{N} \psi_n = 1$.

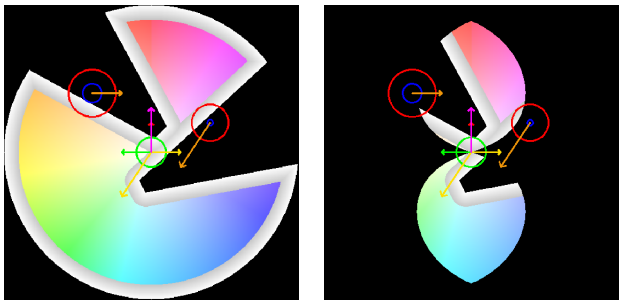
$$h(V_i) = \varepsilon^{\intercal} \cdot \psi \tag{7}$$

As for the error vector ε, (8), all features are normalized to be between $[0,1]$ to ensure the contribution made by each error is given through the weights ψ.

$$\varepsilon = \begin{bmatrix} |\Delta v_d| \\ |\Delta \theta_d| \\ \nabla_i \end{bmatrix}, \ \Delta v_d, \Delta \theta_d \in [-1,1], \ \nabla_i \in [0,1] \tag{8}$$

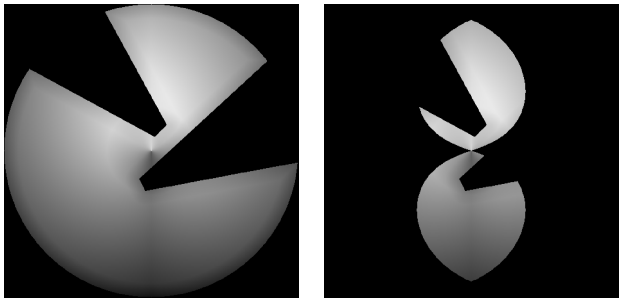
The metric $h(V_i)$ will serve as a quality indicator to be used for the particle fitness evaluation in the PSO.

2) Local Minima Challenge: The nature of the problem and the evaluation metric presents several local minima capable of leading to suboptimal solutions. Fig. 5 depicts an example of a scenario with two dynamic obstacles, where the desired velocity (V_d) represented by the purple arrow. In this scenario the desired velocity, V_d, is classified as unfeasible, since it is inside a protective zone, making the optimal solution with $h(V_i) = 0$ unreachable. This leads to the multicriteria metric representation present in Fig. 6, where the lighter color represents the best solutions.



(a) Without kinematics (b) With kinematics

Fig. 5: Protective zones representation for local minima analysis.



(a) Without kinematics (b) With kinematics

Fig. 6: Local minima map with the scenario of Fig. 5.

As it can be seen, independently of the presence of the kinematic constraints, local minima are always present in this problem. The multicriteria in the evaluation metric causes several minimization requirements, namely ∇_i takes preference for all velocities with zero cost of danger, $|\Delta v_d|$ minimizes towards a circle with radius v_d and $\Delta\theta_d$ to a line starting in the origin and with a slope of θ_d, as so several high intensity crossings can be seen. For instance a bottom-up search could become stuck in a local minima with more than a 90° heading error.

3) Particle Swarm Optimization: The population based metaheuristic, Particle Swarm Optimization (PSO), proposed by Kennedy et al. in [18] was used to allow diversification to search beyond the local minima. This method allows the possibility of pursuing some suboptimal solutions to generate variability in the search pool.

The PSO uses particles with an assigned velocity, $V = [v_l, v_h]$, and pose, $P = [l,h]$, to search the environment for better solutions, where l is the linear velocity and h the heading. At each iteration the particles move throughout the feasible creating a sort of neighborhood where all particles are neighbors to each other. During a single iteration each particle is attracted to is personal best position ($pBest = [l_{ibest}, h_{ibest}]$) and to the best solution found by all the neighborhood ($gBest = [l_{gbest}, h_{gbest}]$), while considering a convergence from the previous velocity ($V_{i-1} = [v_l, v_h]$). At equations (9) and (10) are presented the update rules for a particle at each iteration [17].

$$\begin{aligned} V_i = \ & \varphi_1 \cdot V_{i-1} + \varphi_2 \cdot r_1 \cdot (pBest - P_{i-1}) + \\ & \varphi_3 \cdot r_2 \cdot (gBest - P_{i-1}) + \varphi_4 \cdot \tau \end{aligned} \tag{9}$$

$$P_i = P_{i-1} + V \tag{10}$$

Where the parameters $\varphi_1 = [\varphi_{1l}, \varphi_{1h}]$ controls the inertia in the particle update (should be set to < 1 for convergence), $\varphi_2 = [\varphi_{2l}, \varphi_{2h}]$ the confidence in the personal best solution and $\varphi_3 = [\varphi_{3l}, \varphi_{3h}]$ the best neighborhood solution weights. r_1, r_2, r_3 are random uniform numbers between $[0,1]$ giving a stochastic characteristic to the PSO. As for $\varphi_4 = [\varphi_{4l}, \varphi_{4h}]$ sets a shacking value for a static particle, where $\tau = [\tau_l, \tau_h]$ is a vector of random uniform numbers between $[-1,1]$ if the particle is static, $V = [0,0]$, and $\tau = [0,0]$ otherwise. This last parameter allows variability around the best minima found to create a better diversification since the particles are kept in motion.

In algorithm 1 during initialization all particles are randomly generated to a solution in the feasible set, F, and set to an initial velocity between $[-v_{ini}, v_{ini}]$ due to the uniform random number vector κ generated between $[-1,1]$.

In algorithm 2 the pseudocode for the proposed PSO is unveiled. As it can be seen after the initialization of all the particles, PSO searches for the best solution until the multicriteria metric drops bellow a predefined threshold, accepting it as a reasonable solution, or a maximum number of iterations is reached.

Furthermore, at each iteration the improvement in the particle personal best and in the global best solution is verified for all particles, updating their state accordingly. If a particle is set to an unfeasible space a respawn to F is made.

Algorithm 1 Particle Swarm Optimization - Particle initialization

Input:
Number of particles ($npart$)
List of feasible velocities (f)
Initial particle velocity (v_{ini})
Output:
Particle current solution vector (p)
Particle velocity vector (v)
for all $npart$ **do**
$r \leftarrow \text{rand}(0, size(f))$
$p_i \leftarrow f_r$
$v_i \leftarrow v_{ini} \cdot \kappa$
return p, v

Algorithm 2 Particle Swarm Optimization - Solution evaluation method

Input:
Number of particles ($npart$)
Maximum number of iterations ($maxit$)
Acceptable error for early stop (a_e)
Desired velocity (v_d)
List of feasible velocities (f)
Particle current solution vector (p)
Particle velocity vector (v)
Output:
Global best solution ($gBest$)
initialize($npart$)
while ($iter < maxit$ **and** h(v_d, $gBest$) $> a_e$) **do**
for all $npart$ **do**
objf $\leftarrow$ h(v_d, p_i)
if (objf $<$ h(v_d, $pBest$)) **then**
$pBest \leftarrow p_i$
if (objf $<$ h(v_d, $gBest$)) **then**
$gBest \leftarrow p_i$
for all $npart$ **do**
Solve (9) and (10)
if (p_i == unfeasible) **then**
$initialize\_particle(i)$
return $gBest$

IV. Results

A. Experimental Setup

The performance of the adaptive velocity obstacle avoidance (*AVOA*) algorithm is analyzed based on several scenarios of increasing complexity (prepared through a realistic 3D simulated environment, *Gazebo* [19]). The ASV model used in the aforementioned simulation was the Zarco [15] from the Center for Robotics and Autonomous Systems of INESC TEC. To simulate the obstacles, buoys were used for static elements and a second Zarco ASV was introduced, moving at a fixed rate, to simulate the presence of a concurrent vessel.

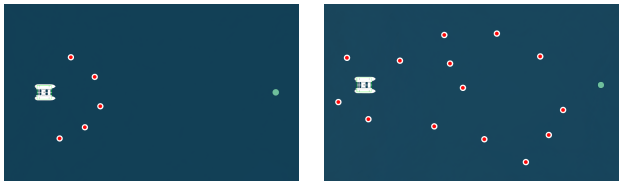
All experiments are considering a zero wind velocity and tidal formed by composition of three waves with the configuration presented in Table I. The *AVOA* method is compared with the Timed-Elastic-Band (TEB) [11] implementation in ROS (*Robot Operating System*) [20], assuming the default parameterization, and both methods defined for the same robot specific parameters, with a maximum linear velocity allowed for the ASVs of 1.5ms^{-1} and an angular velocity of 0.3751rads^{-1}.

TABLE I: Wave configuration for simulation.

Component	Amplitude	Period	Direction (x,y)
1	0.06	12.6	(-1.0,0.0)
2	0.04	3.7	(-0.7,0.7)
3	0.03	6.3	(0.7,0.7)

B. Discussion

Towards the analysis of the method several configuration spaces were defined both with only static obstacles and with dynamic objects to the mix. Starting with the static environments, in Fig. 7 can be seen two test scenarios. First, an u-shaped wall of buoys placed right in front of the vessel, Fig. 7a, while in the second, a cluttered environment, Fig. 7b. On both the green dot represents the desired goal.



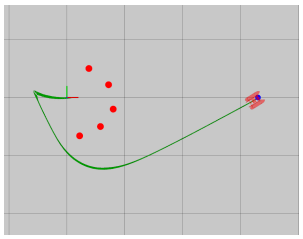
(a) U-shaped wall. (b) Cluttered environment.

Fig. 7: Obstacle avoidance with static obstacles.

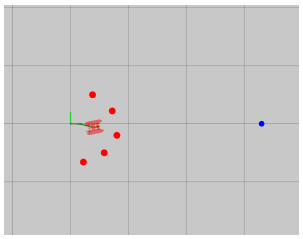
In Figs. 8a and 8b the trajectories from the proposed method and TEB are presented for the u-shaped wall scenario. In this scenario TEB became locked in the local minima defined by the obstacles placement. The *AVOA* was capable of avoiding the entrapment, retreating to find a possible clean path and followed by contouring the obstacles reaching the goal, blue dot. A smooth path with small heading variations was achieved due to the kinematics constraints representation. The travel time in the proposed method was around 33s.

On the cluttered scenario, Figs. 8c and 8d, despite almost all the headings relative to the starting pose presenting a collision risk at some point in time, both approaches performed a collision-free trajectory. This comes to show the removal of the non-imminent collision window (PO^{O}_{RH}) on the proposed method allows the selection of velocities that would cause a collision without compromising the safety. Thus, *AVOA* is capable of prioritizing the avoidance of the most immediate obstacles while maintaining awareness for future decisions.

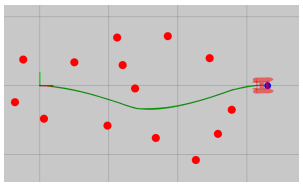
The trajectory obtained with *AVOA* for the cluttered scenario depicts a smoother movement in comparison to TEB. The proposed approach also required a smaller travel time



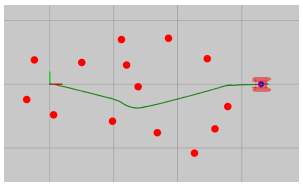
(a) *AVOA* u-shaped wall.



(b) TEB u-shaped wall.



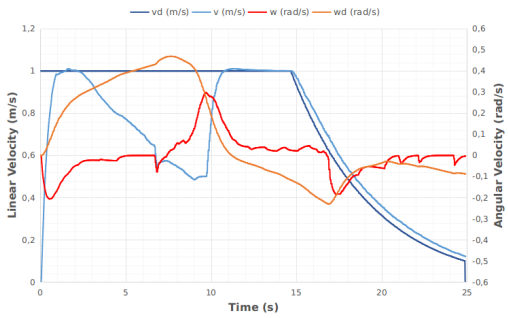
(c) *AVOA* cluttered environment.



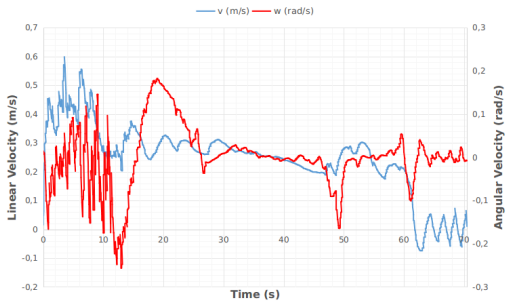
(d) TEB cluttered environment.

Fig. 8: Obstacle avoidance with static obstacles.

to reach the goal, 25s versus the 71s for TEB, showing a 65% improvement. This behavior demonstrates the *AVOA* capability to minimize the deviation towards the desired velocity ($[v_d, \omega_d]$) arranging for smoother velocity transitions, while TEB estimates the velocity with relation to the estimated path inducing a higher variability in the ASV velocity, as is depicted in Fig. 9, where the light blue and red lines represent the robot linear (v in ms^{-1}) and angular (ω in $rads^{-1}$) velocities and the dark blue and orange lines represent the desired linear (v_d in ms^{-1}) and angular (ω_d in $rads^{-1}$) velocities. Higher linear velocities are achieved by *AVOA* compared to TEB.



(a) *AVOA*.



(b) TEB.

Fig. 9: Robot velocities for the cluttered environment.

The test set with static and dynamic obstacles, Fig. 10 presents different scenarios with a single vessel moving at a fixed velocity ($0.55ms^{-1}$), where in scenario A the obstacle vessel moves head on to the ASV, in B the movement is orthogonal to the line between the ASV and the goal and in C the obstacle moves diagonally. In Figs. 11 the results using *AVOA* and TEB are displayed for scenario A (Figs. 11a and 11b), scenario B (Figs. 11c and 11d) and scenario C (Figs. 11e and 11f).

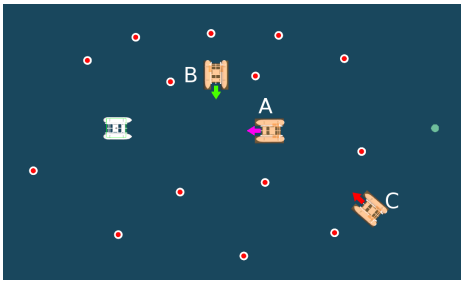
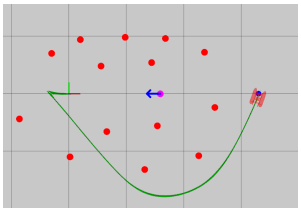
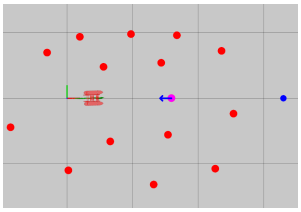


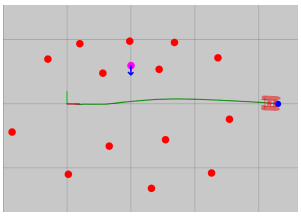
Fig. 10: Obstacle avoidance with a moving dynamic obstacle simulator.



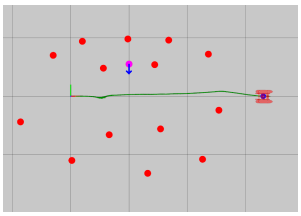
(a) *AVOA* (Scenario A).



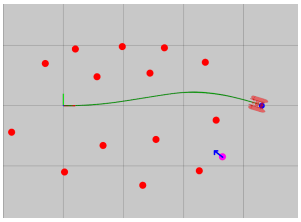
(b) TEB (Scenario A).



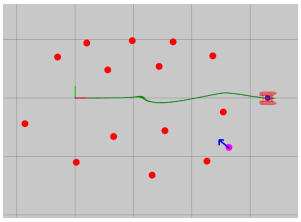
(c) *AVOA* (Scenario B).



(d) TEB (Scenario B).



(e) *AVOA* (Scenario C).



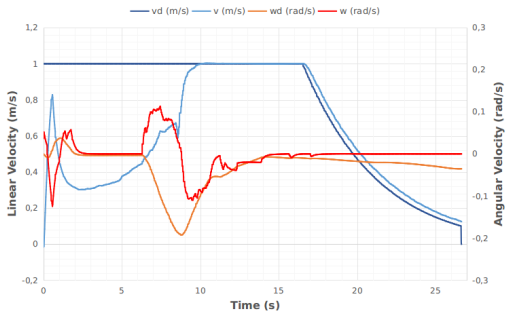
(f) TEB (Scenario C).

Fig. 11: Obstacle avoidance with a moving dynamic obstacle.

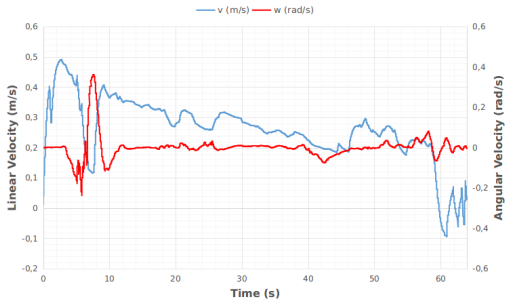
In scenario A, with a head on moving obstacle, the *AVOA* was able to predict the incoming collision, Fig. 11a, where the obstacle initial position is presented as the pink circle with the blue arrow heading. The obstacles placement impedes any middle course crossing, causing the ASV to retreat and find a path through the static obstacles, taking about 35s to reach the goal. On the other hand TEB was not

capable of performing a collision-free path, Fig. 11b.

When considering a laterally moving obstacle, scenario B, both methods present collision-free paths, as shown in Figs. 11c and 11d, taking 27s for *AVOA* to reach the goal and 64s for TEB. This shows a 58% more effective solution for *AVOA*. In the proposed approach the ASV was allowed to remain close to the direct route by slowing down after an initial acceleration, allowing the obstacle vessel to pass and then the ASV carried on with the mission, trying to respect the desired command, as it can be seen in Fig. 12a. As for TEB, due to the slow velocity produced, the obstacle also passed first, however an indecision before the rupture of the initially defined elastic band was observed conducting to a linear velocity reduction, almost to $0ms^{-1}$, while changing the rotation direction, Fig. 12b.



(a) *AVOA*.

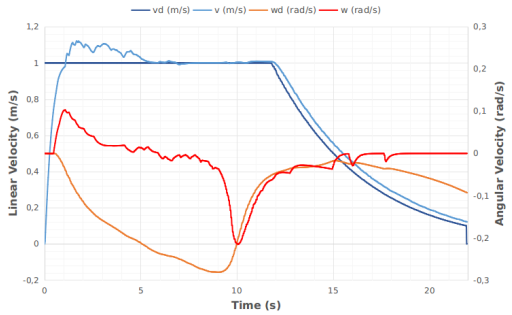


(b) TEB.

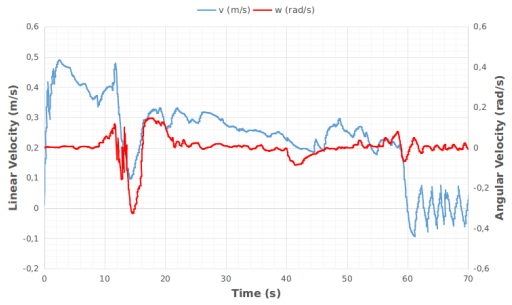
Fig. 12: Robot velocities for the avoidance scenario B.

In the last scenario (C), *AVOA* was able to reach the goal in 22s with the trajectory presented in Fig. 11e. A higher initial velocity than the desired and with a positive heading was estimated to grasp a safe position, before the obstacle vessel reaches the crossing point, Fig. 13a. With this the ASV was able to avoid the collision route first and then proceeded to the target. In this configuration, TEB also generated a complete solution as seen Fig. 11f. A travel time of about 70s was required to finish the route, and as so, a 69% better performance was obtained with *AVOA*. The path performed with TEB allowed the obstacle vessel to cross to the otherside first, however during the trajectory the same indecision as in scenario B occurred, depicted in Fig. 13b.

The performance of *AVOA* on a simulated maritime environment showed promising results in anticipating obstacles



(a) *AVOA*.



(b) TEB.

Fig. 13: Robot velocities for the avoidance scenario C.

as well performing fast and smooth trajectories when compared to TEB. It was able to conclude all the experimental sets contrary to TEB (only concluded three out of five scenarios), and outperformed the travel time with at least 58% more effective results.

In [21] a video with the experimental sets presented in this paper can be found.

V. Conclusion

This paper proposed an obstacle avoidance technique for the estimation of a collision-free velocity. The method is capable of predicting and avoiding threats and, at the same time, it tries to maintain a coherent motion. The algorithm includes a protective zone representation and takes advantage of the kinematic constraints and a cost of danger, formally defined using a multicriteria metric.

Throughout extensive experiments, the *AVOA* method showed promising results towards a collision-free approach for ASVs when considering tidal waves constraints. When compared with an existing algorithm [20], *AVOA* showed an improvement in the travel time of at least 58% and all five courses were finished with no collisions, while for the compared method one collision and one lock situation were obtained.

Moreover the proposed algorithm demonstrates a predictive behavior capable of undertaking preemptive actions to maintain the predefined course. The non-imminent collision window removal allows the selection of an otherwise unfeasible solution in cluttered scenarios, nevertheless maintaining the safety effectiveness of the algorithm. This time window grant the configuration of how preemptive the actions can be.

Therefore, several improvements were proposed in this paper to increase the safety and effectiveness towards a collision-free path.

ACKNOWLEDGMENT

This work is financed by the ERDF European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project POCI-01-0145-FEDER-030010.

REFERENCES

[1] Allianz Global Corporate & Specialty, *Safety and Shipping Review 2018*, 2018.

[2] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, 1998.

[3] J. D. Van Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1928–1935, 2008.

[4] J. Snape, J. van den Berg, S. J. Guy, and D. Manocha, "Smooth and collision-free navigation for multiple robots under differential-drive constraints," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4584–4589, Oct 2010.

[5] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-Body Collision Avoidance," in *Robotics Research* (C. Pradalier, R. Siegwart, and G. Hirzinger, eds.), (Berlin, Heidelberg), pp. 3–19, Springer Berlin Heidelberg, 2011.

[6] Z. Allawi and T. Abdalla, "A pso-optimized reciprocal velocity obstacles algorithm for navigation of multiple mobile robots," *IAES International Journal of Robotics and Automation*, vol. 4, no. 1, p. 31, 2015.

[7] 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, pp. 110–119, Jan 2014.

[8] D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen, "Proactive collision avoidance for asvs using a dynamic reciprocal velocity obstacles method," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2402–2409, Oct 2018.

[9] IMO, "COLREGS - International Regulations for Preventing Collisions at Sea," *Convention on the International Regulations for Preventing Collisions at Sea, 1972*, 2012.

[10] M.-C. Tsou and C.-K. Hsueh, "The Study of Ship Collision Avoidance Route Planning by Ant Colony Algorithm," *Journal of Marine Science and Technology*, vol. 18, no. 5, pp. 746–756, 2010.

[11] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.

[12] M. d. Berg, O. Cheong, M. v. Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Santa Clara, CA, USA: Springer-Verlag TELOS, 3rd ed. ed., 2008.

[13] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory of computing*, vol. 8, no. 1, pp. 415–428, 2012.

[14] A. M. Pinto, P. G. Costa, M. V. Correia, A. C. Matos, and A. P. Moreira, "Visual motion perception for mobile robots through dense optical flow fields," *Robotics and Autonomous Systems*, vol. 87, pp. 1–14, 2017.

[15] N. Cruz, A. Matos, S. Cunha, and S. Silva, "Zarco an Autonomous Craft for Underwater Surveys," in *Proceedings of the 7th Geomatic Week*, 2007.

[16] M. Pinto, A. P. Moreira, and A. Matos, "Localization of mobile robots using an extended kalman filter in a lego nxt," *IEEE Transactions on Education*, vol. 55, no. 1, pp. 135–144, 2012.

[17] A. M. Pinto, A. P. Moreira, and P. G. Costa, "A Localization Method Based on Map-Matching and Particle Swarm Optimization," *Journal of Intelligent & Robotic Systems*, vol. 77, no. 2, pp. 313–326, 2015.

[18] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948 vol.4, Nov 1995.

[19] RoboNation, "Maritime RobotX." https://www.robotx.org/, 2019.

[20] C. Rösmann, "teb_local_planner." http://wiki.ros.org/teb_local_planner, 2019.

[21] D. F. Campos, "Avoa - Adaptive Velocity Obstacle Avoidance." https://youtu.be/jOTsiBlszuA, 2019.