Considerations of Collision Avoidance by a Maritime Robot

Lucian GRIGORE, Amado ȘTEFAN, Florina BUCUR, Bogdan VLĂDESCU,
Cristian MOLDER, and Damian GORGOTEANU

***Abstract*—This work highlights the problem of planning the movement of a maritime robot, in an unstructured environment, to avoid collision. The motion planning algorithm for Unmanned Surface Vehicles (USVs) addresses stationary and moving obstacle avoidance. The algorithm takes into account the International Regulations for the Prevention of Collisions at Sea (COLREGS). To determine the dimensions of the obstacles and the Kinematic and dynamic constraints of the robot we will apply the principle of Velocity Change Space (VCS), using the changes of velocity and direction of the USV. The COLREG handling rules will be considered: crossing, overtaking, orientation. Also, to consider the velocity of the obstacles (non-stationary), the principle of the Velocity Obstacles (VO) will be applied, which generates a cone of the velocity space.**

***Index Terms*—robot, algorithm, velocity, obstacle, constraint.**

I. INTRODUCTION

USVs are autonomous marine vehicles whose command-and-control system can be taken over by a human operator whenever needed. This specification comes in the context of the fact that their use takes place in a maritime environment specific to a port area. As stated, the working environment is unstructured because it introduces disturbances such as: the configuration of the water surface at the interface between the two unstructured environments water and air, underwater currents, air currents, displacement of water volumes near the coasts, wrecks located on the bottom of harbour berths, etc. [1].

The development of USVs is part of the current policy due to the increased need to monitor and secure port environments [2]. Although marine vessels use the Automatic Identification System (AIS), small vessels (USVs) may not use it, as they can change their direction and speed depending on the data they receive from the sensors. These autonomous systems must be able to develop optimized travel paths and adaptive actions that allow them to avoid obstacles and cancel disturbances.

In order to be able to connect ships and plan travel paths, maritime authorities must define the different levels of navigation, operational actions, tolerance and redundancy regarding navigation and monitoring [3].

To create a real-time deliberative route planner so that autonomous ships can avoid obstacles, the travel rules provided in COLREG are used [4,5]. This standardization is the basis of Models of Predictive Control (MPC), which can numerically calculate an optimal trajectory over a finite motion horizon based on predictions of obstacle motion, in order for the algorithm to allow obstacles to be avoided regardless of whether the intersecting vessels comply with the three COLREG rules (Fig. 1).

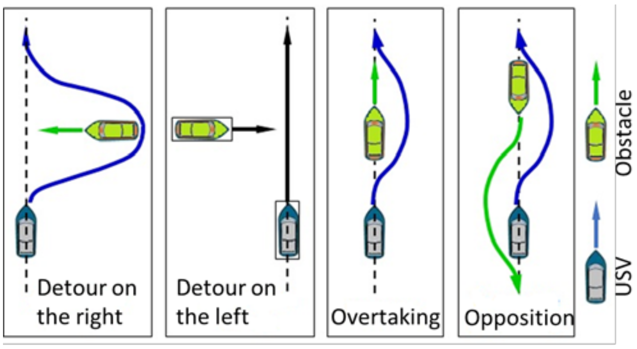

Figure 1. Collision avoidance of maritime vessels according to COLREG [1]

Thus, the paper presents an algorithm based on *VO* [6], which generates a cone-shaped obstacle in the velocity space. The specific velocity of the obstacles and which side of the robot the obstacle is on during the avoidance maneuver are encoded in the speed space in a natural way. The *VO* approach to safe navigation serves as a local planner in the motion planning algorithm.

There are several analytical-numerical models that describe the *VO* to avoid obstacles, but also to generate an additional set of constraints in the VCS velocity space for the situations described by COLREG.

Various identification and control modelling are presented in [7-9], where the objective is to maneuver a ship on desired routes at different velocity. The equations describe the hydrodynamic motion of the rigid body (6 DOF - degrees-of-freedom): Surge, Sway, Heave, Roll, Pitch and Yaw.

"This work was supported in part by the disertation of Master „Design and construction of a ground robot with hybrid undercarriage for monitoring missions" Military Technical Academy, Master „Applied Electronics in Security and Defense Robotics", Center of Excellence in Robotics and Autonomous Systems—CERAS.

L.Ș. GRIGORE and F. BUCUR are with the Military Technical Academy "Ferdinand I", Faculty of Integrated Armament Systems, Military Engineering and Mechatronics, Department of Civil Engineering, Military Engineering and Geomatics, Bucharest, CO 050141 RO (e-mail: lucian.grigore@mta.ro, florina.bucur@mta.ro).

C. MOLDER and D. GORGOTEANU are with the Military Technical Academy "Ferdinand I", Faculty of Communications and Electronic Systems for Defense and Security, Department of Electronic Systems and Military Equipment, CERAS, CO 050141 RO (e-mail: cristian.molder@mta.ro, damian.gorgoteanu@mta.ro)

A. ȘTEFAN and B. VLĂDESCU are with the Military Technical Academy "Ferdinand I", Faculty of Aircraft and Military Vehicles, Department of Integrated Aviation Systems and Mechanics, CO 050141 RO (e-mail: amado.stefan@mta.ro, vladescubogdan23@yahoo.com).

Digital Object Identifier 10.32754/JMT.2023.1.01

The present paper aims to highlight a method for calculating the space and time required to avoid collision by a USV with a potential obstacle. The aim is to present a computational method to be implemented on a collaborative robot system.

II. APPROACH *VO* VELOCITY OBSTACLE

By the *VO* approach we understand that the robot will generate a space through which it defines the respective objects in order to avoid obstacles. The *VO* generates a cone-shaped obstacle whose kinematic characteristics unfold in velocity space. This approach aims to keep the robot from collision as long as the velocity vector is outside the *VO*.

Avoiding a potential hazard is based on anticipating the position of the obstacle it could collide with and of course predicting future positions. For each prediction, the calculations regarding the probability of collision of the two objects are repeated. Thus, the calculations are dependent on certain arbitrary trajectories. *VO* makes linear predictions, and collision checking is done for arbitrary future trajectories.

From the point of view of motion planning we have two problems: route planning and velocity planning. In the first iteration, the optimal route between the static obstacles *PS* (Position Space), *PO* (Position Obstacle) will be calculated, and then the speed along the chosen route, so that they are avoided. If a moving obstacle appears on the path, the robot will have to modify its dynamics (Fig. 2). The *VO* method defines obstacles in the *VS* (Velocity Space), where the velocity is non-linear. Thus, a space of accelerations *AS* (Acceleration Space) will be defined [10]. *VO*, on the other hand, will not specify where and when a collision can occur, for example it can choose the longest path.

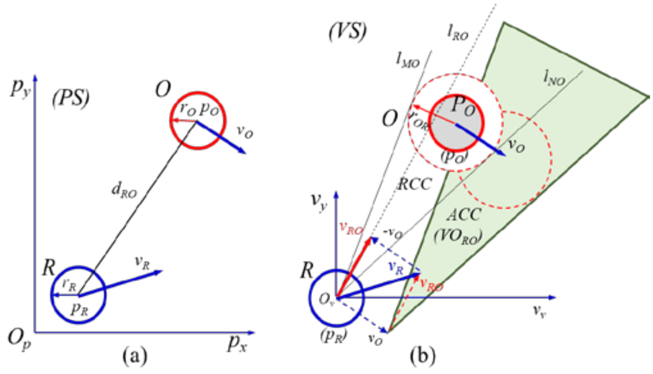

Figure 2. Representation of the velocity cone [10]

In Fig. 2 R is robot; p_R – position; v_R – velocity; r_R – radius of the *R*; *O* – dynamic obstacle; p_O – position; v_O – velocity; r_O – radius of the *O*; d_{RO} – the distance between the centres of the two objects R and *O*; *PO*: $r_{OR} = r_O + r_R$; l_{MO} and l_{NO} – the left tangent radius and the right tangent radius from the *PO* having the position as the starting point p_R; l_{RO} – the radius starting from the position p_R and represents the direction of the velocity v_{RO}.

III. COLLISION CONE - VELOCITY OBSTACLE *VO*

To determine the collision cone, the relative speed of the robot to the obstacle will be calculated $v_{RO} = v_R - v_O$, which helps to establish the space necessary to avoid the collision. The collision condition assumes that the imaginary line represents the direction of the velocity v_{RO} to not intersect *PO*: $l_{RO} \cap PO \neq \varnothing$, equivalent to a set of relative velocities, which form the Relative Collision Cone (RCC – Relative Collision Cone) $RCC = \{v_{RO} \,|\, l_{RO} \cap PO \neq \varnothing\}$. As can be seen (Fig. 2) *RCC* represents the space between the two tangent lines to the obstacle l_{MO} and l_{NO}. In other words, the collision avoidance condition implies that $v_{RO} \notin RCC$.

If the RCC is found in the direction of the obstacle, i.e., along it v_O then we will get a space called the Absolute Collision Cone (*ACC* – Absolute Collision Cone) $ACC = RCC \oplus v_O$, where $\oplus$ – is the Minkowski sum [11].

If the condition is met $\{(v_R \in ACC) \equiv (v_{RO} \in RCC)\}$ then *ACC* represents the multitude of velocity vectors v_R, which can facilitate the realization of the collision between the robot and the obstacle, and is called Velocity Obstacle *VO*.

$$\begin{cases} VO_{RO}(v_O) = \{v_{Rnew} \,|\, (v_{Rnew} - v_O) \in RCC\} \\ v_{Rnew} \in VO_{RO}(v_O) \Leftrightarrow (v_{Rnew} - v_O) \in RCC \end{cases} \tag{1}$$

where: v_{Rnew} – is the new velocity of the robot R.

IV. INTERACTIVE VELOCITY OBSTACLE IVO

In order for the robot to avoid the collision, it will have to choose another speed, different from the one contained in *ACC*, respectively v_{ROnew}. This new speed places itself outside the *RCC*.

If the obstacle is also a robot, then its relative speed will be opposite to that of the base robot (as direction) $v_{ROnew}(v_{ORnew})$ and has an average current velocity value $v_{RO}(v_{OR})$ and one outside the *RCC*:

$$\begin{cases} IVO_{RO}(v_O) = \left\{ v_{Rnew} \,\middle|\, \begin{matrix} \left(v_{Rnew} - v_O = \dfrac{v_{RO} + v_{RO\,\text{oricare}}}{2} \right), \\ v_{RO\,\text{oricare}} \in RCC \end{matrix} \right\} \\ IVO_{RO}(v_O) = \{v_{Rnew} \,|\, (2 \cdot v_{Rnew} - v_R - v_O) \in RCC\} \end{cases} \tag{2}$$

where:

$$[(2 \cdot v_{Rnew} - v_R - v_O) \in RCC] \equiv [(2 \cdot v_{Rnew} - v_R) \in RCC].$$

Depending on how it is positioned relative to the space between the two objects $(p_O - p_R)$ we have two situations:

1. when on the same side of the median, the speed of the obstacles is given by $IVO_{RO}\left(v_O\right)$;
2. when they are not on the same side, it will be chosen $VO_{RO}\left(v_O\right)$.

The existence of the two situations of choosing the speed of the obstacles led to the definition of a hybrid solution, which would satisfy any of the working scenarios, namely the Hybrid Interactive Velocity of the Obstacle *HIVO*:

$$HIVO_{RO}\left(v_O\right)=\begin{cases} VO_{RO}\left(v_O\right) \left| \left(\theta_{ROnew}-\alpha_{RO}\right)\cdot\left(\theta_{RO}-\alpha_{RO}\right)\le 0\right. \\ IVO_{RO}\left(v_O\right) \left| \left(\theta_{ROnew}-\alpha_{RO}\right)\cdot\left(\theta_{RO}-\alpha_{RO}\right)> 0\right. \end{cases} \quad (3)$$

where: $\alpha_{RO}=\angle\left(p_O-p_R\right)$; $\theta_{RO}=\angle\left(v_{RO}\right)$; $\theta_{ROnew}=\angle\left(v_{ROnew}\right)$.

V. MOTION PLANNING IN VCS

The space in which the USV moves, and potential obstacles are characterized by intersections and edges whose configuration is variable in time. This makes motion planning quite complicated. That is why the acceleration selection problem in a *VCS* space will be based on mapping the space and obstacles, but also by establishing dynamic constraints. For planning, the following must be determined: the safety distance, the collision distance, and the collision time.

i. the safety distance is calculated considering the time of the action period T, in which the obstacle accelerates suddenly, a period of time in which the risk of collision between the robot and the obstacle may occur:

$$d_{safe}=r_{OR}+v_{OR}\cdot T+0.5\cdot a_{O\max}\cdot T^2 \quad (4)$$

where: $a_{O\max}$ - the maximum acceleration of the obstacle; $v_R\in VO_{RO}$ - safety distance.

ii. the collision distance is calculated considering the current time before the collision, since the *VO* only specifies that the robot will collide with an obstacle, but not when (time) and where (space):

$$d_c=d_{RO}-r_{OR} \quad (5)$$

$$r_{OR}\equiv r_{ORsafe}=k_{dtf}\cdot\left[f_d\left(\gamma_d\right)+f_t\left(\gamma_t\right)\right]\cdot d_{safe} \quad (6)$$

$$\begin{cases}\gamma_d=k_{df}\cdot\dfrac{d_c}{d_{safe}}\\ \gamma_t=k_{tf}\cdot\dfrac{t_c}{T}\end{cases} \quad (7)$$

where: k_{df} and k_{tf} are distance factors; $f_d\left(\gamma_d\right)$ and $f_t\left(\gamma_t\right)$ are time-varying functions.

iii. time to collision is the time required for the velocity vector to reach the safety limit of the robot and the obstacle:

$$\left|p_{\mathcal{A}}+\tau\cdot\left(v_{\mathcal{A}}-v_{\mathcal{B}}\right)\right|\in\partial\left(\mathcal{B}\oplus-\mathcal{A}\right), \quad (8)$$

where: $\mathcal{A}$ – the outline of the robot; $v_{\mathcal{A}}\left[{m}/{s}\right]$ – the velocity inside the *VO* of the robot; $\mathcal{B}$ – the outline of the obstacle; $v_{\mathcal{B}}\left[{m}/{s}\right]$ – the velocity inside the *VO* of the obstacle; $\tau\left[s\right]$ – time to collision; $\partial\left(\bullet\right)$ – defines the contour to be bypassed by the robot.

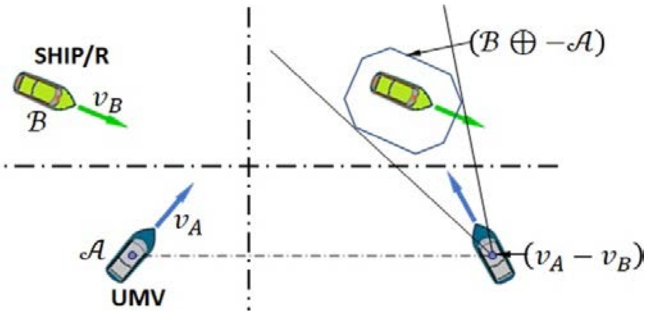

Figure 3. Schematic representation of the principle of determining the collision time [1]

The velocity vector is a representation of the robot crossing the obstacle. It is observed (Fig. 3) that the collision avoidance spaces respect the anti-collision condition if the *VO* falls into a cone shape (velocity cone). The simplified explanation is that as long as $v_{\mathcal{A}}$ (velocity of the robot) is outside the *VO* then no collision will occur.

VI. CONCLUSION

Collision avoidance methods by a USV maritime robot in unstructured dynamic environments are based on the *VS* Velocity Space and *AS* Acceleration Space principles. These methods are quite unstable due to the oscillations of the ships during the movement.

Through the present work, two methods of *VO* and *IVO* collision avoidance have been presented.

In future work we aim to develop a maritime robot system on which to validate this obstacle avoidance method.

REFERENCES

[1] L. Grigore, I. Priescu, D.-L. Grecu, *Artificial Intelligence Applied in Fixed and Mobile Robotic Systems*, AGIR Publishing House, pp. 703, Bucharest, 2020.

[2] B. Tirthankar, L. Sarcione, and F.S. Hover. “A Simple Reactive Obstacle Avoidance Algorithm and Its Application in Singapore Harbor.” *Field and Service Robotics*. Ed. Andrew Howard, Karl Iagnemma, & Alonzo Kelly. Vol. 62. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010 pp. 455–465, http://dx.doi.org/10.1007/978-3-642-13408-1_41.

[3] M. Blanke, M. Henriques, J. Bang, “A pre-analysis on autonomous ships”, Technical Report, Technical University of Denmark DTU Electro, Elektrovej 326 and DTU Management Engineering, Produktionstorvet 426 DK-2800 Kongens Lyngby, 2017.

[4] J. Larson, M. Bruch, R. Halterman, J. Rogers, R. Webster, “Advances in Autonomous Obstacle Avoidance for Unmanned Surface Vehicles”, Technical Report, Space and Naval Warfare Systems Center, San Diego, 53560 Hull Street, San Diego, CA, 92152, 2007.

[5] T.A. Johansen, A. Cristofaro, T. Perez, “Ship Collision Avoidance Using Scenario-Based Model Predictive Control”, in *IFAC-International Federation of Automatic Control*, Vol. 49, Issue 23, 2016, pp. 014–021, https://doi.org/10.1016/j.ifacol.2016.10.315.

[6] Y. Kuwata, M.T. Wolf, D. Zarzhitsky, T.L. Huntsberger, “Safe Maritime Navigation with COLREGS Using Velocity Obstacles”, *IEEE Journal of Oceanic Engineering*, Vol. 39, Issue: 1, January 2014, pp. 110-119, https://doi.org/10.1109/JOE.2013.2254214.

[7] R. Skjente, Ø.N. Smogeli, T.I. Fossen, “A Nonlinear Ship Manoeuvering Model: Identification and adaptive control with experiments for a model ship”, *Modeling, Identification and Control*, vol. 25, No. 1, pp. 3-27, 2004, https://doi.org/10.4173/mic.2004.1.1.

[8] M. Breivik, V.E. Hovstein, T.I. Fossen, “Straight-Line Target Tracking for Unmanned Surface Vehicles”, *Modeling, Identification and Control*, vol. 29, No. 4, pp. 131-149, 2008, https://doi.org/10.4173/mic.2008.4.2.

[9] D.B. Lee, E. Tatlicioglu, T.C. Burg, D.M. Dawson, “Adaptive Output Tracking Control of a Surface Vessel”, *Proceedings of the 47th IEEE Conference on Decision and Control* Cancun, Mexico, Dec. 9-11, 2008, https://doi.org/10.1109/CDC.2008.4739313.

[10] Xunyu Zhong, Xngao Zhong, Xiafu Peng, “Velocity-Change-Space-based dynamic motion planning for mobile robots navigation”, *Neurocomputing*, Vol. 143, 2 November 2014, pp. 153-163, http://doi.org/10.1016/j.neucom.2014.06.010.

[11] H. Barki, F. Denis, F. Dupont, “A new algorithm for the computation of the Minkowski difference of convex polyhedra”, *Conference: SMI 2010, Shape Modeling International Conference*, Aix en Provence, France, June 21-23 2010, pp. 206-210, http://doi.org/10.1109/SMI.2010.12.