sensors

MDPI

Article

A Robust Reactive Static Obstacle Avoidance System for Surface Marine Vehicles

Rafael Guardeño [1,*], Manuel J. López [1], Jesús Sánchez [2], Alberto González [3] and Agustín Conseglière [1]

1 Escuela Superior de Ingeniería, Universidad de Cádiz, 11519 Puerto Real, Spain; manueljesus.lopez@uca.es (M.J.L.); agustin.conseglière@uca.es (A.C.)
2 Sistemas, Navantia, 11100 San Fernando, Spain; jsanchezpa@navantia.es
3 Smart Sustainment, Navantia, Sydneyn NSW 2000, Australia; agonzalezc@navantia.es
* Correspondence: rafael.guardeno@uca.es; Tel.: +34-638-88-70-86

Received: 22 September 2020; Accepted: 30 October 2020; Published: 3 November 2020

Abstract: This paper is centered on the guidance systems used to increase the autonomy of unmanned surface vehicles (USVs). The new Robust Reactive Static Obstacle Avoidance System (RRSOAS) has been specifically designed for USVs. This algorithm is easily applicable, since previous knowledge of the USV mathematical model and its controllers is not needed. Instead, a new estimated closed-loop model (ECLM) is proposed and used to estimate possible future trajectories. Furthermore, the prediction errors (due to the uncertainty present in the ECLM) are taken into account by modeling the USV's shape as a time-varying ellipse. Additionally, in order to decrease the computation time, we propose to use a variable prediction horizon and an exponential resolution to discretize the decision space. As environmental model an occupancy probability grid is used, which is updated with the measurements generated by a LIDAR sensor model. Finally, the new RRSOAS is compared with other SOA (static obstacle avoidance) methods. In addition, a robustness study was carried out over a set of random scenarios. The results obtained through numerical simulations indicate that RRSOAS is robust to unknown and congested scenarios in the presence of disturbances, while offering competitive performance with respect to other SOA methods.

Keywords: unmanned surface vehicle; autonomous navigation; static obstacle avoidance; LIDAR sensor modeling; occupancy probability grid; exponential discretization; estimated closed-loop model; repulsive forces; random scenario generation; unknown and congested scenarios

1. Introduction

Nowadays, the application fields of unmanned surface vehicles (USVs) are very diverse [1], such as environmental control, scientific activities, commercial work, national security issues and surveillance. Consequently, the systems that provide the capacities required in order to consider a vessel as autonomous have become a very active field of research [1–3]. These systems, which are shown in Figure 1, must: estimate the vessel's state vector [4,5], control its course and velocity through the actuators [4,5], process the sensor measurements to generate a model of the environment surrounding the USV [6–11] and carry out safe guidance of the vehicle towards its goal [3–5,7,9,12–22]. Therefore, the autonomy of a USV is determined by the level of development of the estimation, control, obstacle detection and guidance systems, and by their integration in a hardware platform [1–3]. More specifically, a USV's guidance system is formed by three main subsystems: path planning, path following and obstacle avoidance (reactive algorithms). First, planning algorithms [3,7,18,23,24] generate collision-free paths with known obstacles that, in turn, lead the vehicle to its goal. Once a path is established, the path following algorithms change the course and velocity of the USV to follow

it [4,5,19,25]. Finally, due to the presence of unknown obstacles for the path planner, reactive algorithms use the information provided by the obstacle detection system (generated from sensors' measurements) to modify the course/velocity setpoints in order to avoid a collision [9,12–16,26–30]. Of the systems shown in Figure 1, this work is focused on the obstacle avoidance (OA) systems applied to the USV.

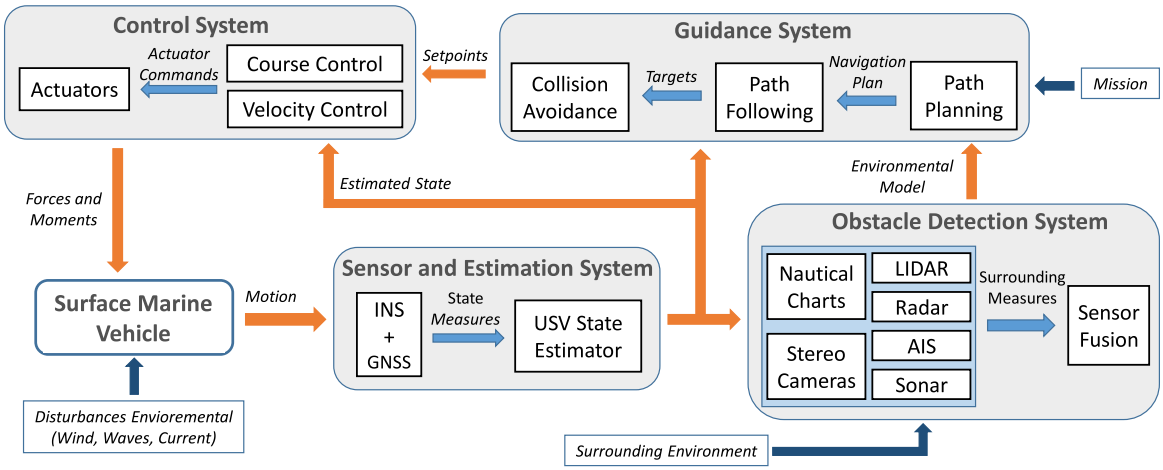

Figure 1. Interaction between the systems of an unmanned surface marine vehicle [31].

1.1. Overview of Reactive Algorithms Applied to USVs

Today there are many reactive methods that provide a vessel with the capacity to avoid obstacles [9,11,13–17,23,25,32–48]. To illustrate this diversity, while the great majority of these methods have been designed for propulsion vessels, the approach proposed in [48] uses a potential reactive field to take into account the kinematic limitations of a sailboat due to wind direction. In the case of propelled vessels, most of these works propose dynamic obstacle avoidance (DOA) systems for situations in open sea [9,13–17,23,33–40,44–47]. Although some of them also consider a specific approach for static obstacles [16,17,23,34,35,37,40,45,46], their performances and robustness have not been evaluated in depth. As a minor proportion, works such as [11,25,32,41–43], propose static obstacle avoidance (SOA) systems for USVs. In addition to the classification based on the kind of obstacle, AO systems for USVs can also be categorized depending on the used approach. Firstly, there is a wide variety of OA systems [11,13,14,32,35,38,40–43,45,48] that use or modify reactive methods proposed for mobile robots [26–28,30,49–54]. Specifically, in [32,35,38,42,45,48] OA systems are based on the potential fields approach [28,49,50]. Alternatively, in [11,40,41,43] the authors adapted the dynamic window [27] and the VFH+ (vector field histogram plus, [30,51]), which they used as SOA systems for USVs. In addition, a collision cone [54] was also used in [40] to avoid dynamic obstacles. One of the DOA methods from mobile robotics most applied to USVs [13,14,44] is velocity obstacles [26] (it has variants [52,53]). On the other hand, in works such as [23,33,37,47,55,56] the OA systems are based on modifications of path planning algorithms like A* [7]. Another important approach was used in [9,15], whose OA systems carried out the fuzzification of target vessel variables through membership functions. Finally, a commonly used approach is based on predicting the possible future paths that the USV could follow if a collision risk were to appear [16,17,34,36,39,40,44,46].

1.2. Related Works

First, before reading this work, we recommended that the reader consults in detail the paper [31], where a new autotuning environment is proposed for static obstacle avoidance methods applied to USVs. Specifically, to develop and evaluate the RRSOAS, the numerical simulation environment proposed in [31] was used. Moreover, several reactive methods [30,35,41,42] were autotuned in that paper, and later used in this work to make a comparison with the RRSOAS.

Due to the intensive research on obstacle avoidance methods applied to mobile robots, there are reactive algorithms that can guarantee the safety of these systems [12,20–22,26,27,30]. For instance, the approach proposed in [22] for an underwater robot is based on two behavioral modules: avoiding obstacles and seeking the goal, which are constituted by two ANFISs (adaptive neuro fuzzy inference systems) that enable the robot's motion in vertical and horizontal planes. Another important approach is proposed in [21], where from a tangent graph (that considers the robot's minimum turning radius) the authors define a probabilistic algorithm that ensures safe navigation to the goal in 100% of cases. In view of these results, the proposal of new reactive algorithms for USVs is justified due to the disturbances inherent to the marine environment [4], and the dynamics and establishment times characteristic of marine surface vehicles [5,57–60]. For these reasons, the reactive algorithm proposed in this work, in line with the approach followed in [16,17,34,36,39,40,44,46], takes into account the USV's dynamics when it requests any change in the control setpoints. In case of a possible collision, OA systems study a set of alternative setpoints for the course and velocity for the vehicle [12,20]. This set, or decision space (DS), contains infinite course/velocity setpoints, so it can be discretized for study. In works such as [11,13,14,16,17,26,27,30,40,41,43], the discrete decision spaces (DDS) are obtained from fixed resolutions. As a novel contribution, we propose to discretize the SD by applying an exponential resolution for the course's setpoints. In this way, a variable resolution is achieved, high in the vicinity of the current course and low in the limits of DS, which reduces the number of alternative setpoints. Furthermore, to take into account the vessel's dynamics, it is common to make predictions of the possible paths that the USV would follow if the course/velocity setpoints were modified. For this purpose, in [16,17,34,36,39,40,44,46] it is necessary to know the mathematical modeling of the vessel and its controllers. In contrast with these methods, RRSOAS does not need these models. Instead, we propose a new estimated closed-loop system model (ECLM) to estimate possible future paths. This ECLM simplifies the dynamics of the closed-loop system under the hypothesis of a correct operation of the course/speed controllers, and in turn, considers the characteristic dynamics of marine surface vehicles [5,57–60]. With respect to the prediction horizon that defines the limit of these future paths, the majority of the reactive algorithms for USVs [16,17,34,36,39,44,46] use a fixed horizon. As a result, depending mainly on the velocity setpoint, these paths cover different distances for the same prediction time. In order to reduce the runtimes of these predictions, this paper proposes to use a variable prediction horizon. In [61] the prediction horizon is defined as a variable that is optimized by a predictive controller designed for unmanned aerial vehicles [62]. Nevertheless, in our work the prediction horizon depends on each pair of alternative setpoints; thus, all possible future paths cover the same distance. Moreover, the uncertainty present in the mathematical model used to make the predictions is not taken into account in [16,17,34,36,39,40,44,46]. This uncertainty, due to the recursive calculation of the predictions, causes an error that increases with the prediction time. In order to make the RRSOAS robust to these prediction errors, we propose to model the USV contour as an ellipse whose perimeter varies with the prediction step. Thus, the shape associated with the vehicle increases with the prediction error. Finally, RRSOAS uses an occupation grid (OG) as a model of the environment [6], which must be generated by an obstacle detection system. OGs have been widely used in mobile robotics [6–8,29] and in guidance systems for USVs [23,24,33,37,47]. As a novel contribution, in contrast to the general approach in which the predictions are evaluated for each obstacle [16,17,34,36,39,40,44,46], we propose to translate the possible future paths to an occupancy probability grid. For this purpose, the concept of repulsive forces proposed in [29] is adapted and an estimated collision time based on the probability of occupation is proposed. In this way, taking into account the USV's dynamics, RRSOAS considers the measurement/estimation errors present in the environment model. In reactive algorithms, it is very common to use a heuristic to determine the control commands requested to the autonomous vehicles [11–14,16,17,20,26,27,30,33,40,41,44,46,47]. Therefore, once the alternative setpoints have been characterized according to a repulsive force and an estimated collision time, both associated with each future path, a heuristic is employed to calculate the course and velocity setpoints demanded by the USV controllers.

The organization of this paper is as follows: mathematical models used to evaluate the new RRSOAS by means of numerical simulations are described in the Section 2. As a simulation environment (USV and LIDAR sensor models), the one proposed in [31] is used. In addition, a Bayesian filter [8] is employed to generate the occupancy probability grids at each sample time of the reactive algorithm [63]. On the other hand, Section 3 presents in detail the four subsystems that form the RRSOAS. As results, the RRSOAS's performance is compared with those of other SOA methods [30,31,41] in Section 4. Furthermore, a study of the robustness of the algorithm was also carried out on nine-hundred obstacle avoidance scenarios, in which the USV navigates at different velocities and is affected by several levels of sea current. Finally, Section 5 contains a general discussion of the results, and in Section 6 the conclusions are presented.

2. Mathematical Models

This section describes the mathematical models used to evaluate the RRSOAS by means of numerical simulations.

2.1. USV Model

As in most OA methods applied to USVs [14–17,25,33–35,38–41,43–47], the algorithm proposed in this work was evaluated through numerical simulations with a three-degrees of freedom vessel model. Furthermore, in line with the general approach followed in this field [11,13,14,16,33–39,41–43], the RRSOAS uses course and velocity controllers to govern the vessel. In particular, the modeling proposed in [31] for a USV of 9.2 m of length is used. Compared with other USV's models [34,38,40,64,65], this modeling takes into account vessel model (1), current effect (2) in terms of relative velocity, actuator model (3) and course/velocity controllers (5). A complete description of these models and the numerical values of their parameters is available in [31].

$$\begin{aligned}
&\dot{u}(m - X_{\dot{u}}) = mvr + mx_g r^2 - Y_{\dot{v}} v_r r + X_u u_r + X_{|u|u}|u_r|u_r + \tau_x \\
&\dot{v}(m - Y_{\dot{v}}) + mx_g \dot{r} = -mur - X_{\dot{u}} u_r r + Y_v v_r + Y_r r + Y_{|v|v} v_r|v_r| + Y_{|r|v}|r|v_r + Y_{|v|r}|v_r|r + \tau_y \\
&\dot{r}(I_z - N_{\dot{r}}) + mx_g \dot{v} = -mx_g ur - N_{uv} v_r u_r + N_r r + N_v v_r + N_{|v|r}|v_r|r + N_{|r|v}|r|v_r + N_{|r|r}|r|r + \tau_n \\
&\underbrace{\begin{pmatrix} \dot{x}_E \\ \dot{y}_E \\ \dot{\psi} \end{pmatrix}}_{\dot{\boldsymbol{\eta}}} = \underbrace{\begin{pmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{R}(\psi)} \underbrace{\begin{pmatrix} u \\ v \\ r \end{pmatrix}}_{\boldsymbol{\nu}}
\end{aligned} \tag{1}$$

where $\boldsymbol{\eta}$ represents the position vector of the USV referring to Earth-axes, $\boldsymbol{\nu}$ and $\boldsymbol{\nu_r}$ are the velocity and relative velocity vectors referring to body-axes, $\boldsymbol{R}(\psi)$ is the rotation matrix from body-axes to Earth-axes (see Figure 2) and $\boldsymbol{\tau_{act}} = (\tau_x \ \tau_y \ \tau_z)^T$ are the forces and moment applied by the actuators.

$$\boldsymbol{\nu_r} = (u_r \ v_r \ r)^T = \begin{pmatrix} (u - V_c cos(\beta_c - \psi)) & (v - V_c sin(\beta_c - \psi)) & r \end{pmatrix}^T \tag{2}$$

with V_c as the module of the current velocity and β_c as its direction.

$$\boldsymbol{\tau_{act}} = \begin{pmatrix} \tau_x \\ \tau_y \\ \tau_n \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -l_x \end{pmatrix} \begin{pmatrix} T_{act} - D_{act} \\ L_{act} \end{pmatrix}, \qquad \begin{aligned} T_{act} &= T_{|n|n}|n|n - T_{|n|u}|n|u_r \\ D_{act} &= D_{|\delta|}|\delta||u_r|u_r \\ L_{act} &= (L_\delta \delta - L_{|\delta|\delta}|\delta|\delta)|u_r|u_r \end{aligned} \tag{3}$$

where l_x is the distance from the actuator to the vessel's center of gravity; n represents the propeller angular velocity; δ is the rudder angle; T_{act} represents the thrust generated by the propeller; and finally,

D_{act} and L_{act} are the drag and lift forces generated by the rudder, respectively. In addition, the dynamics of the actuator are considered as:

$$\begin{aligned} \dot{\delta} &= (1/\tau_\delta)(\delta_c - \delta), \qquad \delta \in [-\delta_{lim}, \delta_{lim}], \qquad \dot{\delta} \in [-\dot{\delta}_{lim}, \dot{\delta}_{lim}] \\ \dot{n} &= (1/\tau_n)(n_c - n), \qquad n \in [n_{min}, n_{max}] \end{aligned} \tag{4}$$

where the time constants τ_δ and τ_n characterize rudder and propulsion dynamics, while δ_c and n_c represent the setpoints for rudder angle and propeller angular velocity, respectively.

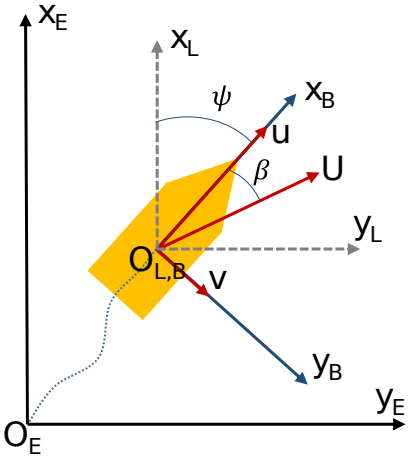

Figure 2. Earth, local and body axis reference systems adopted in this work, where β is the slip angle.

The respective discrete time control algorithms are given by

$$\begin{aligned} \delta_c(k) &= \underbrace{K_{P_\chi} e_\chi(k)}_{\delta_P(k)} + \underbrace{T_m^c K_{I_\chi} e_\chi(k) + \delta_I(k-1)}_{\delta_I(k)} + \underbrace{\frac{K_{D_\chi}}{T_m^c}\left(e_{f_\chi}(k) - e_{f_\chi}(k-1)\right)}_{\delta_D(k)} \\ n_c(k) &= \underbrace{K_{P_U} e_U(k)}_{n_P(k)} + \underbrace{T_m^c K_{I_U} e_U(k) + n_I(k-1)}_{n_I(k)} + \underbrace{\frac{K_{D_U}}{T_m^c}\left(e_{f_U}(k) - e_{f_U}(k-1)\right)}_{n_D(k)} \\ e_{f_i}(k) &= \frac{1}{c_f + 1}\left(e_i(k) + c_f e_{f_i}(k-1)\right), \quad e_i(k) = sp_i(k) - i(k), \quad i \in \{U, \chi\} \end{aligned} \tag{5}$$

where the USV course is defined as $\chi = \psi + \beta$ (see Figure 2), the sample time has been established as $T_m^c = 0.1$ s and $c_f = \tau_f / T_m^c$. Moreover, due to the strong non-linearity of model (1), the authors in [31] use the gain scheduling technique [57,66] to make these PID structures adaptive.

2.2. Sensor Environment Modeling

In order to achieve a realistic simulation environment, this work uses a 2D model of the Ultra Puck LIDAR sensor proposed in [31]. This sensor model has a measurement range $d_{range} = 200$ m and a horizontal field of view $h_{range} = 360°$. Furthermore, looking for the worst case [67], the lowest angular resolution ($h_{res} = 0.4°$) and minimum rotation frequency ($f_m^L = 5$ Hz) were fixed in the numerical simulations. As inputs, as stated in [31], the sensor model requires the USV position vector and a set of segments that define the obstacle scenario (S_E),

$$S_E = \bigcup_{i_O=1}^{n_O} Obst_E^{i_O}, \quad Obst_E^{i_O} = \left\{ \begin{array}{l} seg_{1E}^{i_O} = \left(x_{1E}^{i_O}, y_{1E}^{i_O}, x_{2E}^{i_O}, y_{2E}^{i_O}\right), \; seg_{2E}^{i_O} = \left(x_{2E}^{i_O}, y_{2E}^{i_O}, x_{3E}^{i_O}, y_{3E}^{i_O}\right), \\ seg_{3E}^{i_O} = \left(x_{3E}^{i_O}, y_{3E}^{i_O}, x_{4E}^{i_O}, y_{4E}^{i_O}\right), \; seg_{4E}^{i_O} = \left(x_{4E}^{i_O}, y_{4E}^{i_O}, x_{1E}^{i_O}, y_{1E}^{i_O}\right) \end{array} \right\} \tag{6}$$

with n_O as the number of obstacles that form a scenario, where each obstacle $Obst_E^{i_O}$ is defined as four concatenated segments $seg_{1E\cdots 4E}^{i_O}$ of coordinates $x_{1E\cdots 4E}^{i_O}$ and $y_{1E\cdots 4E}^{i_O}$ referring to Earth-axes.

Next, in [31] the authors use the set of segments L_B to model the LIDAR sensor as a set of beams referring to body-axes:

$$\begin{aligned} L_B &= \left\{ seg_B^{i_L}\left(x_{1B}^{i_L}, y_{1B}^{i_L}, x_{2B}^{i_L}, y_{2B}^{i_L}\right) \mid x_{1B}^{i_L} = y_{1B}^{i_L} = 0,\ x_{2B}^{i_L} = cos(i_L h_{res}) d_{range},\ y_{2B}^{i_L} = sin(i_L h_{res}) d_{range} \right\} \\ I_L &= \left\{ i_L \mid i_L \in \mathbb{N},\ i_L < n_L \right\}, \quad n_L \in \mathbb{N} \end{aligned} \tag{7}$$

where $x_{1B,2B}^{i_L}$ and $y_{1B,2B}^{i_L}$ represent the coordinates of each pair of points that form a beam and n_L is the number of beams that compose L_B ($n_L = h_{range}/h_{res} = 900$).

In order to generate the distances measured by the sensor, in each sample time ($T_m^L = 1/f_m^L$) the set L_B is rotated and translated to Earth-axes as a function of the USV position vector, resulting in set L_E [31]. Then, the points of intersection between the sets of segments S_E and L_E are calculated. Of all the points of intersection obtained for each beam, only the one closest to the USV is considered, while the rest of the intersections with S_E are ignored. Thus, obstacles that are hidden due to the perspective effect are not detected. In addition, a measurement error is added to the detected points depending on their distance to the vehicle. As output, sensor model delivers a vector that contains the measured distances ($d_m^{i_L}$) at each angular position ($\theta_m^{i_L}$); see Equation (8). In this way, a difference with other works that also evaluate OA methods for USVs through numerical simulations [11,13,14,16,33–43], measurement uncertainty is added to the environmental information used by the RRSOAS.

$$\begin{aligned} \boldsymbol{\theta_m^L} &= \begin{pmatrix} \theta_m^0 & \ldots & \theta_m^{i_L} & \ldots & \theta_m^{n_L-1} \end{pmatrix}^T, \quad \theta_m^{i_L} = i_L h_{res} \\ \boldsymbol{D_m^L} &= \begin{pmatrix} d_m^0 & \ldots & d_m^{i_L} & \ldots & d_m^{n_L-1} \end{pmatrix}^T, \quad d_m^{i_L} = \begin{cases} d_{min}^{i_L} + \eta^{i_L} & \text{if } f_{Inter}^{i_L} = 1 \\ d_{range} & otherwise \end{cases} \end{aligned} \tag{8}$$

where $f_{Inter}^{i_L} = 1$ represents that an intersection has occurred between the beam $\theta_m^{i_L}$ and some segment of S_E, $d_{min}^{i_L}$ is the minimum distance of each beam to the obstacles and η^{i_L} is the WGN that affects each measurement, whose variance depends on $d_{min}^{i_L}$ [31].

2.3. Obstacle Scenarios Used to Evaluate the New RRSOAS

This section presents the obstacle scenarios used to evaluate the RRSOAS. These scenarios have been established while taking into account the limitations of reactive obstacle avoidance methods [9,11,13–16,27–30,34–43,49,51], which are designed to avoid unknown obstacles that have not been considered in the path planning [1–3]. To this end, these methods use the environmental information processed in real time from the sensor measurements. Due to measurement errors and the limited ranges of these sensors, reactive methods do not guarantee that the vehicle will reach the goal if there are local minimums [12]. For this reason, in order to provide a global solution to the problem of autonomous navigation [1–3], reactive methods are combined with path planning algorithms [7,18,32,55]. Since in this work the obstacle scenarios are used to evaluate the RRSOAS, without considering its integration with any global algorithm, these do not contain local minimums (as might happen in some marine environments). Instead, unknown scenarios are used, which the USV detects during navigation through the measurements provided by the LIDAR model (7). Specifically, each obstacle scenario is defined by S_E (6), the ocean current (V_c and β_c), a goal waypoint ($\boldsymbol{P_{goal}}$), a goal velocity (U_{goal}, speed over ground [4]) and the initial vectors of position and velocity of the USV ($\boldsymbol{\eta_0}$, $\boldsymbol{\nu_0}$). These scenarios are used to study two features of the RRSOAS: performance and robustness. First, in order to evaluate the performance of the RRSOAS, it is compared with other reactive SOA methods. For this purpose, the five obstacle scenarios proposed in [31] and shown in Figure 3 are used.

However, there are infinite geometric/environmental combinations that define the possible scenarios where a USV can be located. Hence, once the OA system has been adjusted over specific scenarios, it is necessary to study its robustness in other scenarios whose obstacle distributions and environmental conditions are different. To this end, we evaluated the robustness of the RRSOAS through a statistical study. This study was based on the success rate of the reactive algorithm when it guided a USV over a population of randomized scenarios. With respect to most OA systems for USV [13–17,25,32–43,45,47] in which the results shown are limited to a number of specific scenarios, a more exhaustive robustness analysis was performed.

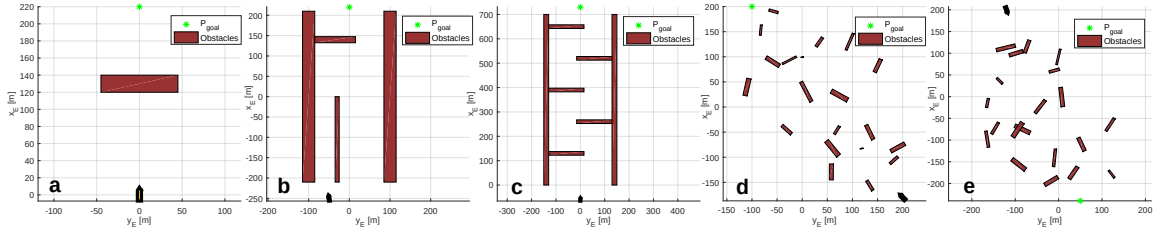

Figure 3. Obstacle scenarios used to evaluate the performances of SOA methods: (**a**) scenario 1, (**b**) scenario 2, (**c**) scenario 3, (**d**) scenario 4 and (**e**) scenario 5. Scenarios' descriptions are available in [31].

To carry out this study, a sample of scenarios (n_E) with a constant number of obstacles (n_o) was generated. These obstacles were randomly generated, as was the current direction, the initial position vector of the USV and the goal waypoint. In particular, the four points of the segments (6) were generated from a rectangle of random dimensions, position and orientation.

$$\boldsymbol{Obst}_{\boldsymbol{E}}^{\boldsymbol{i_O}} = \begin{pmatrix} x_{1E}^{i_O} & y_{1E}^{i_O} \\ x_{2E}^{i_O} & y_{2E}^{i_O} \\ x_{3E}^{i_O} & y_{3E}^{i_O} \\ x_{4E}^{i_O} & y_{4E}^{i_O} \end{pmatrix} = \underbrace{\frac{1}{2}\begin{pmatrix} a^{i_O} & b^{i_O} \\ -a^{i_O} & b^{i_O} \\ -a^{i_O} & -b^{i_O} \\ a^{i_O} & -b^{i_O} \end{pmatrix}\begin{pmatrix} cos(\psi^{i_O}) & sin(\psi^{i_O}) \\ -sin(\psi^{i_O}) & cos(\psi^{i_O}) \end{pmatrix}}_{Rectangle\ Definition} + \underbrace{R^{i_O}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}\begin{pmatrix} cos(\theta^{i_O}) \\ sin(\theta^{i_O}) \end{pmatrix}^T}_{Rectangle\ Placement} \quad (9)$$

Each random rectangle ($\boldsymbol{Obst}_{\boldsymbol{E}}^{\boldsymbol{i_O}}$) is defined by its dimensions (a^{i_O}, b^{i_O}) and its orientation ψ^{i_O}, while its polar coordinates (R^{i_O}, θ^{i_O}) place it on the scenario. These parameters are calculated as:

$$a^{i_O} = a_{max}\zeta_1^{i_O}, \quad b^{i_O} = b_{max}\zeta_2^{i_O}, \quad \psi^{i_O} = \pi\zeta_3^{i_O}, \quad R^{i_O} = R_E\zeta_4^{i_O}, \quad \theta^{i_O} = \pi(2\zeta_5^{i_O} - 1) \quad (10)$$

where $\zeta_{1\cdots5}^{i_O}$ represents a uniformly distributed random number in the interval $(0,\ 1)$; a_{max} and b_{max} define the maximum dimensions of the rectangle; and finally, R_E is a radius that adjusts the dispersion of the obstacles, which defines the occupation zone of the scenario.

As a starting point for all scenarios, the USV is outside of the occupation zone and its heading is pointed towards the goal waypoint, while sailing at a velocity U_{goal}. In this way, the reactive algorithm can always choose the most conservative solution: surround the scenario. In addition, the vessel is affected by a random direction current. Therefore, the initial position and velocity vectors of the USV, along with the β_c angle, are defined as:

$$\begin{gathered} \boldsymbol{\eta_0} = \begin{pmatrix} x_0 & y_0 & \psi_0 \end{pmatrix}^T = \begin{pmatrix} R_0 cos(\theta_0) & R_0 sin(\theta_0) & atan2(y_0,\ x_0) + \pi \end{pmatrix}^T, \quad \boldsymbol{\nu_0} = \begin{pmatrix} U_{goal} & 0 & 0 \end{pmatrix}^T \\ \theta_0 = \pi\left(2\zeta_{\eta} - 1\right), \quad R_0 = R_E + \frac{1}{2}(d_{range} + a_{max}), \quad \beta_c = \pi\left(2\zeta_{\beta c} - 1\right) \end{gathered} \quad (11)$$

where the variables $(\zeta_{\eta},\ \zeta_{\beta_c})$ are random numbers uniformly distributed in the interval $(0,\ 1)$ and the polar coordinates $(R_0,\ \theta_0)$ define the initial location of the USV, whose initial heading (ψ_0) is oriented towards the occupation zone.

On the other hand, the goal waypoint is located in an obstacle-free zone [12]. Moreover, since the USV begins pointing to $\boldsymbol{P_{goal}}$, it is defined as:

$$\boldsymbol{P_{goal}} = \begin{pmatrix} x_{goal} & y_{goal} \end{pmatrix}^T = \begin{pmatrix} x_0 & y_0 \end{pmatrix}^T + 2R_0 \begin{pmatrix} cos(\psi_0) & sin(\psi_0) \end{pmatrix}^T \tag{12}$$

Finally, in line with most of the OA systems [11–14,16,17,26,27,30,33,40,41], it is necessary to define a goal course (χ_{goal}) at each simulation instant (k). In this paper the USV's mission is to arrive at $\boldsymbol{P_{goal}}$, so the goal course is defined by the Equation (13). If the mission involves following a concrete path, χ_{goal} must be calculated according to the kind of path [4,5,19,25].

$$\chi_{goal}(k) = atan2\left(y_{goal} - y_E(k),\ x_{goal} - x_E(k)\right) + \psi(k) \tag{13}$$

In short, a sample of scenarios is defined by the vector $\Theta_E = (n_E\ n_0\ R_E\ a_{max}\ b_{max})$, where n_E is the number of scenarios that form the sample; n_0 sets the number of obstacles; R_E limits the zone of occupation; and finally, a_{max} and b_{max} define the maximum dimensions of the obstacles. In addition, each sample of scenarios can be parameterized in terms of U_{goal} and V_c. Thus, it is possible to study how the magnitudes of environmental disturbances affect the RRSOAS at different points of operation of the USV. As illustrative example, Figure 4 shows eight random scenarios generated by this procedure.

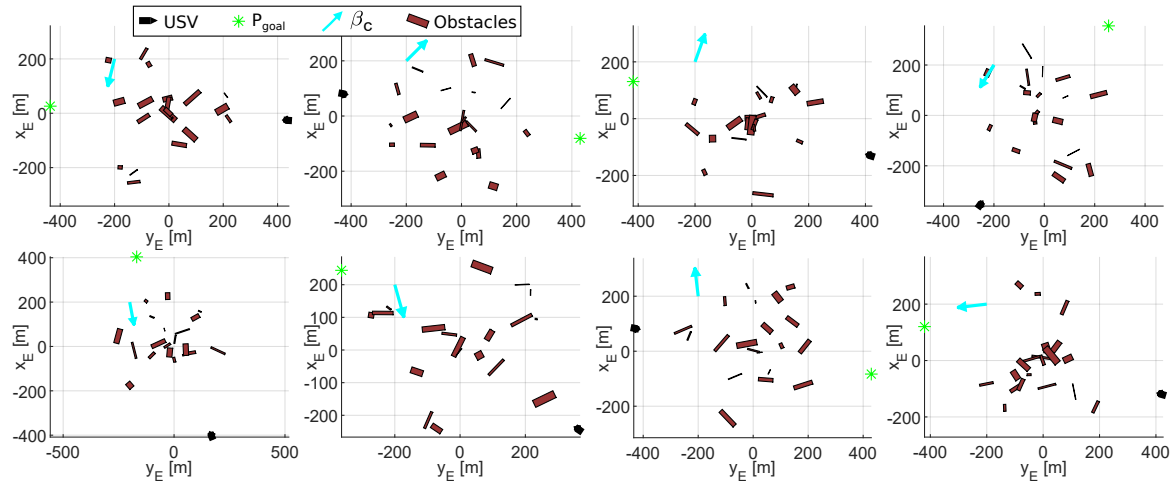

Figure 4. Random sample of obstacle scenarios, where $\Theta_E = (8\ 20\ 300\ 75\ 25)$.

2.4. Model of the Environment Surrounding the USV: Occupation Grid

In order to generate a model of the environment that surrounds the USV, the obstacle detection systems [6–10] process and fusion the information obtained from different sensors and data sources (see Figure 1). Thus, the guidance system can use all the information to carry out a path planning and avoid any unforeseen obstacle. In this work, the obstacle detection system generates an Occupation Probability Grid (OPG) [6,8,29] which is used by the RRSOAS. OPGs allow decoupling the OA system and the complexity of the scenario (shape and number of obstacles). For this purpose, it is necessary to fix a resolution (g_{res}) in order to discretize the space around the vehicle in cells, which quantify the uncertainty contained in the measurements as an occupancy probability. Hence, the environmental analysis carried out by the reactive method depends only on the grid dimensions $(m_g \times n_g)$. In this work, OPG is centered on the USV and it moves with the vehicle [29,47]. Specifically, the reference system of the OPG matches with the vessel's local horizon reference system; see Figure 2. Furthermore, based on the numerical simulations carried out and in line with the cell size used in [47] for a vessel

similar to the USV (1), the resolution has been set as $g_{res} = 1$ m. Moreover, with the aim of using the maximum detection range of the sensor (7), the grid is set square and its dimension depends on d_{range}.

$$i_g \in [1, n_g], \qquad j_g \in [1, m_g], \qquad n_g = m_g = 2\lceil d_{range}/g_{res} \rceil + 1 \tag{14}$$

where $\lceil x \rceil$ represents the function *floor(x)* and $(i_g,\ j_g)$ are the indexes of the grid.

Since this work is not focused on the development of a obstacle detection system, in the numerical simulations OPG was calculated using the *MATLAB Navigation Toolbox* [63]. In particular, the *occupancyMap* function was used to define OPG according to its dimensions and resolution. This function is based on a Bayesian filter, an approach widely used to generate OPGs [8]. Once the grid is defined, in each sample time T_m^L the measurements delivered by the LIDAR sensor (7) are translated to this grid. For this purpose, the *insertRay* function is used, whose inputs are directly compatible with the measured distances (8) and the position vector of the USV, $\boldsymbol{\eta(k)}$. Furthermore, in order to avoid possible truncation errors caused by the discretization of the space, function *inflate* is used to expand each occupied cell a radius equivalent to g_{res}. Finally, the function *occupancyMatrix* is used to obtain the probability of occupation of all the cells (p_{ij}); see Equation (15). As a result, Figure 5 shows: the USV navigating over an obstacle scenario, the LIDAR sensor measurements and the OPG.

$$\boldsymbol{g_p}(k) = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nm} \end{pmatrix} = \boldsymbol{f_B}\left(\boldsymbol{\theta_m^L}(k),\ \boldsymbol{D_m^L}(k),\ \boldsymbol{\eta}(k),\ \boldsymbol{g_p}(k-1)\right) \tag{15}$$

where $\boldsymbol{f_B}$ represents the Bayesian filter used to generate the OPGs at each sample time k.

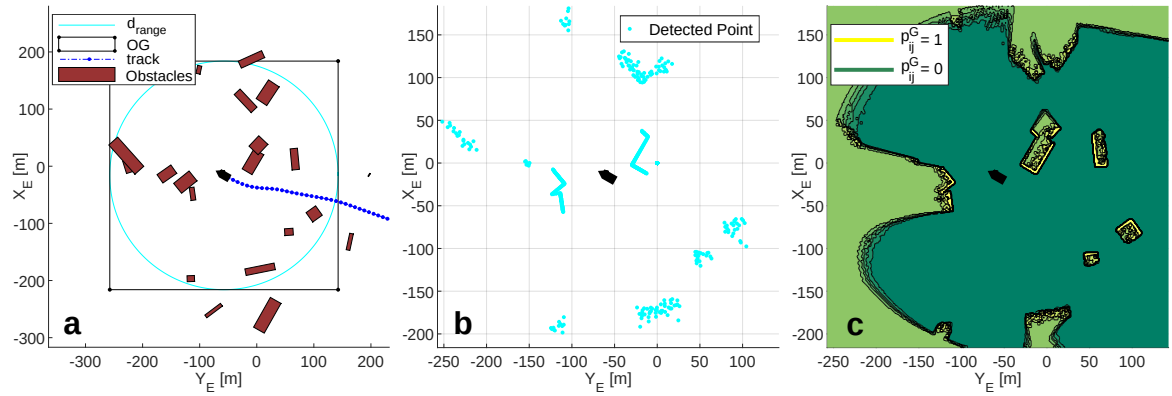

Figure 5. (**a**) Obstacle scenario. (**b**) LIDAR sensor measurements. (**c**) Occupancy probability grid.

3. Robust Reactive Static Obstacle Avoidance System

This system has been designed from the synthesis of several basic concepts in the field of OA methods [12,27,29,30,51], taking into account general criteria used in USVs [16,17,34,40] and new contributions in this field in order to increase the robustness and applicability of SOA methods to vessels. A general scheme of this new method is shown in Figure 6. As inputs, the algorithm requires the state vector of the USV ($\boldsymbol{x_{USV}}$), an OPG ($\boldsymbol{g_p}$) and goal setpoints ($\chi_{goal},\ U_{goal}$). As outputs, the RRSOAS generates the course and velocity setpoints $\boldsymbol{sp}$ $(sp_\chi,\ sp_U)$ for the USV controllers.

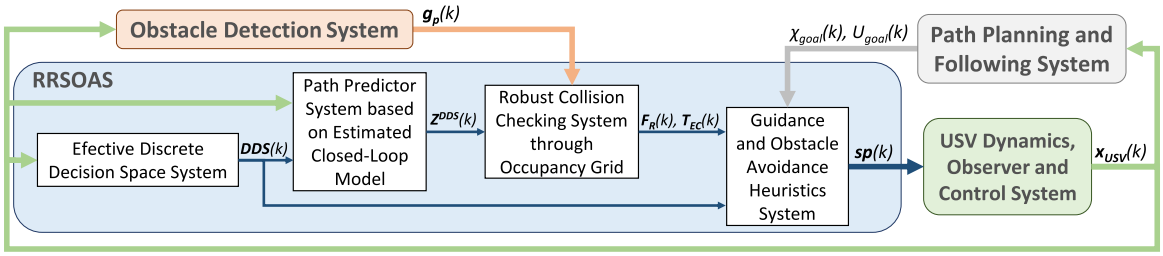

Figure 6. Functional diagram of the RRSOAS: inputs/outputs and the main subsystems.

In each sample time (T_m) this reactive algorithm uses four subsystems:

- *Efective discrete decision space (DDS) System*. With the objective of avoiding a possible collision, this block generates a set of alternative government setpoints.
- *Path predictor system based on estimated closed-loop model*. Taking into account the USV's state vector ($\boldsymbol{x_{USV}}$), *DDS* is used to make predictions of possible paths ($\boldsymbol{Z^{DDS}}$) that the USV could follow to avoid a collision.
- *Robust collision checking system through occupancy grid*. The $\boldsymbol{Z^{DDS}}$ paths are translated into the occupancy grid and characterized by a repulsive force ($\boldsymbol{F_R}$) and an estimated collision time ($\boldsymbol{T_{EC}}$).
- *Guidance and obstacle avoidance heuristic system*. Finally, based on $\boldsymbol{F_R}$ and $\boldsymbol{T_{EC}}$, and the goal setpoints (χ_{goal}, U_{goal}), a heuristic is used to restrict, weight and decide over *DDS* the setpoints that are demanded by the controllers; $\boldsymbol{sp}$ $(sp_\chi,\ sp_U)$.

3.1. Effective Discrete Decision Space System

When sailing in the presence of obstacles, a vessel's current velocity and course can lead to a collision. In order to avoid such possible collisions, it is necessary to study a set of alternative setpoints for the course (sp_χ) and velocity (sp_U) of the USV [11–14,16,17,41]. The combination of these alternative setpoints defines the decision space at each sample time (k):

$$DS(k) = \left\{ \boldsymbol{sp}(sp_\chi, sp_U) \mid sp_\chi \in \left[\chi(k) - \Delta\chi_{max},\ \chi(k) + \Delta\chi_{max}\right],\ sp_U \in [0,\ U_{lim}] \right\} \tag{16}$$

where $\Delta\chi_{max}$ and U_{lim} are adjustment parameters that define the maneuvering range of the USV, limiting the values of the alternative course/velocity setpoints.

Since the set (16) is continuous, it can be discretized according to a resolution [13,14,16,17,26,27,30,40,41]. A disadvantage when a fixed resolution is used for the course setpoints and DS is centered on the current vehicle course, a standard approach followed in mobile robotics [12,26,27,30,51], is that the resolution must be increased to reduce the tracking error of the goal course [40]. Consequently, the calculation time increases with the number of alternative setpoints. An alternative approach is to center DS directly on the goal course [16,37]. In this way, the authors ensure that the guidance of the USV follows χ_{goal} without the need to use high resolutions in the discretization of the course commands. Although, as a disadvantage, the symmetry (number of alternative setpoints to port and starboard) of space (16) depends on the angular distance between χ and χ_{goal}. In order to reduce the tracking error with respect to χ_{goal} without using high resolutions, while DS is centered around χ, we propose an exponential discretization for the course setpoints; see Equation (17). This discretization limits candidate courses to bring navigation more in line with COLREGS [68], carrying out minor corrections to compensate any course changes due to disturbances and applying significant course changes (easily perceived by other vessels [68]) if there is a risk of collision. Moreover, although the obstacles present in the environment force the RRSOAS to demand changes of course in only one direction, the symmetry of space (16) is not affected.

$$SP_\chi(k) = \left\{ sp_\chi \mid sp_\chi = \chi(k) \pm \Delta\chi_{max} e^{(-i_\chi/\tau_\chi)},\ i_\chi \in [0, n_\chi] \right\} \tag{17}$$

where τ_χ is a positive parameter that defines the variation of the resolution along the range $\Delta\chi_{max}$ and n_χ represents the even number of alternative courses (port/starboard symmetric).

On the other hand, the use of low resolutions to discretize velocity setpoints does not affect to the performances of OA systems for USVs [16,17]. For this reason, two fixed resolutions are used in this work: U^+_{res} and U^-_{res}, which define alternative velocities higher and lower than U_{goal}, respectively. In addition, goal velocity and zero velocity (for unavoidable collision situations) are also considered as commands. Thus, the alternative setpoints for the velocity of navigation are given by:

$$SP_U(k) = \begin{cases} \{0, U_{goal}(k)\} & \text{if } n_U = 0 \\ \{0, U_{goal}(k)\} \cup SP^+_U \cup SP^-_U & \text{otherwise} \end{cases} \tag{18}$$

where n_U is the even number of alternative velocities, while the sets SP^+_U and SP^-_U are defined as:

$$\begin{aligned} SP^+_U &= \left\{ sp_U \mid sp_U = U_{goal}(k) + i_U U^+_{res}(k) \right\}, \quad U^+_{res}(k) = \left(U_{lim} - U_{goal}(k) \right) / n_U \\ SP^-_U &= \left\{ sp_U \mid sp_U = U_{goal}(k) + i_U U^-_{res}(k) \right\}, \quad U^-_{res}(k) = \left(U_{gov} - U_{goal}(k) \right) / n_U \end{aligned}, \quad i_U \in [1, n_U] \tag{19}$$

where U_{gov} is the minimum velocity at which the course controller can govern the vessel.

As a result, using an exponential resolution (17) to discretize the courses and two fixed resolutions (18) for the velocities, the decision space (16) is discretized according to Equation (20). Figure 7 shows one of the possible discrete decision space (DDS) that defines the set of alternative setpoints (sp_χ, sp_U) evaluated by the RRSOAS.

$$DDS(k) = \left\{ \boldsymbol{sp}^{i_c}(sp_\chi, sp_U) \mid sp_\chi \in SP_\chi(k),\ sp_U \in SP_U(k) \right\}, \quad i_c \in \left([1, n_c] \cap \mathbb{N} \right) \tag{20}$$

where $\boldsymbol{sp}^{i_c}(sp_\chi, sp_U)$ represents a pair of possible government setpoints and $n_c = (2n_\chi + 1)(2n_U + 2)$ defines the total number of alternative government setpoints taken into account.

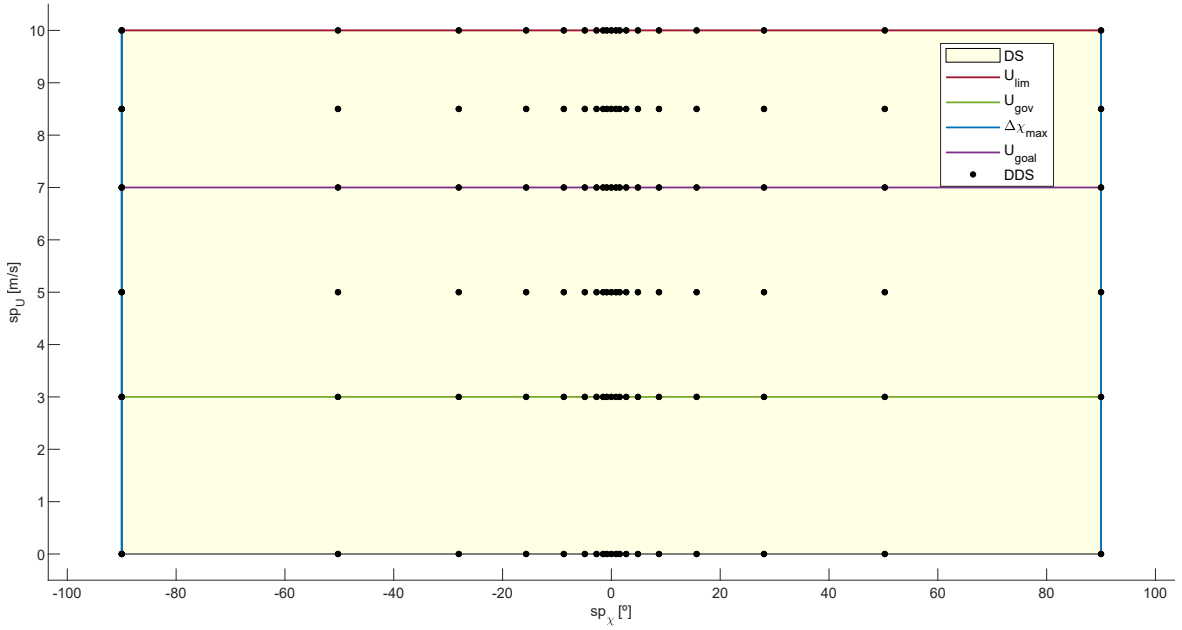

Figure 7. Discretization of space (16). Settings: $\chi(k) = 0°$, $n_\chi = 9$, $\tau_\chi = 1.7$ and $n_U = 2$.

Finally, it is necessary to consider the oscillations that external disturbances (wind, waves and current) cause in the vessel's course. These oscillations, although corrected by the autopilot in a finite

time, may generate undesired changes in DDS (20), since these are not due to avoidance or following maneuvers. To reduce the variation of set (17) caused by these oscillations, in this work it is proposed to apply hysteresis (21) to the USV's course, so χ is replaced by χ_{hyst} in Equations (16) and (17).

$$\chi_{hyst}(k) = \begin{cases} \chi(k) & \text{if } |\chi(k) - \chi_{hyst}(k-1)| \geq \kappa_\chi \\ \chi_{hyst}(k-1) & \text{otherwise} \end{cases} \tag{21}$$

where κ_χ is the hysteresis range, set according to the sea state and the autopilot's performance.

3.2. Path Predictor System Based On Estimated Closed-Loop Model

This system estimates the possible future paths that the USV could take to avoid a collision. For this purpose, it does not need the mathematical modeling of the vessel or its controllers. Instead, it uses the new estimated closed-loop model proposed in this paper. Moreover, each prediction horizon is adjusted to ensure that all future paths associated with DDS (20) cover the same distance.

3.2.1. Estimated Closed-Loop Model

In the first place, the closed-loop system formed by the vessel and the course/velocity controllers is modeled as two second-order differential equations with delay; see expressions (22) and (23). Each of these models relates the setpoints (sp_χ, sp_U) with a linear estimation of the controlled variables $(\tilde{\chi}_l, \tilde{U}_l)$. Both models are defined by: effective time constants (τ_χ, τ_U), effective delay times (d_χ, d_U), damping coefficients (ζ_χ, ζ_U) and stationary gains (K_χ, K_U). These last ones, due to the error compensation made by the controllers to ensure that the USV follows the setpoints, are taken as $K_\chi = K_U = 1$.

$$\frac{d\tilde{\chi}_l(t)}{dt} = \tilde{r}_l(t), \qquad \frac{d\tilde{r}_l(t)}{dt} = \frac{1}{\tau_\chi^2}\left(K_\chi sp_\chi\,(t - d_\chi) - 2\zeta_\chi\tau_\chi\tilde{r}_l(t) - \tilde{\chi}_l(t)\right) \tag{22}$$

$$\frac{d\tilde{U}_l(t)}{dt} = \tilde{a}_l(t), \qquad \frac{d\tilde{a}_l(t)}{dt} = \frac{1}{\tau_U^2}\left(K_U sp_U\,(t - d_U) - 2\zeta_U\tau_U\tilde{a}_l(t) - \tilde{U}_l(t)\right) \tag{23}$$

where $\tilde{r}_l$ and $\tilde{a}_l$ are estimates of the angular velocity and linear acceleration of the USV, respectively.

Taking into account the non-linear dynamics characteristic of marine surface vehicles and the strong coupling between their state variables [5,57–60], it is necessary to consider additional effects on the linear dynamics represented by models (22) and (23). First, speed falls due to changes of course are considered. Physically, this is caused by the coupling that exists between the components of the vessel's velocity vector ($\boldsymbol{\nu}$) [5,57–60,69] and the drag forces generated by the action of the actuators; see Equations (1) and (3). To model this effect, the estimated linear velocity is redefined and depends on the estimated angular velocity:

$$\tilde{U}(t) = \tilde{U}_l(t - T_p) - |\tilde{r}(t - T_p)|\left(c_1\tilde{U}^2(t - T_p) + c_2\tilde{U}(t - T_p) + c_3\right), \quad \tilde{U}(t) \in \left[U_{gov},\ U_{lim}\right] \tag{24}$$

where (c_1, c_2, c_3) are positive coefficients that model the loss of velocity experienced by the vessel during a change of course, T_p is the integration step and $\tilde{U}$ is limited to the USV navigation range.

Secondly, it is necessary to take into account the non-linear dynamics of the vehicle's rotational rate. According to [5,57–60], in vessels the dynamics of r becomes faster when the forward velocity increases. Therefore, the parameters of the model (22) that define the rapidity of the course dynamics become non-linear and variable depending on $\tilde{U}$,

$$\tau_\chi(t) = c_4 + c_5/\tilde{U}(t - T_p) + c_6/\tilde{U}^2(t - T_p), \qquad d_\chi(t) = c_7 + c_8/\tilde{U}(t - T_p) \tag{25}$$

where the coefficients $(c_4, c_5, c_6, c_7, c_8)$ are defined positive to ensure the stability of the ECLM and model how $\tau_\chi(t)$ and $d_\chi(t)$ decrease when the velocity of the vessel increases. Thus, the angular velocity and course of the new ECLM is defined as:

$$\tilde{\chi}(t) = \int \tilde{r}(t)dt, \quad \tilde{r}(t) = \int \frac{1}{\tau_{\tilde{\chi}}^2(t)} \left(sp_\chi \left(t - d_\chi(t) \right) - 2\zeta_\chi \tau_\chi(t) \tilde{r}(t) - \tilde{\chi}(t) \right) dt, \quad \tilde{r}(t) \in \{-r_{max}, r_{max}\} \tag{26}$$

with r_{max} as the maximum rotation rate of the vessel.

Finally, the positions $(\tilde{x}_E, \tilde{y}_E)$ are estimated according to:

$$\tilde{x}_E(t) = \int \tilde{U}(t) cos\left(\tilde{\chi}(t)\right) dt, \qquad \tilde{y}_E(t) = \int \tilde{U}(t) sin\left(\tilde{\chi}(t)\right) dt \tag{27}$$

Then, Equations (23)–(27) are discretized using the forward Euler method with the aim of predicting possible future paths. As a pre-tuning value, the integration step T_p is established equal to the sample time of the Controllers (5). By regrouping terms, the new ECLM is obtained, a non-linear and time-varying model:

$$\underbrace{\begin{pmatrix} \tilde{a}_l(k) \\ \tilde{U}_l(k) \\ \tilde{U}(k) \\ \tilde{r}(k) \\ \tilde{\chi}(k) \\ \tilde{x}_E(k) \\ \tilde{y}_E(k) \end{pmatrix}}_{\boldsymbol{x_{ECLM}}(k)} = \underbrace{\begin{pmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 & 0 \\ T_p & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & a_{33} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{44} & a_{45} & 0 & 0 \\ 0 & 0 & 0 & T_p & 1 & 0 & 0 \\ 0 & 0 & a_{63} & 0 & 0 & 1 & 0 \\ 0 & 0 & a_{73} & 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{A_{ECLM}}(k-1)} \underbrace{\begin{pmatrix} \tilde{a}_l(k-1) \\ \tilde{U}_l(k-1) \\ \tilde{U}(k-1) \\ \tilde{r}(k-1) \\ \tilde{\chi}(k-1) \\ \tilde{x}_E(k-1) \\ \tilde{y}_E(k-1) \end{pmatrix}}_{\boldsymbol{x_{ECLM}}(k-1)} + \underbrace{\begin{pmatrix} b_{11} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & b_{32} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}}_{\boldsymbol{B_{ECLM}}(k-1)} \underbrace{\begin{pmatrix} sp_U(k-\gamma_1) \\ sp_\chi(k-\gamma_2) \end{pmatrix}}_{\boldsymbol{u_{ECLM}}(k-1)}$$

$$\underbrace{\begin{pmatrix} \tilde{\chi}(k) \\ \tilde{x}_E(k) \\ \tilde{y}_E(k) \end{pmatrix}}_{\boldsymbol{y_{ECLM}}(k)} = \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{C_{ECLM}}} \boldsymbol{x_{ECLM}}(k), \qquad \underbrace{\begin{pmatrix} a_{11} \\ a_{12} \\ \gamma_1 \end{pmatrix} = \begin{pmatrix} 1 - (2\zeta_U T_p)/\tau_U \\ -T_p/\tau_U^2 \\ 1 + \lceil d_U/T_p \rceil \end{pmatrix}}_{Non\text{-}Time\ Dependent\ Parameters} \tag{28}$$

$$\underbrace{\begin{pmatrix} a_{33} \\ a_{44} \\ a_{45} \\ a_{63} \\ a_{73} \\ \gamma_2 \end{pmatrix} = \begin{pmatrix} c_1 & c_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\zeta_\chi & 0 & 0 & 0 \\ 0 & 0 & 1/\tau_\chi & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -|\tilde{r}|\tilde{U} \\ -|\tilde{r}|/\tilde{U} \\ -T_p/\tau_\chi \\ T_p cos(\tilde{\chi}) \\ T_p sin(\tilde{\chi}) \\ \lceil d_\chi/T_p \rceil \end{pmatrix} - \begin{pmatrix} |\tilde{r}|c_2 \\ -1 \\ 0 \\ 0 \\ 0 \\ -1 \end{pmatrix}, \quad \begin{pmatrix} \tau_\chi \\ d_\chi \end{pmatrix} = \begin{pmatrix} c_4 & c_5 & c_6 \\ c_7 & c_8 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1/\tilde{U} \\ 1/\tilde{U}^2 \end{pmatrix}}_{Time\ Dependent\ Parameters\ (k-1)}$$

where $\lceil x \rceil$ represents the function *floor(x)*, $b_{32} = -a_{45}$ and $b_{11} = -a_{12}$.

As can be seen, the dynamics of the ECLM is defined by the parameter vector (29). To identify this vector, in a real vehicle, time series would be recorded in which an experienced captain performs obstacle avoidance maneuvers with the USV (logging the vessel's course, linear and angular velocities). Thus, the ECLM takes into account the dynamics of the USV during collision risk situations.

$$\boldsymbol{\Theta_{ECLM}} = \begin{pmatrix} \tau_U & d_U & \zeta_U & \zeta_\chi & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 \end{pmatrix} \tag{29}$$

Since we do not have a real vessel, the RRSOAS algorithm itself has been used as an experienced pattern to generate the time series that will be used in order to identify the model (28). In these

simulations, the predictions used by the RRSOAS are made from the models (1), (3) and (5), which have been discretized with the forward Euler method and an integration step $T_p = 0.1$ s.

Moreover, as the ECLM is a system with delays, in line with the work carried out in [70–72], a numerical optimization method based on a genetic algorithm (GA, [73]) is used for the identification of the vector (29). In this way, each $\boldsymbol{\Theta_{ECLM}}$ is considered as an individual of a population, and consequently, each parameter represents a gene. In addition, to quantify the fit of the model (28) to the time series, the SAE (sum absolute errors) indicators are used:

$$SAE_\chi = \sum_{k=1}^{n_{ts}} \left| \chi(k) - \tilde{\chi}(k) \right|, \qquad SAE_r = \sum_{k=1}^{n_{ts}} \left| r(k) - \tilde{r}(k) \right|, \qquad SAE_U = \sum_{k=1}^{n_{ts}} \left| U(k) - \tilde{U}(k) \right| \tag{30}$$

where $n_{ts} = T_t / T_p$ is an integer that represents the length of the time series, which are obtained in navigation trials, and T_t is its duration.

Accordingly, the cost function used by the GA to calculate the fit of each individual is defined as:

$$J_{ECLM} = SAE_\chi / \pi + SAE_r / r_{max} + SAE_U / U_{lim} \tag{31}$$

Due to the random nature of evolutionary algorithms [73], the identification for the same time series has been repeated 32 times, and from the parameter vectors obtained, the one that best fits a second validation time series was chosen; see Figure 8.

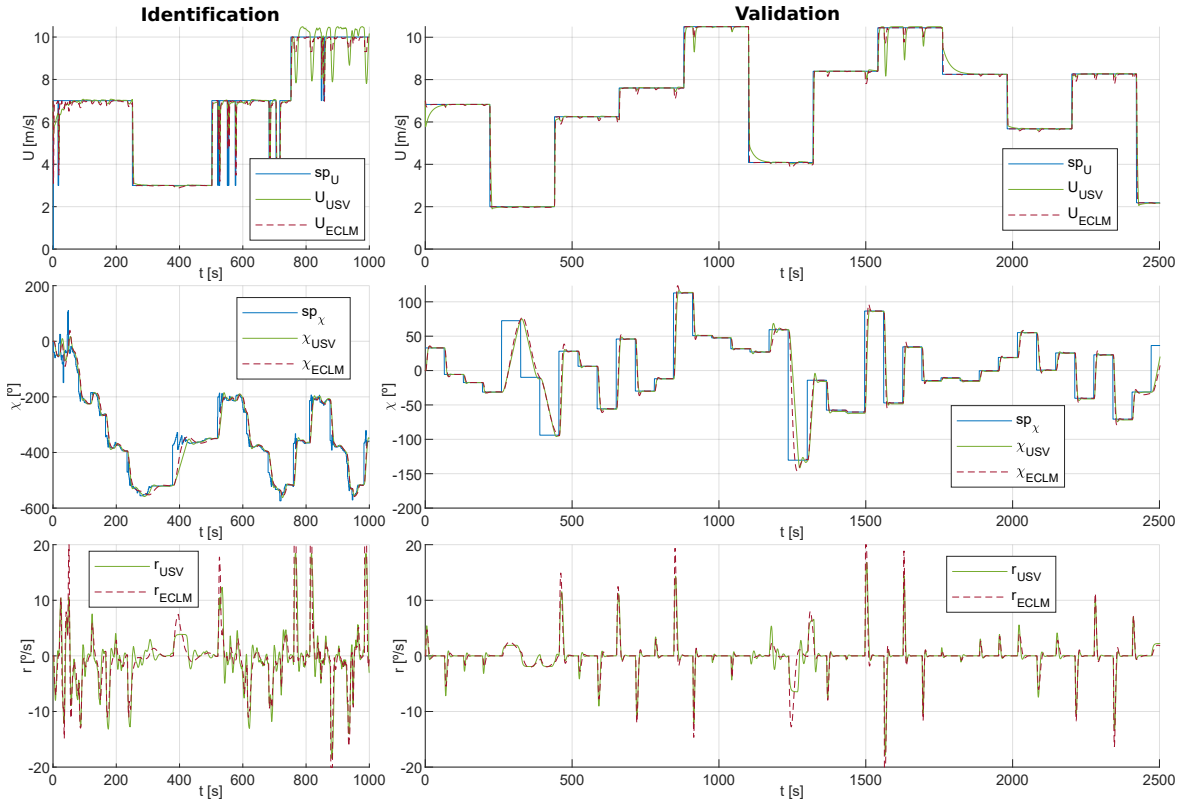

Figure 8. Time series obtained through numerical simulations with the unmanned surface vehicle (USV), Equations (1), (3) and (5), versus ECLM (28). Two time series: identification and validation.

To implement the GA, Toolbox within Matlab was used [74]. In order to ensure the stability of the model and to limit the search space, the parameters of the vector $\boldsymbol{\Theta_{ECLM}}$ were restricted while taking into account its physical meaning and the USV's behavior. Consequently, the option *mutationadaptfeasible* was set as a mutation function [74] and the initial population was also limited. In addition, we allowed: a population of 200 individuals, a maximum of 100 generations and the

parameter *MaxStallGenerations* was established as 15% of the maximum of generations. Finally, to avoid early convergence [73,74], elite individuals were limited to 2% of the population and the crossover factor was reduced (increasing the mutation factor) to 0.7. The other parameters were the ones established by default in [74]. As a result, the parameters identified for the ECLM, which estimates the dynamics of the closed-loop system formed by Equations (1), (3) and (5), were as follows:

$$\tilde{\boldsymbol{\Theta}}_{\boldsymbol{ECLM}} = \begin{pmatrix} 0.7 & 0.2 & 0.9 & 0.6 & 0.0002 & 0.0003 & 0.015 & 0.1 & 15.6 & 49 & 0.8 & 5.7 \end{pmatrix} \tag{32}$$

3.2.2. Path Predictor with Variable Prediction Horizon

The path predictor proposed in this work uses a variable prediction horizon M^{i_c} to ensure that all future paths cover the same prediction distance (D_p). In this way, the prediction time is adapted to each pair of $\boldsymbol{sp}^{\boldsymbol{i_c}}$ setpoints as a function of USV's dynamics. In particular, we propose to set D_p according to the measurement range of the sensors (d_{range}). Furthermore, if this measurement range is oversized due to the maneuverability of the USV to avoid static obstacles, the distance D_p can be reduced as long as it ensures that the course and speed of the ECLM (28) reaches the stationary regime.

From the current state of the vessel, the ECLM status vector is obtained to initialize the predictions:

$$\boldsymbol{x}_{\boldsymbol{ECLM}}(k) = \begin{pmatrix} \tilde{a}_l(k) & \tilde{U}_l(k) & U(k) & r(k) & \chi(k) & x_E(k) & y_E(k) \end{pmatrix}^T \tag{33}$$

where $\tilde{U}_l(k)$ is obtained according to the Equation (24) and $\tilde{a}_l(k)$ is calculated using Euler's forward approximation:

$$\tilde{U}_l(k) = U(k) + |r(k)| \left(c_1 U^2(k) + c_2 U(k) + c_3 \right), \qquad \tilde{a}_l(k) = \frac{U(k) - U(k-1)}{T_m} \tag{34}$$

In addition, due to the delays defined in the ECLM (28), the control setpoints demanded by the controllers in previous periods are recorded:

$$\begin{aligned} \boldsymbol{sp}_{\boldsymbol{\chi}}^{\boldsymbol{past}} &= \begin{pmatrix} sp_\chi(k-1) & \dots & sp_\chi(k-\gamma_2^b) \end{pmatrix} \\ \boldsymbol{sp}_{\boldsymbol{U}}^{\boldsymbol{past}} &= \begin{pmatrix} sp_U(k-1) & \dots & sp_U(k-\gamma_1) \end{pmatrix} \end{aligned}, \qquad \gamma_2^b = \left\lceil \frac{c7 + c_8 / U_{gov}}{T_p} \right\rceil + 1 \tag{35}$$

where γ_2^b is the maximum delay taken into account for the estimation of course dynamics. This delay is obtained at the minimum government velocity U_{gov}; see Equations (24) and (25).

With the initial state (33) and previous ECLM entries (35) established, each pair of alternative government setpoints is held for the entire prediction horizon. As a result, by carrying out iterative calculations with the ECLM (28), an estimated future path is obtained for each pair of $\boldsymbol{sp}^{\boldsymbol{i_c}}$,

$$\boldsymbol{Y}^{\boldsymbol{i_c}}(k) = \begin{pmatrix} \boldsymbol{y}_{\boldsymbol{ECLM}}(k+1) & \dots & \boldsymbol{y}_{\boldsymbol{ECLM}}(k+m^{i_c}) & \dots & \boldsymbol{y}_{\boldsymbol{ECLM}}(k+M^{i_c}) \end{pmatrix}, \quad m^{i_c} \in [1,\ M^{i_c}] \tag{36}$$

According to Equation (37), each horizon M^{i_c} must ensure that the path reaches the prediction distance D_p, which has been set equal to the measurement range of the LIDAR sensor (7). As an exception, if a zero velocity setpoint is evaluated, M^{i_c} is truncated when U_{gov} is reached. In a real hazardous situation, when a stop command is demanded, it will be necessary to switch the USV to dynamic positioning mode, or if it is not available, manual/emergency mode.

$$M^{i_c} = \begin{cases} m^{i_c} & \text{if } \tilde{d}(k+m^{i_c}) \geq D_p \\ M_{max} & \text{otherwise} \end{cases} \tag{37}$$

where $\tilde{d}(k+m^{i_c})$ is the distance traveled in each prediction step and M_{max} represents the maximum prediction horizon allowed.

With the aim of characterizing each possible future path, the distance traveled $\tilde{d}(k+m^{i_c})$ and the prediction time $\tilde{t}(k+m^{i_c})$ are calculated. For that purpose, the vector (36) was extended:

$$\boldsymbol{Z^{i_c}}(k) = \begin{pmatrix} \boldsymbol{Y^{i_c}}(k) \\ \boldsymbol{D^{i_c}}(k) \\ \boldsymbol{T^{i_c}}(k) \end{pmatrix}, \quad \begin{aligned} \boldsymbol{D^{i_c}}(k) &= \begin{pmatrix} \tilde{d}(k+1) & \dots & \tilde{d}(k+m^{i_c}) & \dots & \tilde{d}(k+M^{i_c}) \end{pmatrix} \\ \boldsymbol{T^{i_c}}(k) &= \begin{pmatrix} \tilde{t}(k+1) & \dots & \tilde{t}(k+m^{i_c}) & \dots & \tilde{t}(k+M^{i_c}) \end{pmatrix} \end{aligned} \tag{38}$$

By grouping the vectors $\boldsymbol{Z^{i_c}}$, the possible future paths for the entire DDS (20) are obtained:

$$\boldsymbol{Z^{DDS}}(k) = \begin{pmatrix} \boldsymbol{Z^{1}}(k) & \dots & \boldsymbol{Z^{i_c}}(k) & \dots & \boldsymbol{Z^{n_c}}(k) \end{pmatrix} \tag{39}$$

3.3. Robust Collision Checking System through Occupancy Grid

In this work, two novel contributions have been developed to increase the robustness of OA methods applied to USVs [16,17,34,36,39,40,44,46]. The first one takes into account the uncertainty present in the mathematical modeling of the USV that is used to make predictions. The second contribution deals with the measurement/estimation errors contained in the environmental model.

3.3.1. Variable Shape as a Function of the Prediction Step

Due to the recursive character of the predictions, parametric and structural uncertainties present in mathematical modeling [66] cause prediction errors that increase with each step (m^{i_c}). In order to take into account these errors, as well as the shape of the vessel, we propose to model the USV's contour as an ellipse whose perimeter varies as a function of m^{i_c}. For this purpose, the shape of the vessel is approximated to a discrete ellipse referenced to body-axis system; see Equation (40). In this ellipse, the major and minor diagonals are sized according to the length and the beam of the USV (L_{USV} and M_{USV}). On the other hand, the number of points used to discretize the ellipse, n_ϵ, can be calculated according to an approximation of its perimeter (*Ramanujan II-Cantrell*, [75]) and the resolution of the occupation grid (g_{res}). However, with the aim of reducing the algorithm's run time, n_ϵ can be tuned based on the computational capacity of the hardware employed.

$$\boldsymbol{\epsilon_B} = \begin{pmatrix} x^1_{\epsilon B} & y^1_{\epsilon B} \\ \vdots & \vdots \\ x^{i_\epsilon}_{\epsilon B} & y^{i_\epsilon}_{\epsilon B} \\ \vdots & \vdots \\ x^{n_\epsilon}_{\epsilon B} & y^{n_\epsilon}_{\epsilon B} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} cos\,(res_\epsilon) & sin\,(res_\epsilon) \\ \vdots & \vdots \\ cos\,(i_\epsilon res_\epsilon) & sin\,(i_\epsilon res_\epsilon) \\ \vdots & \vdots \\ cos\,(n_\epsilon res_\epsilon) & sin\,(n_\epsilon res_\epsilon) \end{pmatrix} \begin{pmatrix} \gamma_L L_{USV} & 0 \\ 0 & \gamma_M M_{USV} \end{pmatrix}, \quad \begin{aligned} i_\epsilon &\in [1, n_\epsilon] \\ res_\epsilon &= 2\pi/n_\epsilon \end{aligned} \tag{40}$$

where γ_L and γ_M are safety factors used to oversize the vessel's shape.

The points $\boldsymbol{\epsilon_B}$ are translated to the reference system of the occupancy grid or local-axis system,

$$\boldsymbol{\epsilon_L}(k+m^{i_c}) = \boldsymbol{\epsilon_B} \begin{pmatrix} cos\left(\tilde{\chi}(k+m^{i_c})\right) & sin\left(\tilde{\chi}(k+m^{i_c})\right) \\ -sin\left(\tilde{\chi}(k+m^{i_c})\right) & cos\left(\tilde{\chi}(k+m^{i_c})\right) \end{pmatrix} + \boldsymbol{M_{ones}} \boldsymbol{d_L}(k+m^{i_c}) \tag{41}$$

where $\boldsymbol{M_{ones}}$ represents a unit matrix of dimension $n_\epsilon \times 2$, while $\boldsymbol{d_L}$ is defined as:

$$\boldsymbol{d_L}(k+m^{i_c}) = \boldsymbol{diag}\left(\tilde{x}_E(k+m^{i_c}) - x_E(k),\ \tilde{y}_E(k+m^{i_c}) - y_E(k)\right) \tag{42}$$

To consider the prediction errors due to the uncertainty present in the model, the constant dimension ellipse used in Equation (41) is replaced by an ellipse whose perimeter increases with m^{i_c}; see Equation (43). Moreover, to model and limit the growth of this ellipse, it is proposed to

use a hyperbolic tangent (function with continuous bounds). As a visual example, Figure 9 shows a comparative of the ellipses $\boldsymbol{\epsilon}_{\boldsymbol{B}}^{v}$ obtained when γ_v is modified.

$$\boldsymbol{\epsilon}_{\boldsymbol{B}}^{v}(k+m^{i_c}) = \boldsymbol{\epsilon}_{\boldsymbol{B}}\left(1+tanh\left(\frac{m^{i_c}}{M^{i_c}}\right)\gamma_v\right) \tag{43}$$

where γ_v is defined positive and represents a variable oversize factor, which is established according to the degree of fit between the prediction model and the real behavior of the USV.

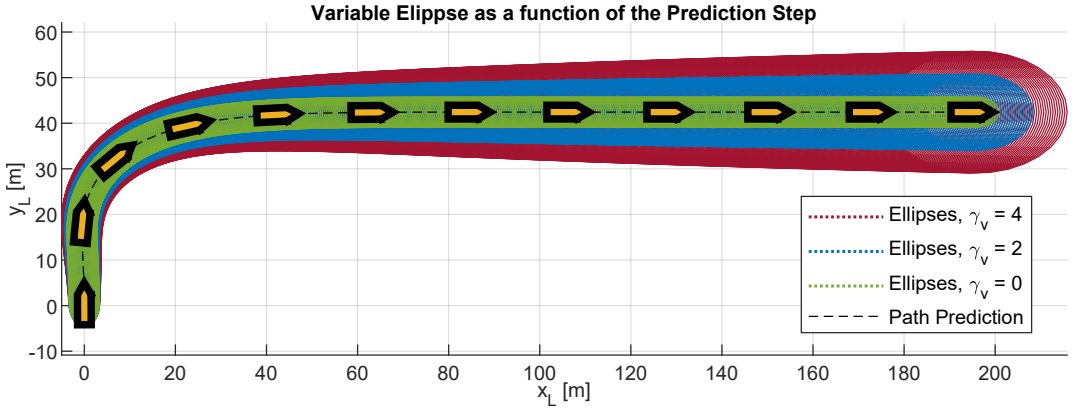

Figure 9. Variable ellipses obtained for a prediction horizon M^{i_c}; $n_\epsilon = 32$ has been set.

3.3.2. New Repulsive Forces and Estimated Collision Times Method

In order to translate the possible future paths to the occupancy probability grid ($\boldsymbol{g_p}$), at each prediction step, the points of the discrete ellipse (41) are expressed as cells:

$$\boldsymbol{\epsilon_{ij}}(k+m^{i_c}) = \begin{pmatrix} i_g^1 & \dots & i_g^{i_\epsilon} & \dots & i_g^{n_\epsilon} \\ j_g^1 & \dots & j_g^{i_\epsilon} & \dots & j_g^{n_\epsilon} \end{pmatrix}^T = \left\lfloor \boldsymbol{\epsilon_L}(k+m^{i_c}) \begin{pmatrix} -1/g_{res} & 0 \\ 0 & 1/g_{res} \end{pmatrix} \right\rceil + b_g \boldsymbol{M_{ones}} \tag{44}$$

where $\lfloor x \rceil$ represents the function *round(x)* and $b_g = (1+n_g)/2$.

From the probability of occupation stored in the cells of $\boldsymbol{\epsilon_{ij}}$ and $\boldsymbol{p_\epsilon}$, the maximum probability is obtained for each step of the prediction (p_{max}); see Equation (45). Then, by grouping $p_{max}(k+m^{i_c})$ for the entire prediction horizon, the vector that contains the maximum occupancy probabilities is formed for each possible future path; see Equation (46).

$$p_{max}(k+m^{i_c}) = max\left(\boldsymbol{p_\epsilon}(k+m^{i_c})\right), \qquad \boldsymbol{p_\epsilon}(k+m^{i_c}) = \begin{pmatrix} p_{ij}^1 & \dots & p_{ij}^{i_\epsilon} & \dots & p_{ij}^{n_\epsilon} \end{pmatrix} \tag{45}$$

$$\boldsymbol{P_{max}^{i_c}}(k) = \begin{pmatrix} p_{max}(k+1) & \dots & p_{max}(k+m^{i_c}) & \dots & p_{max}(k+M^{i_c}) \end{pmatrix} \tag{46}$$

Now, the concept of repulsive forces (f_r) from the VFF (virtual force field, [29]) method is used. In this method, all cells of the grid produce a f_r depending on their current distance to the vehicle and their probability of occupancy. Then, these repulsive forces are added, together with a attraction force, to obtain a resultant guidance force. As a novel contribution, in this work the repulsive forces (47) are used to characterize each possible future path $\boldsymbol{Z^{i_c}}$. For this purpose, at each prediction step m^{i_c}, the force $f_r(k+m^{i_c})$ is calculated as a function of the future distance $\tilde{d}(k+m^{i_c})$ and the maximum probability of the cells crossed by the discrete ellipse (44); $p_{max}(k+m^{i_c})$.

$$\boldsymbol{f_r^{i_c}}(k) = \begin{pmatrix} f_r(k+1) & \dots & f_r(k+m^{i_c}) & \dots & f_r(k+M^{i_c}) \end{pmatrix}, \qquad f_r(k+m^{i_c}) = \frac{p_{max}(k+m^{i_c})}{\tilde{d}^{n_d}(k+m^{i_c})} \tag{47}$$

where n_d defines the variation of the repulsive force as a function of the future distance. In the VFF method [29], the authors set $n_d = 2$. Specifically, they carry out the autonomous guidance of the CARMEL mobile robot, which is equipped with 24 ultrasounds whose detection range covers 3 m. Since this measurement range is much lower than the one used in USVs [1–3,13,31,41,67], it is necessary to study the variation of repulsive forces as a function of the parameter n_d. To this end, Figure 10 shows the repulsive forces obtained, over the measurement range used in this work ($d_{range} = 200$ m), for three different values of n_d. Due to the variation shown in this figure, and with the reason that all obstacles located within d_{range} produce a significant repulsive force, in this work $n_d = 0.5$ has been set.

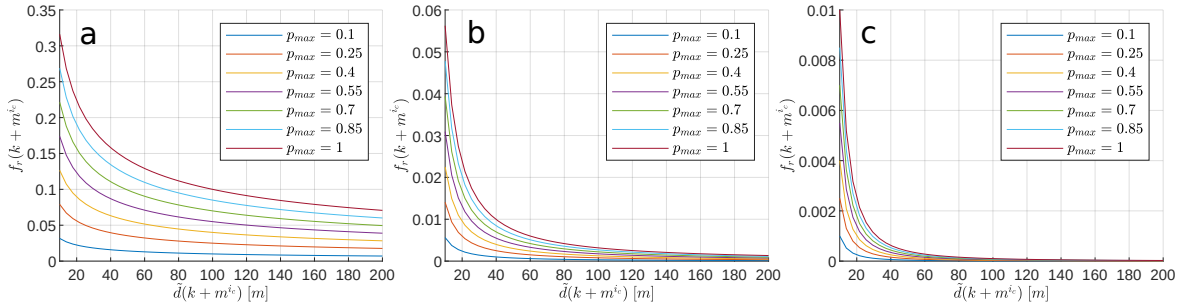

Figure 10. Repulsive forces as a function of possible future distance, maximum probability of occupancy and different values of n_d; (**a**) $n_d = 0.5$, (**b**) $n_d = 1.25$ and (**c**) $n_d = 2$.

On the other hand, in order to constrain the setpoints $\boldsymbol{sp}^{i_c}$ that would lead to an immediate collision, we propose to estimate the collision times for each possible future path; $T_{ec}^{i_c}$. For this purpose, at each prediction step m^{i_c}, $t_{ec}(k+m^{i_c})$ is calculated as a function of the prediction time $\tilde{t}(k+m^{i_c})$ and the maximum probability $p_{max}(k+m^{i_c})$; see Equation (48). Then, by grouping $t_{ec}(k+m^{i_c})$ for the entire prediction horizon M^{i_c}, the vector (49) is formed.

$$t_{ec}(k+m^{i_c}) = \begin{cases} \dfrac{\tilde{t}(k+m^{i_c})}{p_{max}^{n_t}(k+m^{i_c})} & \text{if } p_{max}(k+m^{i_c}) > 0 \\ M_{max} T_P & otherwise \end{cases}, \qquad n_t \in (0,1] \tag{48}$$

where n_t is an parameter used to oversize the occupancy probability, and consequently, decrease the estimated collision time.

$$\boldsymbol{t}_{\boldsymbol{ec}}^{\boldsymbol{i_c}}(k) = \begin{pmatrix} t_{ec}(k+1) & \dots & t_{ec}(k+m^{i_c}) & \dots & t_{ec}(k+M^{i_c}) \end{pmatrix} \tag{49}$$

Finally, looking for the worst case, each $\boldsymbol{sp}^{\boldsymbol{i_c}}$ is defined by the highest repulsive force ($F_R^{i_c}$) and the lowest estimated collision time ($T_{EC}^{i_c}$). Thus, once all alternative government setpoints have been evaluated, the DDS (20) is characterized by the prediction model and the occupancy probability grid:

$$\begin{aligned} \boldsymbol{F}_{\boldsymbol{R}}^{\boldsymbol{DDS}}(k) &= \begin{pmatrix} F_R^1(k) & \dots & F_R^{i_c}(k) & \dots & F_R^{n_c}(k) \end{pmatrix}, & F_R^{i_c}(k) &= max\left(\boldsymbol{f}_{\boldsymbol{r}}^{\boldsymbol{i_c}}(k)\right) \\ \boldsymbol{T}_{\boldsymbol{EC}}^{\boldsymbol{DDS}}(k) &= \begin{pmatrix} T_{EC}^1(k) & \dots & T_{EC}^{i_c}(k) & \dots & T_{EC}^{n_c}(k) \end{pmatrix}, & T_{EC}^{i_c}(k) &= min\left(\boldsymbol{t}_{\boldsymbol{ec}}^{\boldsymbol{i_c}}(k)\right) \end{aligned} \tag{50}$$

3.4. Guidance and Obstacle Avoidance Heuristic System

In line with most reactive obstacles avoidance methods [12], a heuristic is minimized to decide the setpoints $\boldsymbol{sp}(sp_\chi, sp_U)$ demanded by the controllers at each sample time (k). In particular, this heuristic

is a modification of the one proposed for the dynamic windows method [27], with the purpose of improving its performance in USVs.

$$\begin{aligned} &\boldsymbol{J_{RRSOAS}}(k) = \alpha_1 \boldsymbol{J_{Heading}}(k) + \alpha_2 \boldsymbol{J_{Velocity}}(k) + \alpha_3 \boldsymbol{J_{FR}}(k) + \alpha_4 \boldsymbol{J_{Past}}(k) \\ &T_{mac} \boldsymbol{M_{ones}} \leq \boldsymbol{T_{EC}^{DDS}}(k) \end{aligned} \tag{51}$$

where α_i, for $i \in \{1, 2, 3, 4\}$, represent the tuning parameters used in the heuristic weighting, $\boldsymbol{M_{ones}}$ is a unit matrix of dimension $1 \times n_c$, and as in [27], T_{mac} is a parameter used to restrict the setpoints $\boldsymbol{sp^{i_c}}$ according to their estimated collision times. In addition, in order to facilitate the tuning of the algorithm, the indeces taken into account in the heuristics are normalized.

First, based on the proposal made in [27], the indices $\boldsymbol{J_{Heading}}$ and $\boldsymbol{J_{Velocity}}$ are obtained, which define the alignment of the DDS (20) with the goal course and velocity. Moreover, in line with the proposal initially made in [30], and later, used in other OA systems for USVs [16,17], the index $\boldsymbol{J_{Past}}$ is added to take into account the alignment of each $sp_\chi^{i_c}(k)$ with the previous course setpoints $sp_\chi^{i_c}(k-1)$. In this way, the algorithm has a memory effect that reduces commutations in the course setpoints. These commutations, due to small variations in the heuristics between different execution periods, cause indecisive behavior in the USV's guidance that can lead to a collision [30]. In particular, by expressing sets (17) and (18) in vector form:

$$\begin{aligned} \boldsymbol{SP_\chi^{DDS}}(k) &= \begin{pmatrix} sp_\chi^1(k) & \dots & sp_\chi^{i_c}(k) & \dots & sp_\chi^{n_c}(k) \end{pmatrix} \\ \boldsymbol{SP_U^{DDS}}(k) &= \begin{pmatrix} sp_U^1(k) & \dots & sp_U^{i_c}(k) & \dots & sp_U^{n_c}(k) \end{pmatrix} \end{aligned} \tag{52}$$

From the vectors (52), the previous course setpoint $sp_\chi(k-1)$ and the goal setpoints (χ_{goal}, U_{goal}), the indices $J_{Heading}$, J_{Past} and $J_{Velocity}$ are obtained:

$$\begin{aligned} &\boldsymbol{J_{Heading}}(k) = \frac{1}{\pi} \Delta \left(\boldsymbol{SP_\chi^{DDS}}(k),\ \chi_{goal}(k) \boldsymbol{M_{ones}} \right), \quad \boldsymbol{J_{Past}}(k) = \frac{1}{\pi} \Delta \left(\boldsymbol{SP_\chi^{DDS}}(k),\ sp_\chi(k-1) \boldsymbol{M_{ones}} \right) \\ &\boldsymbol{J_{Velocity}}(k) = \frac{1}{\Upsilon_U^{max}(k)} \boldsymbol{\Upsilon_U}(k), \quad \Upsilon_U^{max}(k) = max\left(\boldsymbol{\Upsilon_U}(k) \right), \quad \boldsymbol{\Upsilon_U}(k) = \left| \boldsymbol{SP_U^{DDS}}(k) - U_{goal}(k) \boldsymbol{M_{ones}} \right| \end{aligned} \tag{53}$$

where Δ and $|\ |$ calculate the absolute angular distance and the absolute value, respectively.

On the other hand, instead of weighting the collision distances [16,17,27,40], in this work the index $\boldsymbol{J_{FR}}$ (54) is employed. Thus, when weighting the DDS (20), the occupancy probability of the space is considered through the repulsive forces that characterize each possible future path.

$$\boldsymbol{J_{FR}}(k) = \frac{1}{F_R^{max}(k)} \boldsymbol{F_R^{DDS}}(k), \qquad F_R^{max}(k) = max\left(\boldsymbol{F_R^{DDS}(k)} \right) \tag{54}$$

Finally, if the DDS (20) is empty, which means that all alternative government setpoints have been restricted by the heuristics (51), the RRSOAS will demand a stop command. Consequently, the USV will switch to manual/emergency mode or, if a dynamic positioning control is available [4,5], the yaw of the vessel could be modified to restart the mission in a safe manner.

4. Simulation Results

In this first stage of development, like other reactive algorithms designed for USVs [14,16,35–37,45,47], the RRSOAS was evaluated by means of numerical simulations. The Runge–Kutta numerical integration method of fourth order was used with an integration step $T_s = 0.01$ s. As USV, the set formed by the following models has been used: vessel (1), actuators (3) and course/velocity controllers (5). In addition, the information of the environment that surrounds the USV was generated by the LIDAR sensor model (7) and translated to an occupancy probability grid

by the Bayesian filter (15). In turn, the reactive algorithm ran with a sample time $T_m = 1$ s. As a visual support, Figure 11 shows an instant during RRSOAS execution. On the other hand, each parameter of the algorithm has been adjusted, taking into account its physical meaning, in an iterative process in scenario 3; see Figure 3. As a result, Table 1 was obtained. Scenario 3 was chosen to tune the RRSOAS due to the work previously done in [31], where several SOA methods were tuned on this scenario; and we used the guidance of the USV (1) based on the same environmental information (8). Thus, the RRSOAS can be compared with other SOA methods [11,30,31,41] under the same conditions.

Table 1. RRSOAS parameters in international units, which have been tuned in scenario 3 of Figure 3.

Navigation Behaviour		Safety and Sizing		Path Predictor	Discrete Decision Space	
$\alpha_1 = 0.5$	$\alpha_2 = 0.3$	$T_{mac} = 20$	$\gamma_L = 1.25$	$D_p = 200$	$n_U = 1$	$U_{lim} = 10$
$\alpha_3 = 0.7$	$\alpha_4 = 0.25$	$\gamma_v = 3.4$	$\gamma_M = 2.45$	$T_p = 0.1$	$n_\chi = 9$	$\Delta\chi_{max} = \pi/2$
		$n_t = 0.75$	$n_\epsilon = 32$	$M_{max} = 1100$	$\tau_\chi = 2.2$	$\kappa_\chi = 0.1$

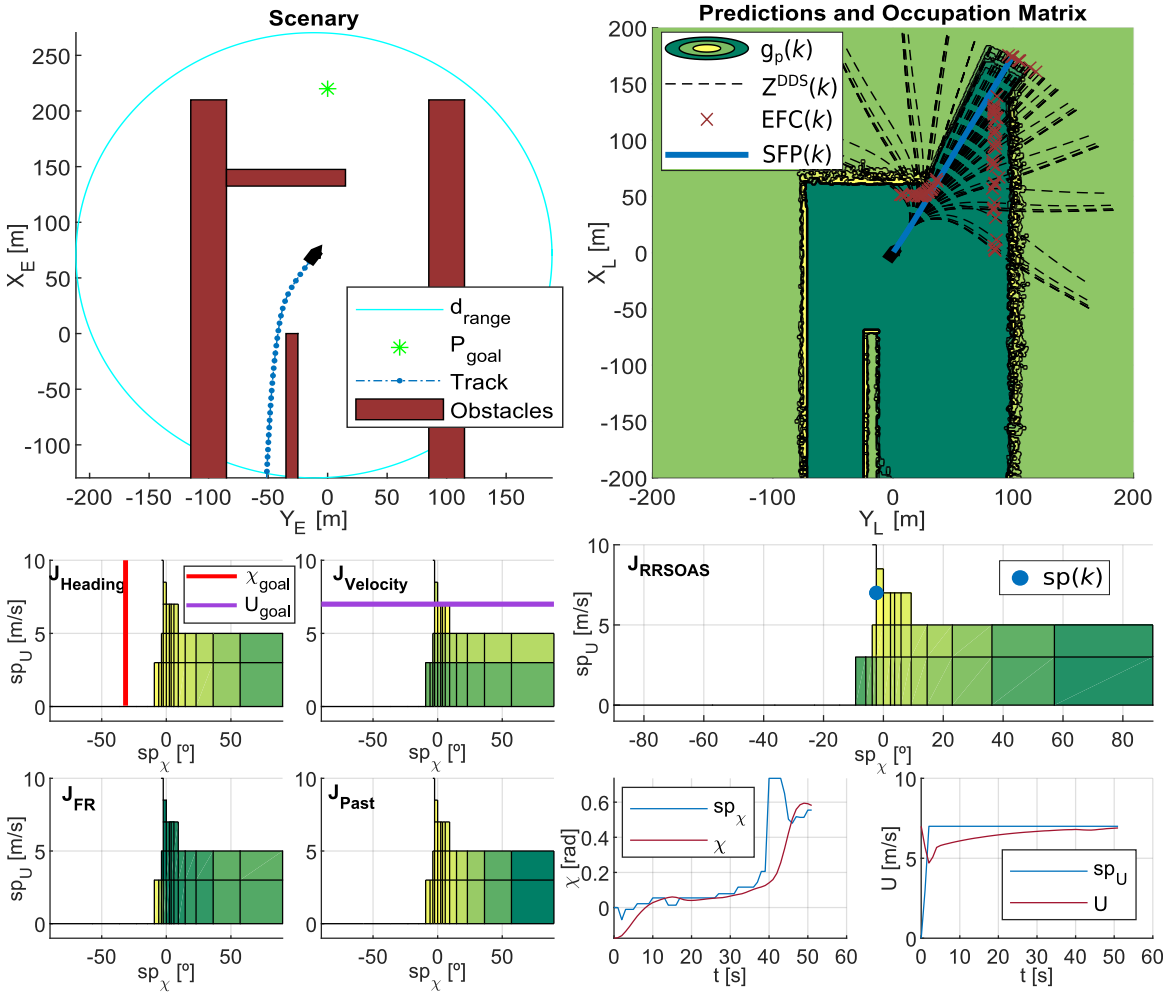

Figure 11. A RRSOAS execution period: evolution of the course/velocity setpoints, USV path, translation of the predictions to the occupancy probability grid (estimated future collisions (EFC) and selected future path (SFP)) and Heuristics (51).

4.1. Performances Analysis

First, the performance of the RRSOAS was evaluated when it used the new ECLM (28) as an estimator of future paths. To that end, this performance was compared with the one that would

be obtained if, when making the predictions, the USV's dynamics was fully known. This analysis would not be possible in real sea trials, since all mathematical modeling presents different kinds of uncertainties that deviate it from the real system's behavior [66]. However, in numerical simulations, it is possible to use the same dynamic model that represents the behavior of the USV as a prediction model. This approach was used in [16,17,34,36,39,40,44,46], where the authors also validated OA systems through numerical simulations and the only uncertainties considered in the prediction model were the ones associated with the effects that external disturbances caused on the vessel. In line with these works, the USV model (Equations (1) and (3)–(5), USVM) was discretized by using the Euler forward method. Thus, the performance of the RRSOAS was compared with two different prediction models: the USVM and the ECLM; in other words, two reactive algorithms were compared: $\text{RRSOAS}_{\text{USVM}}$ and $\text{RRSOAS}_{\text{ECLM}}$. Specifically, this comparative was carried out over the five scenarios defined in [31], in which the vehicle started moving at the goal velocity $U_{goal} = u = 7$ m/s and the ocean current was set as $V_c = 1$ kn with direction $\beta_c = 0^{\text{o}}$. As tuning parameters, in both cases the ones listed in Table 1 were used. As a result, Figure 12 shows the paths obtained with the USV's guidance being carried out by the RRSOAS algorithm with each prediction model. As can be seen, in both cases the algorithm successfully guided the USV to $\boldsymbol{P_{goal}}$ in the presence of disturbances. Furthermore, with the exception of scenario 1, wherein heuristics (51) generated a different initial decision, the paths followed by the USV did not present significant changes. This is due to the fact that, independently of the fit between the prediction model and the behavior of the USV, RRSOAS takes into account that predictions of future paths will always contain accumulative errors caused by the uncertainty. This feature is modeled through the variable ellipse (43), whose perimeter represents the shape of the vessel; see Figure 9.

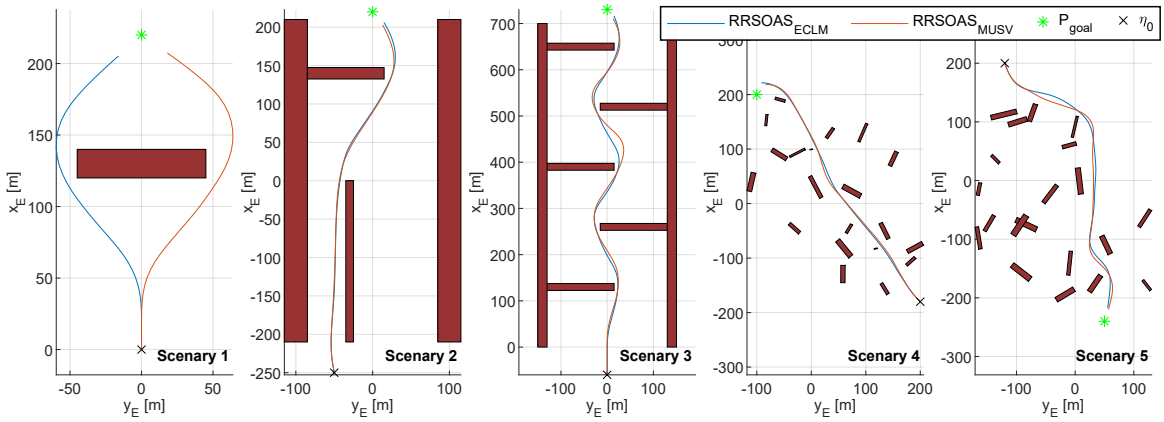

Figure 12. A comparative of the RRSOAS operating with two prediction models: ECLM (28) and USVM—(1), (3), (4) and (5). The adjustment parameters used can be found in Table 1.

In addition to the vessel's paths, this work quantitatively evaluated the performances in the guidance of the USV. These performances, in line with the approach followed in [31], are defined according to the following indicators: mission duration (t_m), distance covered during the mission (d_m) and control effort required (Δc); see Equation (55).

$$\begin{aligned} t_m(k) &= kT_m \\ d_m(k) &= \left\| \boldsymbol{\eta_I}(k) - \boldsymbol{\eta_I}(k-1) \right\| + d_m(k-1), \qquad \boldsymbol{\eta_I}(k) = \left(x_E(k),\, y_E(k) \right) \\ \Delta c(k) &= \frac{|sp_\chi(k) - sp_\chi(k-1)|}{\pi} + \frac{|sp_U(k) - sp_U(k-1)|}{U_{lim}} + \Delta c(k-1) \end{aligned} \tag{55}$$

These indicators have been obtained for the paths shown in Figure 12 and are collected in Table 2. As can be seen, the RRSOAS presents very similar indicators when guiding the USV with both prediction

models. Therefore, given the combination of the ECLM (28) with the variable ellipse (43), both novel proposals of this work can be used to carry out the autonomous guidance of vessels, whose mathematical models, and those of their controllers, are unknown. Hence, in all the results shown below, the ECLM was used as a prediction model (RRSOAS = $\text{RRSOAS}_{\text{ECLM}}$). Moreover, ECLM presents two advantages over the general approach [16,17,34,36,39,40,44,46]. Firstly, it does not depend on previous mathematical modeling of the vessel or its controllers. Secondly, it makes the simulation environment more realistic, since the dynamic model that represents the USV's behavior is not used as predictive model. Consequently, as would happen on a real USV, the estimation of future paths contemplates prediction errors.

Table 2. Indicators (55) achieved by RRSOAS algorithm with each prediction model in the scenarios proposed in [31]. These indicators characterize the paths shown in Figure 12.

Methods	Scenario 1			Scenario 2			Scenario 3			Scenario 4			Scenario 5		
	t_m	d_m	Δc	t_m	d_m	Δc	t_m	d_m	Δc	t_m	d_m	Δc	t_m	d_m	Δc
$\text{RRSOAS}_{\text{ECLM}}$	38	239	0.9	73	733	1.1	123	890	1.8	76	776	1.0	79	746	2.3
$\text{RRSOAS}_{\text{MUSV}}$	39	246	0.9	72	729	0.6	125	904	2.0	76	770	1.2	79	753	1.7

Additionally, here the RRSOAS is compared with two SOA methods. The first method is the LROABRA (local reactive obstacle avoidance based on region analysis) algorithm, which has been specifically designed to provide vessels with the capacity to avoid static obstacles at high navigation velocities [41]. The second SOA method considered is the VFH+, which was proposed in [30] as an improvement of the VFH [51], and later used as an OA system for a USV [11]. Specifically, to carry out the comparison, the results obtained in [31] are used, where both methods were autotuning with a genetic algorithm on scenario 3, and in turn, evaluated over the scenarios shown in Figure 3. These results, which are directly related to the indicators (55), have been included in Table 3 together with the performances obtained by the RRSOAS. Note that, according to the results obtained in [31], the LROABRA algorithm, automatically tuned for scenario 3, does not pass scenario 4. In particular, it can be seen how RRSOAS, with respect to the LROABRA and VFH+ methods, achieves improved mission times and reduced distances covered in most scenarios. As an exception, in scenario 4 the VFH+ method achieves slightly higher performance (it reduces mission time and distance by 1.32% and 3.23%, respectively). Therefore, for the USV (1), using the measurements provided by the LIDAR model (8), the new RRSOAS proposed in this paper offers competitive performance in autonomous guidance with respect to other SOA methods applied to USVs [11,30,41].

Table 3. Indicators (55) achieved by each obstacle avoidance method in the scenarios proposed in [31]. The results of the VHF+ [30,76] and LROABRA [41] methods have been obtained in [31].

Methods	Scenario 1			Scenario 2			Scenario 3			Scenario 4			Scenario 5		
	t_m	d_m	Δc	t_m	d_m	Δc	t_m	d_m	Δc	t_m	d_m	Δc	t_m	d_m	Δc
$\text{RRSOAS}_{\text{ECLM}}$	38	239	0.9	73	733	1.1	123	890	1.8	76	776	1.0	79	746	2.3
LROABRA [41]	69	346	2.4	82	752	0.9	145	929	2.2	—	—	—	88	758	1.5
VFH+ [30,76]	47	268	1.4	78	744	0.8	148	950	2.6	75	751	1.0	92	794	1.8

Continuing with the analysis of the RRSOAS, two new proposals have been presented in this paper with respect to the algorithm's runtime. The first one is focused on the effective discretization of the decision space (16), by using an exponential resolution (17) in the course setpoints. In addition, the prediction horizon (37) for each possible future path that the USV could follow is variable. Since the final objective of an OA system is its implementation in a real vehicle, it is necessary to ensure that its

computation is feasible between periods of execution of the same one. Due to the establishment times associated with the marine surface vehicle [5,57–60], sample times in reactive algorithms applied to USVs are generally higher than those used for other kinds autonomous vehicles [20,21,61,62]. As an example, the authors in [13,15,17,56] established sample times between 1 and 2.5 seconds for their reactive algorithms, which were evaluated with vessels similar to the USV (1) by means of numerical simulations in [13,15,17,56] and on real vessels in [13,17]. In other works [16,34], where the reactive algorithms were evaluated through numerical simulations, the sample times were 5 seconds (for a 32-meter vessel [34]). With regard to the RRSOAS, Figure 13 shows how the algorithm's runtime varies (average of 100 iterations) depending on the number of alternative government setpoints (n_c) and the number of points (n_ϵ) used to discretize ellipse (40). These times have been measured from MATLAB's *Tic-Toc* functions [77], while the algorithm is running in Simulink on a laptop model *msi GT60 2PC Dominator* [78]. As can be seen, for the nominal adjustment shown in Table 1 ($n_c = 76$ and $n_\epsilon = 32$), the average runtime is 0.092 s. Moreover, for the limits considered in this work to implement the algorithm ($n_c = 230$ and $n_\epsilon = 128$), an average runtime lower than the sample ($T_m = 1$ s) time is obtained (0.771 s). Therefore, if a hardware platform with similar computing power is used, it would be feasible to implement the new RRSOAS in a real vessel. In addition, it should be noted that these times could be improved if the algorithm ran on a hardware platform completely dedicated to it.

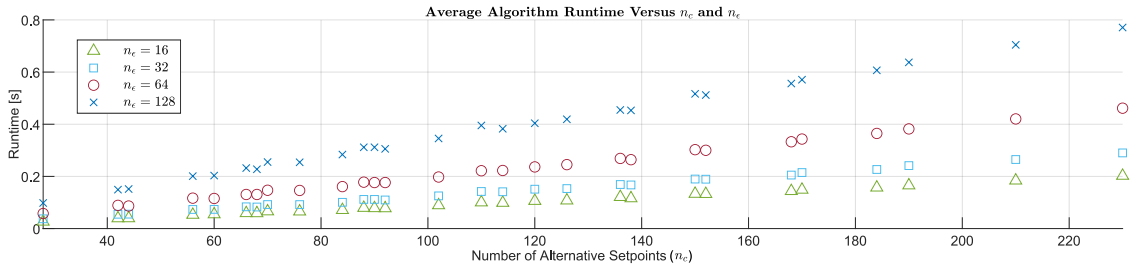

Figure 13. Variation of average RRSOAS runtimes as a function of the number of alternative government setpoints (n_c) and the number of points that model the shape of the USV (n_ϵ).

4.2. Robustness Analysis

Obstacle avoidance methods are adjusted on specific scenarios [12], which does not ensure that the OA system can overcome all situations [31]. This is due to the infinite geometric and environmental combinations that define the possible scenarios in which a USV could navigate. In this sense, as a novel contribution of this work, we propose to evaluate the robustness of the reactive algorithm on a sample of random scenarios. To this end, a sample of one-hundred scenarios has been generated following the procedure described in Section 2.3. $\Theta_E = (100\ 20\ 300\ 60\ 20)$ has been set as the characteristic vector of the sample. In addition, this sample is parameterized in terms of the goal velocity (U_{goal}) and the current velocity (V_c). Thus, it is possible to study the reactive algorithm in several USV's operation points, under the effect of different levels of environmental disturbances and the presence of random distributions of obstacles. Specifically, the sample of random scenarios is parameterized for nine different situations, which are obtained from the combination of $U_{goal} = \{5\ 7\ 9\}$ m/s and $V_c = \{0.5\ 1\ 2\}$ kn. Note that, by combining the above sets with the sample of one-hundred scenarios, the RRSOAS is evaluated over nine-hundred different scenarios. This study includes the following performance indicators: success rate, stop rate, collision rate, average mission time ($\overline{t_m}$), average distance traveled ($\overline{d_m}$) and average control effort ($\overline{\Delta c}$). In the success rate, the guidance is correct if the vehicle reaches $\boldsymbol{P_{goal}}$ without suffering any collision. During the simulations, the collision situation will occur if the distance between the USV and an obstacle is less than $L_{USV}/2$. On the other hand, the stop condition will occur if the reactive algorithm holds a zero velocity setpoint for a time $t_{stop} = 10$ s, whenever a collision situation has not previously occurred. This time has been set based on the braking response of the USV (1) in closed-loop (5). Finally, from the arithmetic mean of the indicators (55), the values ($\overline{t_m}$, $\overline{d_m}$, $\overline{\Delta c}$) are calculated for those scenarios where the USV is

successfully guided to $\boldsymbol{P_{goal}}$. In addition, as a complementary result of this work, a more conservative tuning of the RRSOAS has been done in order to show its capacity to increase the USV's safety. Therefore, the RRSOAS was studied with the performance tuning of Table 1; RRSOAS$^{\mathrm{PT}}$, and with the conservative tuning; RRSOAS$^{\mathrm{CT}}$. In this conservative tuning, with respect to the parameters listed in Table 1, the safety factors that oversize the vessel have been increased ($\gamma_L = 1.5$, $\gamma_M = 3$, $\gamma_v = 4.5$), the algorithm's behavior has been modified to avoid leading the USV towards occupied zones ($T_{mac} = 30$, $\alpha_1 = 0.4$, $\alpha_3 = 1$, $\alpha_4 = 0.2$) and the number of alternative velocity setpoints has been increased ($n_u = 2$). Note that, in both adjustments, the RRSOAS uses the ECLM (28) as the prediction model. Finally, in order to compare the robustness of RRSOAS with another reactive method, VFH+ is also evaluated in the same random scenarios. In particular, the VFH+ (version available in Simulink [76]) is chosen because it is the most robust of the SOA methods [30,35,41,42] studied in [31].

As results, Table 4 presents the indicators obtained by the reactive algorithms for each sample of scenarios parameterized in terms of U_{goal} and V_c. First, a comparative between VFH+ and RRSOAS$^{\mathrm{PT}}$ is made, taking into account that both have been adjusted on the same scenario. As can be seen, the two algorithms present similar success rates for $U_{goal} = 5$ m/s. However, as an advantage of the RRSOAS$^{\mathrm{PT}}$, its collision rate is much lower (0.67% versus 12%). On the other hand, for $U_{goal} = 7$ m/s and $U_{goal} = 9$ m/s, the RRSOAS$^{\mathrm{PT}}$ success rate is much higher than VFH+ (92.83% versus 83.17%), as is its capacity to avoid collisions (99% versus 91%). Moreover, in terms of average performance indicators, RRSOAS$^{\mathrm{PT}}$ outperforms VFH+ in most scenarios. As an exception, in scenarios where $U_{goal} = 9$ m/s, the distance traveled by the USV is less under the government of the VFH+. With regard to the RRSOAS$^{\mathrm{PT}}$ and RRSOAS$^{\mathrm{CT}}$ adjustments, a more conservative tuning increases the robustness of the algorithm and USV's safety, but causes a loss of performance in most cases.

Table 4. Indicators obtained by the reactive algorithms for each sample of one-hundred random scenarios parameterized in terms of U_{goal} and V_c.

Characterization of Random Scenarios	Algorithm	*Success* [%]	*Stop* [%]	*Collision* [%]	$\overline{t_m}$ [s]	$\overline{d_m}$ [m]	$\overline{\Delta c}$
$U_{goal} = 5$ m/s	$VHF+$	79	11	10	181.3	1247	2.36
	$RRSOAS^{PT}$	79	20	01	174.3	1234	1.93
$V_c = 0.5$ kn	$RRSOAS^{CT}$	95	05	00	178.6	1252	1.97
$U_{goal} = 5$ m/s	$VHF+$	72	13	15	181.2	1247	2.24
	$RRSOAS^{PT}$	73	26	01	173.6	1231	1.90
$V_c = 1$ kn	$RRSOAS^{CT}$	92	08	00	177.6	1248	1.92
$U_{goal} = 5$ m/s	$VHF+$	76	13	11	182.1	1249	2.30
	$RRSOAS^{PT}$	72	28	00	173.7	1231	1.90
$V_c = 2$ kn	$RRSOAS^{CT}$	88	22	00	178.1	1251	1.94
$U_{goal} = 7$ m/s	$VHF+$	85	08	07	131.7	1232	2.31
	$RRSOAS^{PT}$	91	18	01	128.1	1231	1.94
$V_c = 0.5$ kn	$RRSOAS^{CT}$	95	05	00	130.0	1235	1.84
$U_{goal} = 7$ m/s	$VHF+$	82	11	07	131.4	1231	2.30
	$RRSOAS^{PT}$	90	09	01	128.1	1231	1.96
$V_c = 1$ kn	$RRSOAS^{CT}$	97	03	00	131.1	1241	1.88

Table 4. *Cont.*

Characterization of Random Scenarios	Algorithm	*Success* [%]	*Stop* [%]	*Collision* [%]	$\overline{t_m}$ [s]	$\overline{d_m}$ [m]	$\overline{\Delta c}$
$U_{goal} = 7$ m/s	$VHF+$	86	10	04	132.1	1233	2.39
	$RRSOAS^{PT}$	90	09	01	127.5	1229	1.93
$V_c = 2$ kn	$RRSOAS^{CT}$	94	06	00	130.5	1241	1.88
$U_{goal} = 9$ m/s	$VHF+$	81	07	12	103.0	1209	2.29
	$RRSOAS^{PT}$	96	03	01	101.3	1221	1.82
$V_c = 0.5$ kn	$RRSOAS^{CT}$	96	04	00	103.8	1234	1.92
$U_{goal} = 9$ m/s	$VHF+$	82	06	12	103.4	1212	2.27
	$RRSOAS^{PT}$	95	04	01	101.5	1220	1.84
$V_c = 1$ kn	$RRSOAS^{CT}$	97	03	00	104.6	1239	1.96
$U_{goal} = 9$ m/s	$VHF+$	83	05	12	103.5	1211	2.32
	$RRSOAS^{PT}$	95	04	01	101.0	1221	1.82
$V_c = 2$ kn	$RRSOAS^{CT}$	95	05	00	105.1	1242	2.00

As a summary, Table 5 shows the average of all the indicators listed in Table 4. As can be seen, the RRSOAS outperforms all the average performances obtained by the VFH+. Furthermore, with a more conservative tuning, the success rate of the RRSOAS has been significantly increased (94.33% versus 86.78%), and over the nine-hundred scenarios analyzed, the vessel did not suffer any collisions. It should be noted that these results do not guarantee that the RRSOAS can overcome any static obstacle scenario. Although, these results demonstrate the viability of implementing and evaluating the new RRSOAS on a real USV.

Table 5. Average indicators for the nine-hundred scenarios shown in Table 4.

Algorithm	*Success* [%]	*Stop* [%]	*Collision* [%]	$\overline{t_m}$ [s]	$\overline{d_m}$ [m]	$\overline{\Delta c}$
$VHF+$	80.67	09.33	10.00	138.9	1230	2.31
$RRSOAS^{PT}$	86.78	12.33	00.89	134.4	1228	1.89
$RRSOAS^{CT}$	94.33	05.67	00.00	137.7	1243	1.92

Finally, as a visual report, Figure 14 shows the paths followed by the USV in sixteen of the nine-hundred scenarios analyzed in this paper. In these scenarios the USV was guided by the VFH+ and RRSOAS algorithms, where the RRSOAS was tuned in two different ways: conservative tuning and performance tuning.

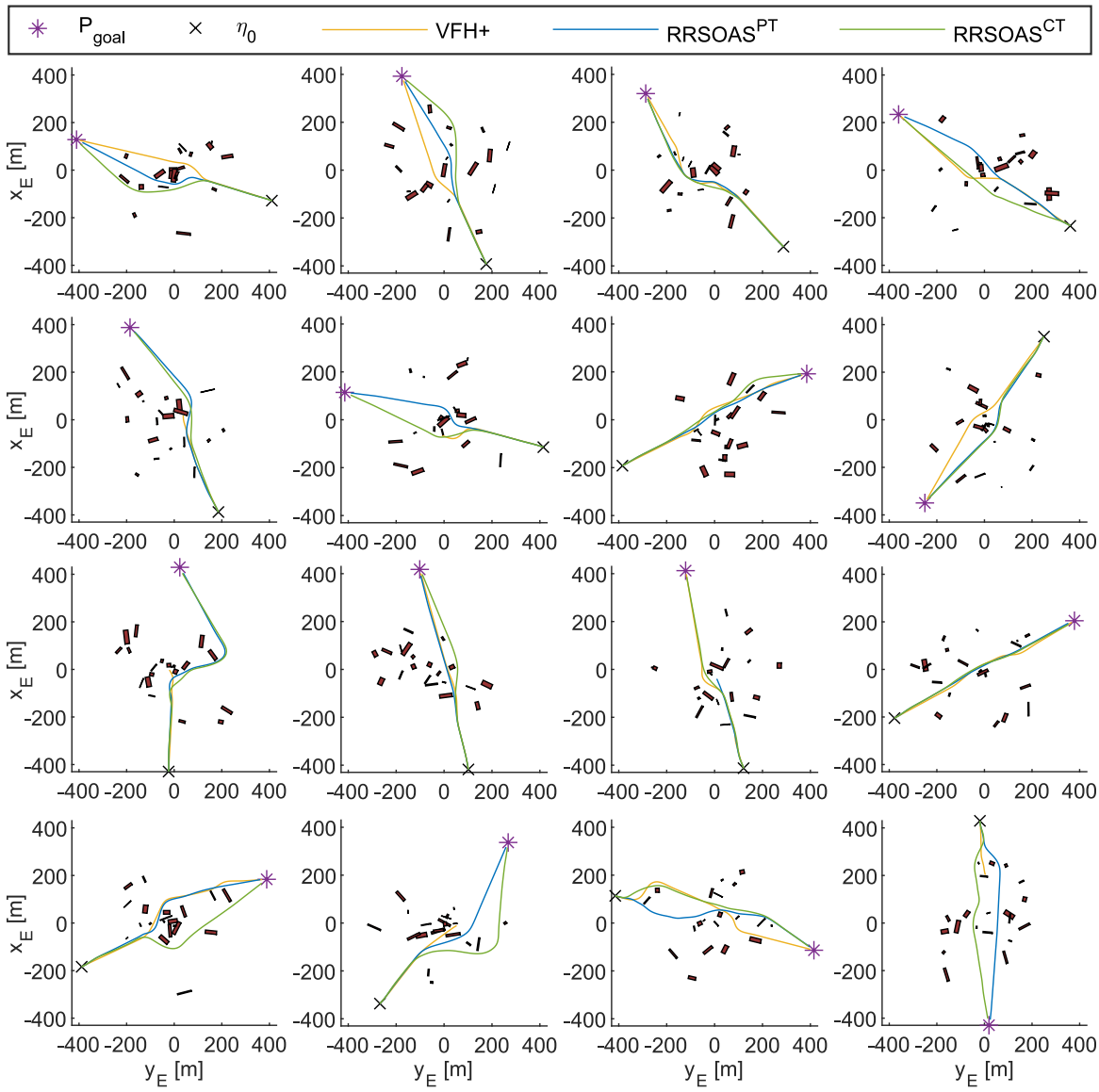

Figure 14. Paths of the vessel (1), in sixteen of the nine-hundred scenarios studied in this work, when it is guided autonomously by the algorithms RRSOAS and VFH+ [30,76].

5. General Discussion

Due to the current interest in the field of USVs, this paper proposes the new Robust Reactive Static Obstacle Avoidance System (RRSOAS) and used computing simulations to test our proposal. In terms of performance, a comparison of the new RRSOAS with other SOA methods applied to the USV has been carried out (VFH+ [11,30] and LROABRA [41]). The obtained results show how, in most cases, the RRSOAS outperforms these reactive algorithms. Furthermore, for several points of operation of the USV, a robustness study of the RRSOAS has been carried out on nine-hundred random scenarios in the presence of different current levels. In this aspect, the RRSOAS is significantly more robust than the VFH+, and with a conservative tuning, it guarantees the safety of the USV. In addition, the algorithm's runtime is considerably less than its sample time. Therefore, its implementation and evaluation in a real USV are feasible and proposed as future work. Moreover, taking into account these results, several of RRSOAS's contributions could be integrated into other reactive algorithms applied to USVs. Firstly, RRSOAS does not depend on a priori knowledge of a mathematical model of the vessel and its controllers for velocity and course. Instead, we propose to use an estimated closed-loop model (ECLM) of the USV, which is identified from time series in which an experienced pilot carries

out obstacle avoidance maneuvers. This ECLM could also be used by other reactive methods that make predictions of possible future paths of the vehicle [16,17,20,34,36,39,40,44,46], thus eliminating the dependence on USV's mathematical modeling that the use of such methods implies. In addition, with the aim of considering the prediction errors due to uncertainty, these works could integrate the variable ellipse with the prediction step proposed in this paper to model the USV's shape. On the other hand, an occupancy probability grid is used by RRSOAS. In this way, the reactive algorithm contemplates the errors and measurement uncertainty of the sensors. Another novel contribution has been made in this aspect; the repulsive forces proposed in [29] have been adapted in order to consider the USV's dynamics and the probability of occupation. These repulsive forces can also be applied to other OA systems for USVs based on potential fields [32,35,38,42,45]. Furthermore, in order to limit the algorithm's runtime, a variable horizon is proposed for the future path predictions, along with an exponential discretization to generate the alternative course setpoints. This last contribution of our method, given the successful results obtained in the numerical simulations, could be applied to other reactive algorithms which require discrete direction commands [13,14,16,17,26,27,30,41]. Finally, in order to take into account dynamic obstacles, the RRSOAS could easily be combined (since it does not need to know the model of the vessel or its controllers) with reactive algorithms designed to avoid dynamic obstacles according to the COLREGS regulation [13,14,16,36,39,40,46].

6. Conclusions

According to the results obtained in the numerical simulations carried out in this work, it is possible to conclude that the new RRSOAS is robust to unknown and congested scenarios in the presence of disturbances, while offering very competitive performance with respect to other reactive algorithms previously applied to USVs. In addition, based on the RRSOAS's runtimes, it could be implemented and evaluated in a real USV as a static obstacle avoidance system. Moreover, for its operation, this reactive algorithm only needs as inputs: the state vector of the USV, an occupancy probability grid and goal setpoints. In particular, the RRSOAS is the result of the integration of several novel contributions with regard to the current state-of-the-art of USVs that we proposed in this work. Firstly, the algorithm generates a new effective discrete decision space from an exponential resolution applied to the course setpoints; this resolution is proposed in our work. Secondly, a new estimated closed-loop model is proposed in this paper, which is used to make predictions of the possible future paths that the USV could follow if a collision risk were to arise. Thirdly, these paths are translated into an occupancy probability grid, where each path is characterized by taking into account the USV's dynamics, the uncertainty present in the prediction model and the occupancy probability. Finally, a heuristic system designed specifically for USVs is used to calculate the course and velociy setpoints. These setpoints are the RRSOAS's outputs, which are demanded by the USV's controllers.

Author Contributions: R.G. proposed most of the new contributions made in this article. Furthermore, he was responsible for implementing the simulation environment and the obstacle avoidance algorithms, and for carrying out the performance and robustness studies presented in the paper. In addition, R.G. wrote most of the work. M.J.L., J.S., A.G. and A.C. participated from the beginning in the work related to this article, in the proposals, analysis and discussion of results, in the approach followed, in the review and by way of contributions to the writing of the text. In addition, M.J.L. and J.S. wrote part of the paper and reviewed all its contents in depth. Finally, A.G. and A.C. reviewed the entire article and contributed to its translation. All authors have read and agreed to the published version of the manuscript.

Funding: This research was carried out within the collaborative framework of an industrial doctoral thesis, which is financed by the Universidad de Cádiz and the shipbuilding company Navantia.

Acknowledgments: The development ofthis work was possible thanks to the industrial doctoral thesis program of the University of Cadiz with the shipbuilding company Navantia. In addition, it is also necessary to thank the University of Cadiz for the funds to publish this paper; the research group GAPSIS (Grupo de Automática, Procesamiento de Señales e Ingeniería de Sistemas); and the Systems Engineering Department of Navantia, whose previous work allowed the formulation of the framework in which this work esd carried out.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, Z.; Zhang, Y.; Yu, X.; Yuan, C. Unmanned surface vehicles: An overview of developments and challenges. *Annu. Rev. Control* **2016**, *41*, 71–93. [CrossRef]
2. Campbell, S.; Naeem, W.; Irwin, G.W. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annu. Rev. Control* **2012**, *36*, 267–283. [CrossRef]
3. González, D.; Joshué, P.; Milanés, V.; Nashashibi, F. A Review of Motion Planning Techniques for Automated Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1135–1145. [CrossRef]
4. Fossen, T.I. *Handbook of Marine Craft Hydrodynamics and Motion Control*; John Wiley & Sons: Trondheim, Norway, 2011; ISBN 9781119991496.
5. Fossen, T.I. *Marine Control Systems*; John Wiley & Sons: Trondheim, Norway, 2002; ISBN 8292356002.
6. Elfes, A. Using Occupancy Grids for Mobile Robot Perception and Navigation. *Computer* **1989**, *22*, 46–57. [CrossRef]
7. Lavalle, S. *Planning Algorithms*; Cambridge University Press: New York, NY, USA, 2006; ISBN 978-0-521-86205-9.
8. Saval-Calvo, M.; Medina-Valdés, L.; Castillo-Secilla, J.M.; Cuenca-Asensi, S.; Martínez-Álvarez, A.; Villagrá, J. A review of the bayesian occupancy filter. *Sensors* **2017**, *17*, 344. [CrossRef] [PubMed]
9. Perera, L.P.; Ferrari, V.; Santos, F.P.; Hinostroza, M.A.; Guedes Soares, C. Experimental Evaluations on Ship Autonomous Navigation and Collision Avoidance by Intelligent Guidance. *IEEE J. Ocean. Eng.* **2015**, *40*, 374–387. [CrossRef]
10. Perera, L.P.; Oliveira, P.; Guedes Soares, C. Maritime Traffic Monitoring Based on Vessel Detection, Tracking, State Estimation, and Trajectory Prediction. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 1188–1200. [CrossRef]
11. Kim, T.; Choi, J.; Lee, Y.; Choi, H.-T. VFH+ based Obstacle Avoidance using Monocular Vision of Unmanned Surface Vehicle. *J. Ocean Eng. Technol.* **2016**, *30*, 426–430. [CrossRef]
12. Minguez, J.; Lamiraux, F.; Laumond, J.-P. Motion Planning and Obstacle Avoidance. In *Handbook of Robotics*; Springer: Berlin/Heidelberg, Germany, 2008; Volume E, pp. 827–852, ISBN 9783540303015.
13. Kuwata, Y.; Wolf, M.T.; Zarzhitsky, D.; Huntsberger, T.L. Safe maritime autonomous navigation with COLREGS, using velocity obstacles. *IEEE J. Ocean. Eng.* **2014**, *39*, 110–119. [CrossRef]
14. Zhao, Y.; Li, W.; Shi, P. A real-time collision avoidance learning system for Unmanned Surface Vessels. *Neurocomputing* **2016**, *182*, 255–266. [CrossRef]
15. Perera, L.P.; Carvalho, J.P.; Guedes Soares, C. Intelligent ocean navigation and fuzzy-Bayesian decision/action formulation. *IEEE J. Ocean. Eng.* **2012**, *37*, 204–219. [CrossRef]
16. Johansen, T.A.; Perez, T.; Cristofaro, A. Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3407–3422. [CrossRef]
17. Sun, X.; Wang, G.; Fan, Y.; Mu, D.; Qiu, B. Collision avoidance using finite control set model predictive control for unmanned surface vehicle. *Appl. Sci.* **2018**, *8*, 18. [CrossRef]
18. Da Silva Junior, A.G.; dos Santos, D.H.; de Negreiros, A.P.F.; de Souza Silva Silva, J.M.V.B.; Gonçalves, L.M.G. High-Level Path Planning for an Autonomous. *Sensors* **2020**, *20*, 1550. [CrossRef]
19. Gonzalez Cantos, A. *Aplicación de la Teoría Cualitativa de Sistemas Dinámicos para el Guiado Autónomo de Vehículos*; Universidad de Cádiz: Cádiz, Spain, 2017.
20. Blanco, J.L.; González, J.; Fernández-Madrigal, J.A. Extending obstacle avoidance methods through multiple parameter-space transformations. *Auton. Robot.* **2008**, *24*, 29–48.
21. Savkin, A.V.; Hoy, M. Reactive and the shortest path navigation of a wheeled mobile robot in cluttered environments. *Robotica* **2013**, *31*, 323–330. [CrossRef]
22. Kundu, S.; Parhi, D.R. Reactive navigation of underwater mobile robot using ANFIS approach in a manifold manner. *Int. J. Autom. Comput.* **2017**, *14*, 307–320. [CrossRef]
23. Blaich, M.; Rosenfelder, M.; Schuster, M.; Bittel, O.; Reuter, J. Extended grid based collision avoidance considering COLREGs for vessels. In Proceedings of the 9th IFAC Conference on Manoeuvring and Control of Marine Craft, Arenzano Italy, 19–21 September 2012; Volume 9, pp. 416–421.
24. Singh, Y.; Sharma, S.; Sutton, R.; Hatton, D. Path Planning of an Autonomous Surface Vehicle based on Artificial Potential Fields in a Real Time Marine Environment. In Proceedings of the 16 International Conference on Computer and IT Applications in the Maritime Industries, Cardiff, Gales, UK, 15–17 May 2017; pp. 48–54.

25. Zhang, G.; Deng, Y.; Zhang, W. Robust neural path-following control for underactuated ships with the DVS obstacles avoidance guidance. *Ocean Eng.* **2017**, *143*, 198–208. [CrossRef]
26. Fiorini, P.; Shiller, Z. Motion Planning in Dynamic Environments Using Velocity Obstacles. *Int. J. Rob. Res.* **1998**, *17*, 760–772. [CrossRef]
27. Fox, D.; Burgard, W.; Thrun, S. The Dynamic Window Approach to Collision Avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [CrossRef]
28. Khatib, O. Real-Time Obstacle Avoidance For Manipulators Additionally, Mobile Robots. In Proceedings of the IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 25–28 March 1985; pp. 500–505.
29. Borenstein, J.; Koren, Y. Real-Time Obstacle Avoidance for Fast Mobile Robots. *IEEE Trans. Syst. Man Cybern.* **1989**, *19*, 1179–1187. [CrossRef]
30. Ulrich, I.; Borenstein, J. VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots. In Proceedings of the 1998 IEEE International Conference on Robotics & Automation, Leuven, Belgium, 16–20 May 1998; pp. 1572–1577.
31. Guardeño, R.; López, M.J.; Sánchez, J.; Consegliere, A. AutoTuning Environment for Static Obstacle Avoidance Methods Applied to USVs. *J. Mar. Sci. Eng.* **2020**, *8*, 300. [CrossRef]
32. Singh, Y.; Sharma, S.; Sutton, R.; Hatton, D. Towards use of Dijkstra Algorithm for Optimal Navigation of an Unmanned Surface Vehicle in a Real-Time Marine Environment with results from Artificial Potential Field. *TransNav Int. J. Mar. Navig. Saf. Sea Transp.* **2018**, *12*, 125–131. [CrossRef]
33. Agrawal, P.; Dolan, J.M. COLREGS-compliant target following for an Unmanned Surface Vehicle in dynamic environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 1065–1070.
34. Abdelaal, M.; Fränzle, M.; Hahn, A. Nonlinear Model Predictive Control for trajectory tracking and collision avoidance of underactuated vessels with disturbances. *Ocean Eng.* **2018**, *160*, 168–180. [CrossRef]
35. Xue, Y.; Lee, B.S.; Han, D. Automatic collision avoidance of ships. *Proc. Inst. Mech. Eng. Part M J. Eng. Marit. Environ.* **2009**, *223*, 33–46. [CrossRef]
36. Zhang, J.; Zhang, D.; Yan, X.; Haugen, S.; Guedes Soares, C. A distributed anti-collision decision support formulation in multi-ship encounter situations under COLREGs. *Ocean Eng.* **2015**, *105*, 336–348. [CrossRef]
37. Naeem, W.; Irwin, G.W.; Yang, A. COLREGs-based collision avoidance strategies for unmanned surface vehicles. *Mechatronics* **2012**, *22*, 669–678. [CrossRef]
38. Mousazadeh, H.; Jafarbiglu, H.; Abdolmaleki, H.; Omrani, E.; Monhaseri, F.; Abdollahzadeh, M.; Mohammadi-Aghdam, A.; Kiapei, A.; Salmani-Zakaria, Y.; Makhsoos, A. Developing a navigation, guidance and obstacle avoidance algorithm for an Unmanned Surface Vehicle (USV) by algorithms fusion. *Ocean Eng.* **2018**, *159*, 56–65. [CrossRef]
39. Wang, X.; Liu, Z.; Cai, Y. The ship maneuverability based collision avoidance dynamic support system in close-quarters situation. *Ocean Eng.* **2017**, *146*, 486–497. [CrossRef]
40. Loe, Ø.A.G. *Collision Avoidance for Unmanned Surface Vehicles*; Norwegian University of Science and Technology: Trondheim, Norway, 2008.
41. Tang, P.; Zhang, R.; Liu, D.; Huang, L.; Liu, G.; Deng, T. Local reactive obstacle avoidance approach for high-speed unmanned surface vehicle. *Ocean Eng.* **2015**, *106*, 128–140. [CrossRef]
42. Xie, S.; Wu, P.; Peng, Y.; Luo, J.; Qu, D.; Li, Q.; Gu, J. The obstacle avoidance planning of USV based on improved artificial potential field. In Proceedings of the 2014 IEEE International Conference on Information and Automation (ICIA), Hailar, China, 28–30 July 2014; pp. 746–751.
43. Tang, P.; Zhang, R.; Liu, D.; Zou, Q.; Shi, C. Research on near-field obstacle avoidance for unmanned surface vehicle based on heading window. In Proceedings of the 2012 24th Chinese Control and Decision Conference, Taiyuan, China, 23–25 May 2012; pp. 1262–1267.
44. Bertaska, I.R.; Shah, B.; Von Ellenrieder, K.; Švec, P.; Klinger, W.; Sinisterra, A.J.; Dhanak, M.; Gupta, S.K. Experimental evaluation of automatically-generated behaviors for USV operations. *Ocean Eng.* **2015**, *106*, 496–514. [CrossRef]
45. Mei, J.H.; Arshad, M.R. COLREGs based navigation of riverine Autonomous Surface Vehicle. In Proceedings of the 2016 IEEE 6th International Conference on Underwater System Technology: Theory and Applications, Penang, Malaysia, 13–14 December 2016; pp. 145–149.

46. Shah, B.C.; Bertaska, I.R.; Alvarez, J.; Sinisterra, A.J.; Von Ellenrieder, K.; Dhanak, M.; Gupta, S.K.; Member, S.; Svec, P.; Shah, B.C.; et al. Dynamics-Aware Target Following for an Autonomous Surface Vehicle Operating under COLREGs in Civilian Traffic. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 3871–3878.
47. Blaich, M.; Köhler, S.; Reuter, J.; Hahn, A. Probabilistic collision avoidance for vessels. *IFAC-PapersOnLine* **2015**, *28*, 69–74. [CrossRef]
48. Pêtrès, C.; Romero-Ramirez, M.A.; Plumet, F. A potential field approach for reactive navigation of autonomous sailboats. *Rob. Auton. Syst.* **2012**, *60*, 1520–1527. [CrossRef]
49. Krogh, B.H. A Generalized Potential Field Approach to Obstacle Avoidance Control. In *Robotics Research: The Next Five Years and Beyond*; RI/SME: Bethlehem, PA, USA, 1984.
50. Tilove, R.B. Local obstacle avoidance for mobile robots based on the method of artificial potentials. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Cincinnati, OH, USA, 13–18 May 1990; pp. 566–571.
51. Borenstein, J.; Koren, Y. The vector field histogram—Fast obstacle avoidance for mobile robots. *IEEE Trans. Robot. Autom* **1991**, *7*, 278–288. [CrossRef]
52. Snape, J.; Van Den Berg, J.; Guy, S.J.; Manocha, D. The hybrid reciprocal velocity obstacle. *IEEE Trans. Robot.* **2011**, *27*, 696–706. [CrossRef]
53. Guy, S.J.; Lin, M.; Manocha, D. Reciprocal n -body Collision Avoidance. In *Robotics Research*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1–16, ISBN 978-3-642-19456-6.
54. Chakravarthy, A.; Ghose, D. Obstacle avoidance in a dynamic environment: A collision cone approach. *IEEE Trans. Syst. Man Cybern.* **1998**, *28*, 562–574. [CrossRef]
55. Bibuli, M.; Singh, Y.; Sharma, S.; Sutton, R.; Hatton, D.; Khan, A. A Two Layered Optimal Approach towards Cooperative Motion Planning of Unmanned Surface Vehicles in a Constrained Maritime Environment. *IFAC-PapersOnLine* **2018**, *29*, 378–383. [CrossRef]
56. Blaich, M.; Rosenfelder, M.; Schuster, M.; Bittel, O.; Reuter, J. Fast grid based collision avoidance for vessels using A^* search algorithm. In Proceedings of the 17th International Conference on Methods & Models in Automation & Robotics, Miedzyzdrojie, Poland, 27–30 August 2012; pp. 385–390.
57. Astrom, K.J.; Wittenmark, B. *Adaptive Control*; Addison Whesley: Boston, MA, USA, 1989; ISBN 0201097206.
58. Kallstrom, C.G. *Identification and Adaptive Control Applied to Ship Steering*; Department of Automatic Control, Lund University: Lund, Sweden, 1979.
59. Lewis, E.V. *Principles of Naval Architecture, Vol III—Motions in Waves and Controllability*; SNAME (The Society of Naval Architects and Marine Engineers): Jersey City, NJ, USA, 1989; ISBN 0939773023.
60. López, M.J.; Consegliere, A.; Terrón, J. *Modelado y Simulación de Sistemas. Aplicación al Buque*; Universidad de Cádiz: Cádiz, España, 1997; doi:10.13140/RG.2.1.3400.5843. [CrossRef]
61. Richards, A.; How, J.P. Robust variable horizon model predictive control for vehicle maneuvering. *Int. J. Robust Nonlinear Control* **2006**, *16*, 333–351. [CrossRef]
62. Guardeño, R.; López, M.J.; Sánchez, V.M. MIMO PID Controller Tuning Method for Quadrotor Based on LQR/LQG Theory. *Robotics* **2019**, *8*, 36. [CrossRef]
63. MathWorks. Navigation Toolbox. Available online: https://es.mathworks.com/products/navigation.html (accessed on 4 April 2020).
64. Do, K.D.; Jiang, Z.P.; Pan, J. Robust adaptive path following of underactuated ships. *Automatica* **2004**, *40*, 929–944. [CrossRef]
65. Han, J.; Xiong, J.; He, Y.; Gu, F.; Li, D. Nonlinear Modeling for a Water-Jet Propulsion USV: An Experimental Study. *IEEE Trans. Ind. Electron.* **2017**, *64*, 3348–3358. [CrossRef]
66. Rubio, F.R.; López, M.J. *Control Adaptativo y Robusto*, 1st ed.; Universidad de Sevilla: Sevilla, Spain, 1996; ISBN 84-472-0319-0.
67. Velodyne Lidar. *UltraPuck Data Sheet*; Velodyne: Hellyer Ave, SJ, USA, 2019.
68. Government of Spain. Reglamento Internacional para Prevenir los Abordajes, 1972. *Boletín del Estado* **1977**, *17*, 15421–15432.
69. Sørensen, A.J. *Marine Control Systems. Propulsion and Motion Control of Ships and Ocean Structures*; Norwegian University of Science and Technology: Trondheim, Norway, 2012.

70. Yang, Z.; Seested, G.T. Time-delay system identification using genetic algorithm—Part one: Precise FOPDT model estimation. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*; IFAC: Chengdu, China, 2013; Volume 3, pp. 561–567.
71. Kristinsson, K.; Dumont, G.A. System Identification and Control Using Genetic Algorithms. *IEEE Trans. Syst. MAN Cybern.* **1992**, *22*, 1033–1046. [CrossRef]
72. Nyarko, E.K.; Scitovski, R. Solving the parameter identification problem of mathematical models using genetic algorithms. *Appl. Math. Comput.* **2004**, *153*, 651–658. [CrossRef]
73. Simon, D. *Evolutionary Optimization Algortithms*; John Wiley & Sons: Hoboken, NJ, USA, 2013; ISBN 978-0-470-93741-9.
74. MathWorks. Genetic Algorithm. Available online: https://es.mathworks.com/help/gads/genetic-algorithm.html (accessed on 5 July 2019).
75. Berndt, B.C. *Ramanujan's Notebooks: Part III*, 1st ed.; Springer: New York, NY, USA, 1991; ISBN 978-1-4612-0965-2.
76. MathWorks. Vector Field Histogram. Available online: https://es.mathworks.com/help/nav/ug/vector-field-histograms.html (accessed on 23 May 2019).
77. MathWorks. Measure the Performance of Your Code. Available online: https://es.mathworks.com/help/matlab/matlab_prog/measure-performance-of-your-program.html (accessed on 4 June 2020).
78. MSI. GT60 2PC Dominator. Available online: https://www.msi.com/Laptop/GT60-2PC-Dominator/Specification (accessed on 10 May 2020).