

DATA-DRIVEN EXPLORATION OF LENTIC WATER BODIES WITH ASVS GUIDED BY GRADIENT-FREE OPTIMIZATION/CONTOUR DETECTION ALGORITHMS

Eva Besada-Portas
José María Girón-Sierra
Juan Jiménez
José Antonio López-Orozco

Department of Computer Architecture and Automation.
University Complutense of Madrid
Ciudad Universitaria s/n
28040 Madrid, Spain

ABSTRACT

This paper presents a local-path planner for water quality monitoring involving an Autonomous Surface Vehicle (ASV). The planner determines new measuring waypoints based on the information collected so far, and on two gradient-free optimization and contour-detection algorithms. In particular, the optimization algorithm generates the locations where the variable/substance under study must be measured and use them as the waypoints of the external loop of the Guidance, Navigation and Control system of our ASV. Besides, the contour algorithm obtains useful waypoints to determine the water body locations where the variable/substance under study reaches a given value. The paper also analyzes how the approach works via progressive simulations over an ASV carefully modelled with a set of non-linear differential equations. Preliminary results suggest that the approach can be useful in real-world single-ASV water-quality monitoring missions where there is not previous knowledge of the state and location of the variable/substance under study.

1 INTRODUCTION

Fresh water is a fragile resource that has to be monitored to guarantee its adequate conditions for human consumption and recreational uses. Population growth, water misuse, pollution, extreme floods and droughts threaten the availability of good-quality inland water bodies. The importance of the problem, which also affects the species that live around/within the water body, has made water availability and its management the third goal of the United Nations 2030 Agenda for Sustainable Development (United Nations 2021). International regulations (e.g. of the United States Environmental Protection Agency 2021 and of the European Commission 2021) also face the problem and indicate, among other things, the parameters and substances that should be studied, their admissible or dangerous values, and their monitoring frequency.

Traditional water monitoring approaches range from manually collecting water samples (from manned boats) in a few points of the water body to automatically measuring water parameters/substances with probes placed at the few geographically distributed stations of early warning networks (Storey et al. 2011). To fulfill the newest regulations requirements, these approaches are being complemented with the use of Autonomous Surface Vehicles (ASVs) especially designed for that purpose and capable of moving the probes that measure the parameters/substances under study to the points of interest within the water body (Hitz et al. 2012; Siyang and Kerdcharoen 2016; Shuo et al. 2017). To make these self-driving boats autonomous monitoring platforms, it is necessary to equip them with a complete automation system that

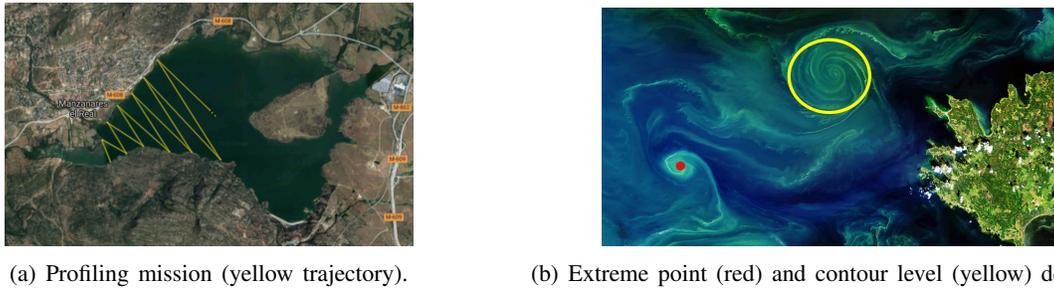


Figure 1: ASV water quality missions.

incorporates advanced location, perception, planning, guidance, navigation and control techniques that decide how to deploy and move the ASVs (Liua et al. 2016).

ASVs distribution and guidance approaches highly depend on the available information of the water body and on the phases contemplated by the automation system that makes the monitoring mission successful. Additionally, they have to consider the purpose of the mission, such as determining, with an adequate granularity, the state of the whole water body or discovering the regions where a selected parameter/substance (e.g. temperature, pH, dissolved oxygen and nitrogen, algae or pollutant concentration) reaches an extreme or fixed value/concentration. The first case, schematized in Fig. 1(a), can consume a lot of time of a single ASV but it is useful to provide a complete profile of the water body. The second one, illustrated in Fig. 1(b), permits to detect regions where the variable/substance is maximal or over the regulated thresholds.

One option to tackle water monitoring problems with ASVs is to use a global planning phase to determine their complete trajectories before the monitoring mission starts. Moreover, the shapes of these trajectories can be optimized according to different criteria. For instance, and to name a few, if no information related to the variables under study is available, the monitoring process can be set up as a coverage problem (where the ASVs routes are distributed uniformly to reach as many parts of the water body as possible) and solved using different types of patterns (Valada et al. 2012; N. Karapetyan and Rekleitis 2019) or as a Travelling Salesman Problem among shore points (Arzamendia et al. 2016; Arzamendia et al. 2019). Otherwise, when the information to gather in the Regions Of Interest (ROIs) is relevant, the mission can be set up as an information gain problem, which has already been tackled with bio-inspired optimization algorithms (Xiong et al. 2019; Xiong et al. 2020). Alternatively, it is also possible to exploit the information provided by simulators of the water body dynamics and of the pollutants dispersion, in order to obtain ASV trajectories that maximize the chances to detect the pollution and that minimize the mission time and trajectory length (Carazo-Barbero et al. 2021). An additional possibility within this group arises when an operator decides which are the points of interest and the planner has to find the best path to visit them (Xia et al. 2019).

Independently of the underlying purpose, the previous global pre-planning techniques do not take into account the observations that are made by the sensors on board the ASV during the monitoring mission. The A*, Potential Field, Rapid-Exploring Random Tree and Fast Marching approaches analyzed in Peralta et al. (2020) re-plan the trajectories between two points of the global trajectory (determined by Arzamendia et al. 2019) in order to consider the current situation, avoiding obstacles and trying to make the ASV path shorter. Nevertheless, the ASV behavior is not driven by the data provided by the sensors that measure the state of the variables/substances under study. Other local planners do not react to this type of information, determining where the next measurements must take place as the intermediate solutions of an optimizer of the unknown function that models the behavior of the variable/substance under study (Bu et al. 2013) or as the most informative points of a Gaussian Process that models the state of the variable under analysis and its uncertainty (Blix 2019). However, the last two works are not intended to guide an ASV, which is both a drawback and an opportunity for developing new approaches.

In fact, the philosophy of the last two data-driven measurement-taken approaches can be adapted to guide the ASV towards the most promising ROIs of the water body, according to the information of the

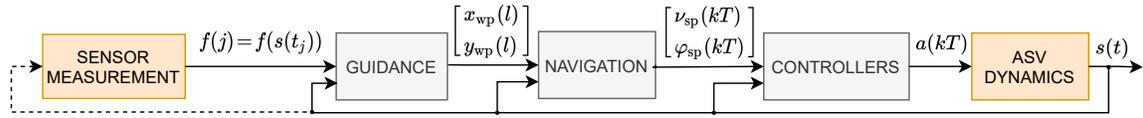


Figure 2: Relationships among the different modules of the whole system.

variables under analysis provided by its sensors. To achieve it, we automate the ASV behavior incorporating approaches capable of exploring unknown functions (in our case of the variable under study) within a Guidance, Navigation and Control (GNC) system that moves the ASV towards the waypoints suggested by the exploring methods. In particular, we design two GNCs for: 1) determining the location of the extreme points of the variable under analysis or 2) obtaining the shape of a level curve. Moreover, their guidance subsystems (or local planners) are directly and respectively inspired by the Nelder-Mead Simplex Optimization Algorithm (NMSOA, Nelder and Mead 1965) and the reliable PATH following algorithm (PAT, Mezher and Philippe 2000), while their navigation and control loops share a streamlined regulation system that makes the ASV follow straight lines between two consecutive waypoints. NMSOA and PAT are especially interesting for our problem, since they only need a single measurement of the function under analysis (i.e of the variable of interest in our case) at each waypoint.

The purpose of this paper is to present both systems and to analyze their performance under different simulations, when they are used to guide a single-propeller single-rudder ASV. Simulations in this case are extremely important as they provide support for the different phases of this research. On one hand, the incremental simulation of the different parts of the system (ASV, ASV+control+ navigation, and ASV+GNC) is required because the non-linearities and high inertia of the ASV make it a difficult system to control and guide. On the other one, a systematic simulation of the whole system allows us to tune the GNC parameters with the purpose of improving the shape of the ASV trajectory and the overall performance of the monitoring mission. Last, but not least, simulations let us anticipate some problems that would have appeared in the experiments, and therefore to optimize our time of research.

2 OVERVIEW OF THE SYSTEM AND COMMON MODULES

Our water quality monitoring system consists of a single ASV equipped with a sensor capable of taking periodic measures of a given variable of the water body. The ASV is automatically guided using a GNC that takes into account the measurements provided by the sensor and the objective of the ASV: either to determine the location of the extreme values of the variable of interest or the contour defined by a fixed level of that variable. Its extension to multiple ASVs or sensors is briefly discussed in Section 5.

The modules involved in the system are represented in Fig. 2, following the common structure of a control system with multiple closed-loops, and using gray to indicate which modules are part of the GNC, and orange to mark which physical elements are simulated in this paper. Besides, $\mathbf{s}(t)$ stands for the state of the ASV, $\mathbf{a}(kT)$ for the control signals for the ASVs actuators (propeller and rudder in the ASV used in this paper), $[\mathbf{v}_{sp}(kT), \phi_{sp}(kT)]^T$ for the ASV speed and angle setpoints, $[x_{wp}(l), y_{wp}(l)]^T$ for the coordinates of the current waypoint, and $f(j) = f(\mathbf{s}(t_j))$ for the measurement taken by the water quality sensor. Finally, variable t implies that \mathbf{s} is a continuous-time signal; kT that \mathbf{a} , \mathbf{v}_{sp} and ϕ_{sp} are discrete-time periodic signals; and l and j that x_{wp} , y_{wp} and f only change when the events of producing a new waypoint or of taking a new measurement happen.

From a simulation point of view, this implies that the ASV state \mathbf{s} is obtained integrating a continuous-time system; that the GNC subsystems are run periodically (at every T s); and that the guidance changes its output and asks for measurements only when required. Moreover, to improve the interactions between the continuous-time ASV dynamics and the discrete-time GNC subsystems, we have performed many different simulations that suggest that, for reducing the oscillations of the ASV trajectories, it is convenient to iterate the GNC subsystems at $T = 0.1$ s and to obtain the ASV state \mathbf{s} with a 4th order Runge-Kutta that iterates at $dt = T$.

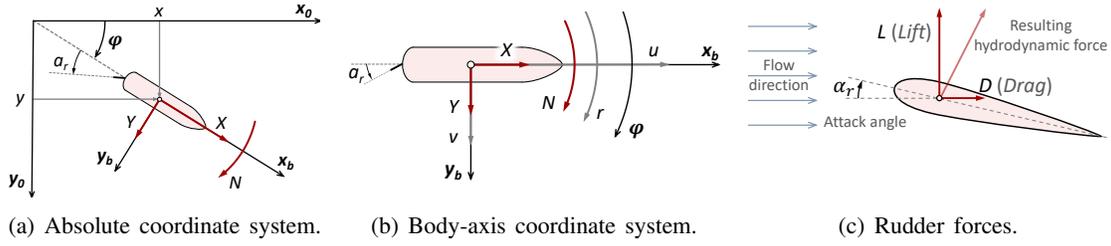


Figure 3: ASV model variables and coordinates systems.

Table 1: ASV complete dynamic model.

Expressions that relate model inputs, states and intermediate variables		Differential expressions	
$v_i = \frac{a_p}{100}$ (A1)	$T_h = 2 \rho A v_i^2$ (A5)	$\left. \begin{aligned} \dot{x} &= u \cos \varphi - v \sin \varphi \\ \dot{y} &= u \sin \varphi + v \cos \varphi \\ \dot{\varphi} &= r \\ \dot{u} &= \frac{X}{m} + r \cdot v \\ \dot{v} &= \frac{Y}{m} - r \cdot u \\ \dot{r} &= \frac{N}{\frac{m}{12} \cdot (b^2 + w^2)} \end{aligned} \right\} \text{(A9)}$	
$u_r = u + v_i$	$\alpha = \alpha_r - a_r$ (A6)		
$v_r = v - r \cdot l_r$	$X_r = -D \cos \alpha + L \sin \alpha$		
$V_r^2 = u_r^2 + v_r^2$	$Y_r = -D \sin \alpha - L \cos \alpha$		
$\alpha_r = \text{atan2}(v_r, u_r) + a_r$ (A3)	$X = -c_{\text{front}} \cdot u + T_h + X_r$ (A7)		
$L = \frac{1}{2} \rho V_r^2 S_r C_{la} \alpha_r$	$Y = -c_{\text{sideways}} \cdot v + Y_r$		
$D = \frac{1}{2} \rho V_r^2 S_r (C_{d0} + C_{da} \alpha_r^2)$ (A4)	$N = -c_{\text{rotate}} \cdot r - l_r \cdot Y_r$ (A8)		

Table 2: Parameters of the ASV dynamic model.

$m = 20 \text{ kg}$	$\rho = 997 \text{ kg/m}^3$	$A = 0.0079 \text{ m}^2$	$l_r = 0.3 \text{ m}$	$b = 1.2 \text{ m}$	$w = 0.4 \text{ m}$	$dt = 0.1 \text{ s}$
$C_{la} = 0.07$	$C_{d0} = 7.37 \cdot 10^{-5}$	$C_{da} = 0.0037$	$S_r = 0.03 \text{ m}^2$	$c_{\text{front}} = 1.6$	$c_{\text{sideways}} = 12.8$	$c_{\text{rotate}} = 0.06$

2.1 ASV Dynamics

The ASV dynamics are captured by the model of 6 degree of freedom presented in Table 1, whose expression (A9) states the differential equations over the variables within $\mathbf{s}(t) = [x(t), y(t), \varphi(t), u(t), v(t), r(t)]^T$, where $(x(t), y(t), \varphi(t))$ stand for the ASV location and orientation in the absolute coordinate system of Fig. 3a, and $(u(t), v(t), r(t))$ stand for the ASV longitudinal, lateral and angular speeds in the body coordinated system of Fig. 3b. The model inputs (highlighted in magenta in Table 1) are the control signals of the ASV actuators $(\mathbf{a}(kT) = [a_p(kT), a_r(kT)]^T)$, i.e. the required ASV propulsion $(a_p(kT))$, provided as a percentage value within $[0, 100]$ and rudder angle $(a_r(kT))$ within $[-30, 30]$ deg). Additionally, Equation (A1) converts the required $a_p(t)$ percentage into the propeller speed v_i ; (A2) obtains the rudder speeds in the body-axis (u_r, v_r) and its module (V_r) taking into account the distance (l_r) between the rudder and the ASV gravity center; (A3) determines the rudder attack angle (α_r) ; (A4) obtains the rudder Drag (D) and Lift (L) forces considering the rudder surface (S_r) and hydrodynamic coefficients $(C_{la}, C_{d0}$ and $C_{da})$; (A5) calculates the ASV Thrust (T_h) taking into account the water density (ρ) and propeller section (A) ; (A6) projects D and L to the body-axis (X_r, Y_r) ; (A7) determines the total forces (X, Y) over the ASV in the body-axis taking into account the friction coefficients $(c_{\text{front}}$ and $c_{\text{sideways}})$; and (A8) calculates the ASV torque (N) considering the friction coefficient (c_{rotate}) . Finally, in (A9) m stands for the ASV mass, b for its length and w for its width (w) . Further details on these expressions can be found in Fossen (2002).

To simulate the ASV behavior we use the constants provided in Table 2. The parameters of the first row correspond to properties of our own-built ASV, while the ones in the second row have been adjusted to make the simulated ASV behave as ours. Besides, we integrate (A9) with a 4th order Runge-Kutta that assumes that inputs $(a_p(kT), a_r(kT))$ are constant during the integration time dt . The results of a few simulations with different control inputs are provided in Fig. 4. In particular, the two left graphics represent the ASV trajectories $(y(t) \text{ vs. } x(t))$ and signals (ASV angle φ and speed $V = \|[u, v]^T\|$) obtained

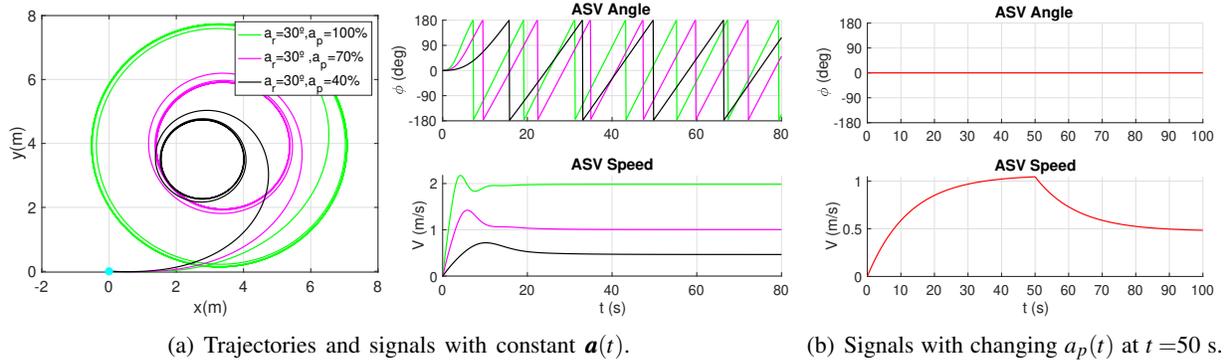

 Figure 4: ASV simulations under different $\mathbf{a}(t)$.

Table 3: Controller common equations.

PI controller usual operations	Anti-windup additional operations
$i_*(kT) = i_*(k-1)T + e_*(kT) \cdot T \quad (C2)$	$\left. \begin{array}{l} \text{if } a_*^{\text{aux}}(kT) \neq a_*(kT) \text{ then} \\ i_*(kT) = i_*(kT) - K_{*,a}(a_*^{\text{aux}}(kT) - a_*(kT)) \end{array} \right\} (C4)$
$a_*^{\text{aux}}(kT) = K_{*,p} \cdot e_*(kT) + K_{*,i} \cdot i_*(kT)$	
$a_*(kT) = \max(\min(a_*^{\text{aux}}(kT), \text{UB}_*), \text{LB}_*)$	

with the same $a_r = 30$ deg at $a_p = 100\%$, $a_p = 70\%$ and $a_p = 40\%$. In them, we can observe that the circular trajectories (obtained after the transient time is over) with smaller radius occur at lower a_p and that the ASV speed V is not proportional to the required propulsion. Besides, Fig. 4(b) represents the signals obtained when we fix the rudder angle to $a_r = 0$ and make the propulsion a_p change from 33% to 22% at $t=50$ s. The purpose of this simulation is to show the high inertia of the ASV, which reaches steady state speeds of 1 and 0.5 m/s after a settling time of 40 s. Furthermore, if we compare the propulsion (labels or values) and speeds signals in the graphics of Fig. 4(a) and 4(b), we can conclude that different propulsion values are required to obtain similar speeds at different rudder angles. All these facts make controlling the ASV speed and orientation, in order to make it follow the linear-trajectories defined by pairs of waypoints, a difficult task that we handle with the GNC presented in the following section.

2.2 Controller and Navigation Modules of the GNC

The controller is in charge of generating the actuators signals $\mathbf{a}(kT) = [a_p(kT), a_r(kT)]^T$ associated to the requested ASV speed and orientation setpoints $[v_{\text{sp}}(kT), \varphi_{\text{sp}}(kT)]^T$ and to the ASV speed and orientation $[V(kT), \varphi(kT)]^T$. To do it, it implements two independent discrete-time Proportional Integral (PI) controllers with anti-windup (Visioli 2006) for the ASV speed and orientation. Their first step (C1) computes the controller error (i.e. it calculates $e_p(kT) = v_{\text{sp}}(kT) - V(kT)$ and $e_r(kT)$ as the shortest difference in degrees between $\varphi_{\text{sp}}(kT)$ and $\varphi(kT)$). The remaining steps are summarized in Table 3, using subindex $*$ to show that they are replicated within each controller (i.e. with $* = r$ and $* = p$). Its left column displays the remaining usual equations of the PI: (C2) integrates the error; and (C3) combines the proportional and integral terms, and bounds the obtained signal (using $[\text{LB}_p, \text{UB}_p] = [0, 100]$ % and $[\text{LB}_r, \text{UB}_r] = [-30, 30]$ deg). Its right column implements the anti-windup operations. Each controller is executed every $T = 0.1$ s to generate, with an appropriated time resolution, the control signals $\mathbf{a}(kT) = [a_p(kT), a_r(kT)]^T$ that are hold constant during the ASV model integration step.

The navigation module calculates the ASV requested setpoints $[v_{\text{sp}}(kT), \varphi_{\text{sp}}(kT)]^T$ given the current ASV state $\mathbf{s}(t)$, and the current and previous waypoints (i.e. $[x_{\text{wp}}(l), y_{\text{wp}}(l)]^T$ and $[x_{\text{wp}}(l-1), y_{\text{wp}}(l-1)]^T$), which are updated by the guidance system whenever the distance between the ASV and current waypoint $[x_{\text{wp}}(l), y_{\text{wp}}(l)]^T$ is smaller than 5 m. According to the internal operations of Table 4, this module calculates

Table 4: Navigation equations.

Internal operations for obtaining d_{Line} and d_{wp}	Speed setpoint regulator
$\left. \begin{aligned} \Delta x_{wp} &= x_{wp}(l) - x_{wp}(l-1) \\ \Delta y_{wp} &= y_{wp}(l) - y_{wp}(l-1) \\ \boldsymbol{\tau} &= [\Delta x_{wp}, \Delta y_{wp}]^T / \sqrt{\Delta x_{wp}^2 + \Delta y_{wp}^2} \end{aligned} \right\} \quad (N1)$	$\left. \begin{aligned} &\text{if } d_{wp}(kT) < 15 \text{ then } v_{sp}(kT) = 1 \\ &\text{else } v_{sp}(kT) = 0.5 \\ &e(kT) = \text{AngleDifferenceDeg}(\varphi_{sp}(kT), \varphi(kT)) \\ &\text{if } d_{Line}(kT) > 1 \text{ then} \\ &\quad v_{sp}(kT) = v_{sp}(kT) - \frac{(v_{sp}(kT)-0.5) \cdot d_{Line}(kT) }{2} \\ &\text{elseif } e(kT) > 20 \text{ then} \\ &\quad v_{sp}(kT) = v_{sp}(kT) - \frac{(v_{sp}(kT)-0.5) \cdot e(kT) }{40} \\ &v_{sp}(kT) = \max(0.5, v_{sp}(kT)) \end{aligned} \right\} \quad (N6)$
$\left. \begin{aligned} A &= \tau[2] \\ B &= -\tau[1] \\ C &= -A \cdot x_{wp}(l) - B \cdot y_{wp}(l) \\ d_{Line}(kT) &= A \cdot x(kT) + B \cdot y(kT) + C \end{aligned} \right\} \quad (N2)$	
$d_{wp}(kT) = \ [x_{wp}(l) - x(kT), y_{wp}(l) - y(kT)]^T\ \quad (N3)$	
Orientation setpoint regulator	
$\beta(kT) = \max(\min(K_{n,p} \cdot d_{Line}(kT), 90), -90) \quad (N4)$	
$\varphi_{sp}(kT) = \text{atan2}(\tau[2], \tau[1]) + \beta(kT) \quad (N5)$	

Table 5: Controller and navigation tunable parameters.

	Rudder Controller			Propulsion Controller			Navigation
	$K_{r,p}$	$K_{r,i}$	$K_{r,a}$	$K_{p,p}$	$K_{p,i}$	$K_{p,a}$	$K_{n,p}$
Testing ranges	[2,10]	[0,0.9]	[0.25,1.25]	[120,160]	[10,50]	[0.05,0.4]	[2.5,17.5]
Selected values	5	0.02	0.28	135	40	0.025	4.3

with (N1) the normalized vector $\boldsymbol{\tau}$ that joins the two waypoints, with (N2) the distance $d_{Line}(kT)$ between the ASV location and the line defined by the waypoints, and with (N3) the distance $d_{wp}(kT)$ between the ASV location and current waypoint. Next, with (N4) and (N5), the orientation setpoint regulator obtains $\varphi_{sp}(kT)$ by correcting the angle of the line with a term ($\beta(kT)$) that reduces the distance to the line. Finally, with (N6) the speed setpoint regulator obtains $v_{sp}(kT) \in [0.5, 1]$ m/s, taking into account the distances $d_{wp}(kT)$ and $d_{Line}(kT)$, and the shortest angular discrepancy between $\varphi_{sp}(kT)$ and $\varphi(kT)$, with the purpose of reducing the ASV speed when the ASV is turning or reaching the current waypoint.

The values of the adjustable parameters of the controller and navigation module, displayed at Table 5 (and the values of the fixed parameters of the speed setpoint regulator) have been tuned performing many simulations of the whole system with a generic guidance module that returns the waypoints of a user-defined trajectory. In particular, we make the GNC use the three fixed waypoints represented as green stars in Fig. 5(a), run thousands of simulations for many combination of values of the parameters within the ranges defined in the second row of Table 5, and successively shorten the ranges of each parameter while determining the combination of values that generate low values of overshooting, settling time and steady state error for $d_{Line}(kT)$. In other words, we tune the parameters to help the ASV follow the straight line that joins $[x_{wp}(l-1), y_{wp}(l-1)]^T$ and $[x_{wp}(l), y_{wp}(l)]^T$. The results of the simulation with the selected values (placed at the bottom row of Table 5) are displayed in Fig. 5(a), using red lines for the ASV trajectory $(x(kT), y(kT))$, actuator signals $[a_p(kT), a_r(kT)]^T$, ASV angle ($\varphi(kT)$) and ASV speed ($V(kT)$), and black lines for the segments that join the waypoints and for the navigation signals $[v_{sp}(kT), \varphi_{sp}(kT)]^T$. The ASV speed graphic shows that the ASV arrives at location [15,0] m after 15 s and is required to maintain a speed smaller than 1 m/s during 28 s, until the ASV is closer than 1 m to the vertical line and its orientation is between [70,110] deg. Besides, the ASV angle graphic shows that the ASV reaches point [25,0] m at 32 s, and that its angle is almost 90 deg after 40 s. Finally, the trajectory shows that the right distance to the vertical line is smaller than 0.5 m, while the rudder angle curve suggests that some parameters could be tuned better to avoid its continuous oscillations (probably, at the expense of increasing the time required to make the ASV trajectory reach the vertical line). Nevertheless, we have decided to use the selected parameters, as the controller and navigation modules are good enough to analyze the performance of the guidance systems presented in Sections 3 and 4, which contain the main contributions of this paper.

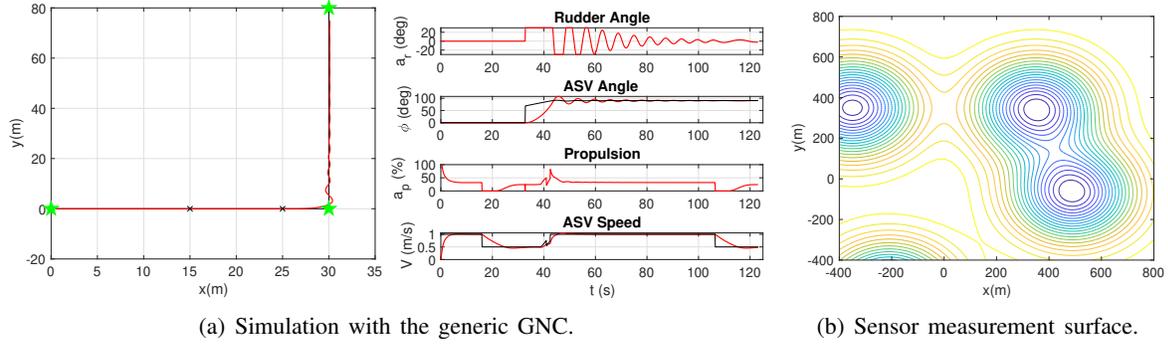


Figure 5: ASV and sensor simulations.

2.3 Sensor Measurements

The ASV is equipped with a static sensor capable of taking instantaneous measurements of the variable/substance under study. To model its behavior, we build the static multi-modal function represented in Fig. 5(b) by its contour curve and defined as $f(x,y) = \sum_{k=1}^4 [e^{-[(x-x_k)^2+(y-y_k)^2]/49000}]/1000$, and set $[x_1, y_1]^T = [350, 350]^T$ m, $[x_2, y_2]^T = [-350, 350]^T$ m, $[x_3, y_3]^T = [490, -70]^T$ m and $[x_4, y_4]^T = [-210, -560]^T$ m in order to distribute $f(x,y)$ extreme points into a water mass of 1200×1200 m². It is worth noting that the sign and magnitude of the function is not relevant for our approach, as the selected exploring algorithms only compare pair of values of the function. So, our proposal can work with any sensor that instantaneously provides continuous values of the variable under study when required.

3 GUIDING THE ASV TOWARDS THE EXTREME VALUES OF THE SENSED VARIABLE

The first guidance module generates waypoints that make the ASV explore the unknown variable under study and determine the locations of its extreme (maximum or minimum) values. It is inspired by the Nelder and Mead Simplex Optimization Algorithm (NMSOA, Nelder and Mead 1965, Mathews and Fink 2004), summarized and adapted in the following two sections. Finally, the last subsection within this section also shows the behavior of the whole system (GNC + ASV + Sensor) over the water body scenario of Fig. 5(b).

3.1 Nelder and Mead Simplex Optimization Algorithm (NMSOA)

NMSOA is an iterative deterministic optimization algorithm that obtains the minimal value of a given function measuring its values in different waypoints which are obtained triangulating the search space through 5 different operations: reflection, expansion, external concentration, internal concentration and shrinking. The order and number of operations is determined by the evaluation of the new points and its relation with the previous. We have selected this approach as it is a good option when the number of evaluations must be reduced and when calculating first or second order derivatives is not straightforward. This happens in our case, as $f(x,y)$ is unknown and its evaluations are obtained from the instantaneous measurements taken over the trajectory of a single ASV. Finally, note that to obtain the maximal value of the unknown function, we only need to change the sign of its values (i.e. run the algorithm with $-f(x,y)$).

For optimizing the two dimensional function $f(x,y)$ required by this paper, the algorithm starts selecting three not-aligned points ($\mathbf{p}_1 = [x_1, y_1]^T$, $\mathbf{p}_2 = [x_2, y_2]^T$ and $\mathbf{p}_3 = [x_3, y_3]^T$) and evaluating them ($f_1 = f(\mathbf{p}_1) = f(x_1, y_1)$, $f_2 = f(\mathbf{p}_2) = f(x_2, y_2)$ and $f_3 = f(\mathbf{p}_3) = f(x_3, y_3)$). Next, it sorts the points and its values, to ensure that $f_1 \leq f_2 \leq f_3$, and starts the iteration process summarized in Fig. 6(a), using a gray block for obtaining the centroid \mathbf{c} , solid-yellow for the reflection operation (to obtain \mathbf{p}_r and f_r), solid-red for the expansion (\mathbf{p}_e and f_e), solid-green for the external concentration (\mathbf{p}_{ce} and f_{ce}) and solid-blue for the internal concentration (\mathbf{p}_{ci} and f_{ci}). The operations order is imposed by the decision (romboid) blocks

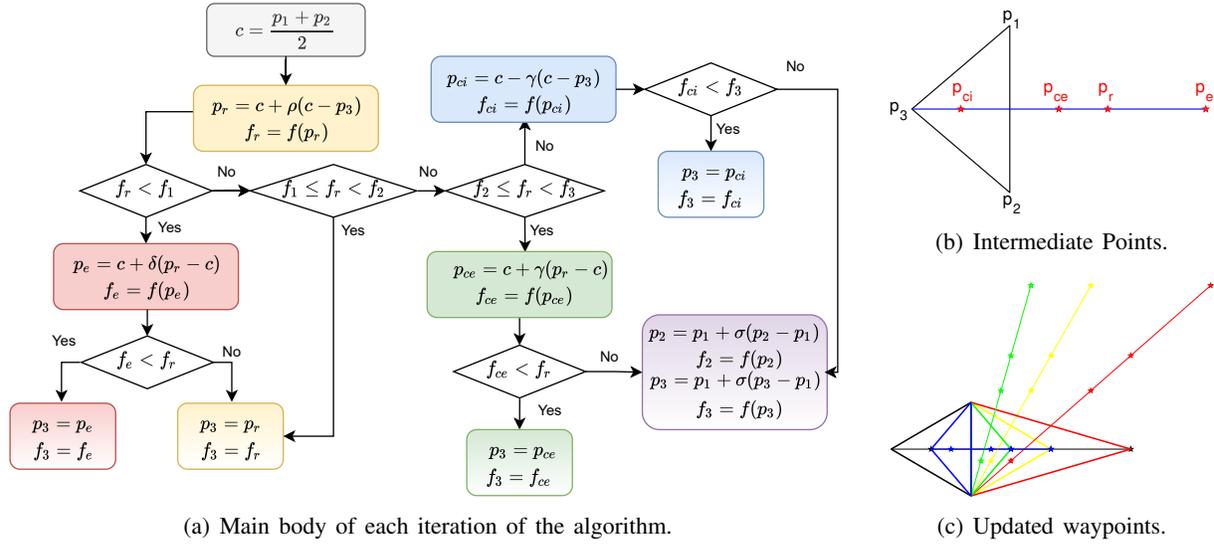


Figure 6: Nelder Mead simplex optimization algorithm (NMSOA).

and the obtained evaluations. The values of the original points $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ of the algorithm are updated according to the operations indicated in the gradient-colored blocks (using the same color schema than for the operation where the updating values were calculated, and gradient-violet for the shrinking operation). Finally, $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ are sorted again according to their values (f_1, f_2, f_3) to re-start the process ensuring that $f_1 \leq f_2 \leq f_3$. The algorithm iterates until the sides of the triangle $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ are smaller than a given value, and returns the lower value of the original unknown $f(x, y)$ and its location in f_1 and \mathbf{p}_1 .

The algorithm is quite simple, as the operations in 6(a) show, and has some interesting properties for guiding an ASV. First, it is possible to calculate, with a negligible computation cost, the intermediate points of the algorithm $(\mathbf{p}_r, \mathbf{p}_e, \mathbf{p}_{ce}$ and $\mathbf{p}_{ci})$ before evaluating the function in any of them. Second, when using the typical values of the algorithm parameters (i.e. $\rho = 1$, $\delta = 2$, $\gamma = 0.5$ and $\sigma = 0.5$), the intermediate points are aligned in the segment defined by \mathbf{p}_3 and \mathbf{p}_e , as the schema represented in Fig. 6(b) for a given set of $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ shows. Finally, the algorithm changes the exploring direction as required, as shown in Fig. 6(c), where the new possible set of points under study after accepting the results of each of the possible operations of the current step are displayed using the color-schema of 6(a).

3.2 Using and Modifying NMSOA for ASV Guidance

NMSOA can be straightforwardly used to produce the waypoints $[x_{wp}(l), y_{wp}(l)]^T$ generated by the guidance system. To do it, it only has to calculate the values of the intermediate points $(\mathbf{p}_r, \mathbf{p}_e, \mathbf{p}_{ce}$ and $\mathbf{p}_{ci})$ and output them as waypoints in the order indicated by its decision blocks. Besides, as the measurement of the variable under study has to be taken once the ASV reaches the waypoint, the evaluation $f_* = f(\mathbf{p}_*)$ has to be postponed to the time instant where the ASV is close enough to the corresponding waypoint. We prefer to anticipate the evaluation of the waypoint slightly, to facilitate the turns that have to be performed in the ASV trajectory after a new intermediate point of NMSOA is output as a waypoint $[x_{wp}(l), y_{wp}(l)]^T$ of the guidance system.

This way of proceeding lets the guidance system identify a local extreme of the variable/substance under study. In particular, to obtain a minimum, we use the value provided by the sensor, and to obtain a maximum, its opposite value. Nevertheless, we propose a modification to consider that 1) the ASV must move from one waypoint to another before taking the measurements required for NMSOA and that 2) in each iteration the ASV can be closer to the new \mathbf{p}_{ci} , \mathbf{p}_{ce} or \mathbf{p}_e than to \mathbf{p}_r (the first point to visit according to NMSOA). The new variant determines first which intermediate point $(\mathbf{p}_r, \mathbf{p}_e, \mathbf{p}_{ce}$ or $\mathbf{p}_{ci})$ is the closest

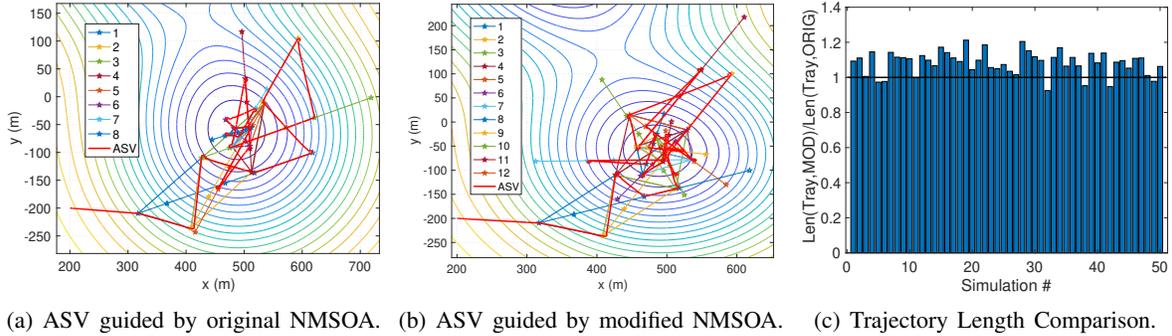


Figure 7: Simulations of the ASV guided to detect the minimum value of a given variable.

($\mathbf{p}_{closest}$) to the current ASV location ($x(t), y(t)$). Second, it makes the ASV go to it, and visit all the intermediate points in its way from $\mathbf{p}_{closest}$ to \mathbf{p}_r without taking any decision during this process. For instance, if $\mathbf{p}_{closest} = \mathbf{p}_e$, the ASV is requested to visit first \mathbf{p}_e and afterwards \mathbf{p}_r , while if $\mathbf{p}_{closest} = \mathbf{p}_{ci}$, it has to visit \mathbf{p}_{ci} , \mathbf{p}_{ce} and \mathbf{p}_r . Finally, once that ASV reaches \mathbf{p}_r , it takes the decision according to the original NMSOA but without making the ASV go back to the points that it has already visited. We have decided to make this change to see if it could help the ASV travel a shortest distance, because it makes the ASV follow as much as possible the line that joins the intermediate points.

Another improvement that can be considered in both variants is to let the sensor take measurements along the ASV trajectory (instead of only in NMSOA intermediate points), with the purpose of trying to estimate the surface that better adjusts to the measurements taken by the ASV along its trajectory. However, as this measurements are not used for guidance, they can be obtained at any sampling rate.

3.3 Simulated Results

In this section we analyze the behaviour of the whole system when the ASV is guided towards the water body regions with maximal values of the variable/substance under study. Although this function is unknown for the GNC (except at those points on the trajectory where the measurements have been taken), we will represent its contour curves over the water body to understand better the GNC and ASV behaviors.

Figures 7(a) and 7(b) respectively show, in red, the ASV trajectories ($x(t), y(t)$) obtained when the ASV, initialized at $\mathbf{s}(0) = [200, -200, 0, 0, 0, 0]^T$, is guided by the original and modified versions of NMSOA, both started with $\mathbf{p}_1 = [322, -210]^T$ m, $\mathbf{p}_2 = [413, -238]^T$ m and $\mathbf{p}_3 = [427, -105]^T$ m. They also show the underlying triangles and intermediate points of each algorithm iteration, using the colors displayed at the legend. Under those initialization, the modified NMSOA requires more iterations (12) than the original one (8), making the length of the ASV trajectory also longer (1918 m vs 1611 m). In order to test if this behavior is observable in other simulations, we setup 50 new scenarios, randomly initializing both the ASV location (within the limits of Fig 5(b)) and original triangle (equilateral, of side $\in [100, 200]$ m, placed $[200, 300]$ m away from $\mathbf{s}(0)$) of NMSOA and run both approaches in each scenario. Next we compare their results, measuring the length of the obtained trajectories and representing in Fig 7(c) the quotient of the length obtained with the modified NMSOA between the length obtained with the original version. We can observe that this ratio is more often bigger than one, which implies that the original version produces shorter trajectories than the modified, contradicting our original thoughts.

4 GUIDING THE ASV AROUND A GIVEN CONTOUR LEVEL

The second guidance module generates waypoints that make the ASV determine the contour defined by a fixed value of the variable under study. It is inspired by the reliable PATH following algorithm (PAT, Mezher and Philippe 2000), summarized and slightly adapted in the following section. This section also shows the behavior of the whole system (GNC + ASV + Sensor) over the water body scenario of Fig. 5(b).

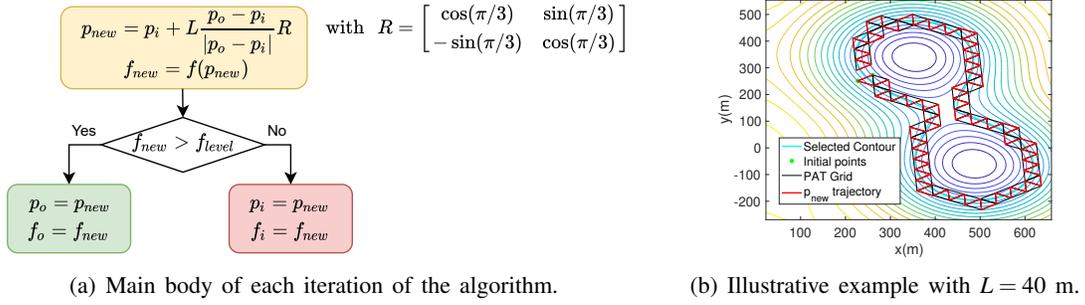


Figure 8: Reliable contour path following algorithm (PAT).

4.1 Reliable Contour PATH following algorithm (PAT)

PAT is an iterative algorithm that places a grid, formed by adjacent rotated equilateral triangles, around the values of a contour curve of a bi-dimensional unknown function. To build the grid, it adds a new vertex to the the side of the last triangle and tests if the new ending point falls inside or outside the contour (i.e. it has a function value lower or higher to the selected level). This algorithm is also a good option when the number of evaluations must be reduced and when calculating first or second order derivatives is not straightforward (as it occurs with the measurements taken from the trajectory of a single ASV).

The implementation of this algorithm is even easier. First, it needs to determine two points \mathbf{p}_i and \mathbf{p}_o of the search space, which are separated a given distance L and that fall inside and outside a contour of constant value f_{level} . Next, and as Fig. 8(a) states, it sequentially obtains a new point (\mathbf{p}_{new}), evaluates it (f_{new}), and use it to substitute the previous inside (\mathbf{p}_i) or outside (\mathbf{p}_o) point. The algorithm ends when the contour is closed because the last \mathbf{p}_{new} reaches one of the original points. Finally, the contour curve can be defined as the curve that joints the mean point of the vertices of the triangular grid that are connected and that fall in both sides of the contour. Fig. 8(b) illustrate the behavior of the algorithm, displaying in black the grid created by PAT over one of the level curves of our scenario, in green the starting \mathbf{p}_i and \mathbf{p}_o , and in red the trajectory defined by the sequence of \mathbf{p}_{new} .

4.2 Using and Modifying PAT for ASV Guidance

PAT can be used to produce the waypoints $[x_{wp}(l), y_{wp}(l)]^T$ of the guidance system. In fact, as PAT generates a unique evaluation point in each iteration, \mathbf{p}_{new} is the waypoint that should be returned by the guidance system. Again, we have to postpone its evaluation until the ASV reaches its proximity.

As ASVs can take measurements while moving, we propose the following change: to provide a new waypoint sooner, after crossing the contour curve (a fact that can be detected by the periodic measurements) and advancing at most M meters. That is, we provide a new waypoint because we have advanced M meters after crossing the contour curve or because we have almost reached the previous waypoint. We could also store the periodic measurements along the ASV trajectory to estimate $f(x, y)$ around the contour curve.

4.3 Simulated Results

In this section we analyze the behavior of the whole system when the ASV is guided to determine a contour curve of the variable measured by the sensor. Again, we draw its unknown function to understand better the GNC and ASV behaviors.

Figures 9(a) and 9(b) respectively show, in red, the ASV trajectory $(x(t), y(t))$ obtained when the ASV, initialized at $\mathbf{s}(0) = [400, -150, 0, 0, 0, 0]^T$ is guided by the original and modified versions of PAT, with $L = 30$ m and $M = L/5$. They also show the PAT grid and the estimated contour, which covers the one corresponding to the sensor simulation function. In this case, the trajectory length obtained with the original PAT is longer (4274 m) than the one obtained with the modified version (3133 m). Again, to test if the

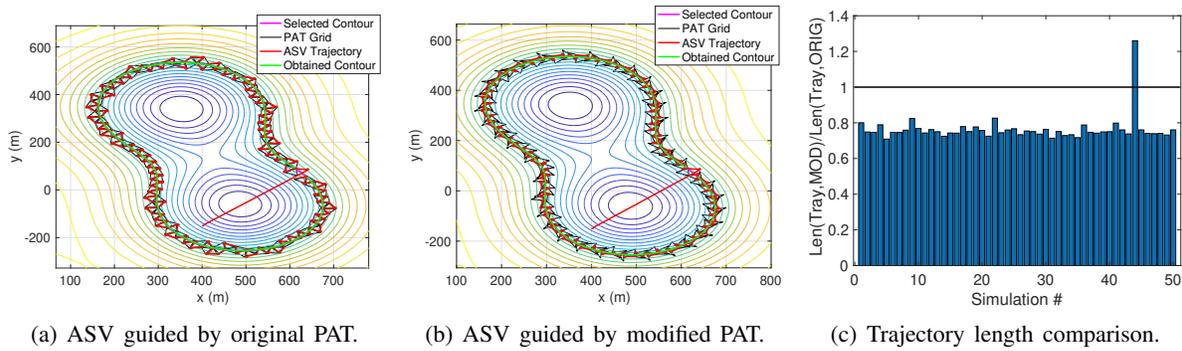


Figure 9: Simulations of the ASV guided to detect a contour level of a given variable.

behavior is systematic, we simulate 50 new configurations, varying the ASV initial location and selected level of the curve under study. Next, we run both algorithm variants over each case, measure the length of the ASV trajectories, and represent the length quotient in Fig. 9(c), which shows that the trajectory length obtained with the modified PAT is usually smaller than the one obtained with the original version.

5 CONCLUSIONS

This paper presents a new GNC especially designed to make an ASV explore a water body in order to determine the regions with extreme or constant values of a variable under study. The approach exploits two gradient-free algorithms to generate the waypoints that guide the ASV towards the ROIs. The contributions of this work are the following. First, its novelty lies on combining those algorithms with the navigation and control modules of the GNC to move the ASV gracefully between the locations that the selected algorithms suggest as measurement points. Second, the simulations are crucial to analyze the behavior of the ASV, tune the parameters of the GNC and demonstrate the viability of our approach. Moreover, the simplicity of all the elements of the GNC make it ideal to implement them in low-cost processors.

As future work we consider the following possibilities. The extension to multiple independent ASV, which is straightforward, as far as they operate in different parts of the water body. Otherwise, the system will require, at least, to incorporate a new module/behavior for avoiding ASVs collisions. We also want to use our approach to monitor multiple variable/substance (something that can be achieved by performing a weighted combination of their values into $f(x, y)$) or to avoid misleading the GNC with the sensor noise (by incorporating a filter into our system). We also want to improve the GNC further (e.g. by getting inspired by other approaches that let us generate waypoints with other purposes, or by modifying the controller and navigation modules). All these changes should be supported by further simulations before performing experiments with our ASVs, which is the final goal of our research.

ACKNOWLEDGMENTS

This work has been supported by the Spanish National Societal Challenges Program, through the AMPBAS project (RTI2018-098962-B-C21). The authors want to thank Alfonso Moneo for his collaboration in the initial simulation setup and analyses.

REFERENCES

- Arzamendia, M., D. Gregor, D. G. Reina, and S. L. Toral. 2019. "An evolutionary approach to constrained path planning of an autonomous surface vehicle for maximizing the covered area of Ypacarai Lake". *Soft Computing* 23:1723–1734.
- Arzamendia, M., D. Gregor, D. G. Reina, S. L. Toral, and R. Gregor. 2016. "Evolutionary Path Planning of an Autonomous Surface Vehicle for Water Quality Monitoring". In *Int. Conf. on Developments in e-Systems Engineering*. Aug. 31st- Sept. 2nd, Liverpool, UK.
- Blix, K. 2019. *Machine Learning Water Quality Monitoring*. Ph. D. thesis, The Artich University of Norway.

- Bu, Q., Z. Wang, and X. Tong. 2013. "An improved genetic algorithm for searching for pollution sources". *Water Science and Engineering* 6(4):392–401.
- Carazo-Barbero, G., E. Besada-Portas, J. M. Girón-Sierra, and J. A. López-Orozco. 2021. "EA-based ASV Trajectory Planner for Pollution Detection in Lentic Waters". In *EvoStar*. April 7th- 9th, Seville, Spain.
- European Commission 2021. "European Commission water related directives". https://ec.europa.eu/environment/water/index_en.htm. Accessed on March 2021.
- Fossen, T. 2002. *Marine Control Systems Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles*. 1st ed. Trondheim, Norway: Marine Cybernetics.
- Hitz, G., F. Pomerleau, M. Ève Garneau, C. Pradalier, T. Posch, J. Pernthaler, and R. Siegwart. 2012. "Design and Application of a Surface Vessel for Autonomous Inland Water Monitoring". *IEEE Robotics Automation Magazine* 19:62–72.
- Liua, Z., Y. Zhanga, X. Yua, and C. Yuana. 2016. "USVs: An overview of developments and challenges". *Annual Reviews in Control* 41.
- Mathews, J., and K. Fink. 2004. *Numerical Methods Using Matlab*. 4th ed. London: Pearson.
- Mezher, D., and B. Philippe. 2000. "PAT- a Reliable Path Following Algorithm". Technical report, INRIA.
- N. Karapetyan, J. M., and I. Rekleitis. 2019. "Meander-Based River Coverage by an Autonomous Surface Vehicle". In *Int. Conf. on Field and Service Robotics*. August 29th- 31st, Tokyo, Japan.
- Nelder, J., and R. Mead. 1965. "A Simplex Method for Function Minimization". *Computer Journal* 7:308–313.
- Peralta, F., M. Arzamendia, D. Gregor, D. Reina, and S. Toral. 2020. "A Comparison of Local Path Planning Techniques of Autonomous Surface Vehicles for Monitoring Applications: The Ypacarai Lake Case-study". *Sensors* 20.
- Shuo, J., Z. Yonghui, R. Wen, and T. Kebin. 2017. "The unmanned autonomous cruise ship for water quality monitoring and sampling". In *Int. Conf. on Computer Systems, Electronics and Control*. Dec. 25th- 27th, Dalian, China.
- Siyang, S., and T. Kerdcharoen. 2016. "Development of unmanned surface vehicle for smart water quality inspector". In *Int. Conf. on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*. June 28th-July 1st, Chiang Mai, Thailand.
- Storey, M., B. van der Gaag, and B. Burns. 2011. "Advances in on-line drinking water quality monitoring and early warning systems". *Water Research* 42(2):741–747.
- United Nations 2021. "2030 Agenda for Sustainable Development". <https://sdgs.un.org/goals>. Accessed on March 2021.
- United States Environmental Protection Agency 2021. "Drinking Water Requirements for States and Public Water Systems". <https://www.epa.gov/dwreginfo/drinking-water-regulations>. Accessed on March 2021.
- Valada, A., P. Velagapudi, B. Kannan, C. Tomaszewski, G. Kantor, and P. Scerri. 2012. "Development of a Low Cost Multi-Robot Autonomous Marine Surface Platform". In *Int. Conf. on Field and Service Robotics*. July 16th- 19th, Matsushima, Japan.
- Visioli, A. 2006. *Practical PID Control*. 2006 ed. London: Springer-Verlag.
- Xia, G., Z. Han, B. Zhao, C. Liu, and X. Wang. 2019. "Global Path Planning for Unmanned Surface Vehicle Based on Improved Quantum Ant Colony Algorithm". *Mathematical Problems in Engineering* 2019:10 pages.
- Xiong, C., D. Chen, D. Lu, Z. Zeng, and L. Lian. 2019. "Path planning of multiple autonomous marine vehicles for adaptive sampling using Voronoi-based ant colony optimization". *Robotics and Autonomous Systems* 115:90 – 103.
- Xiong, C., H. Zhou, D. Lu, Z. Zeng, L. Lian, and C. Yu. 2020. "Rapidly-Exploring Adaptive Sampling Tree*: A Sample-Based Path-Planning Algorithm for Unmanned Marine Vehicles Information Gathering in Variable Ocean Environments". *Sensors* 20(9):18 pages.

AUTHOR BIOGRAPHIES

EVA BESADA-PORTAS is an Associate Professor of Systems Engineering and Automation at University Complutense of Madrid. She holds a PhD in Computer Systems from the same university. Her research interests include uncertainty modeling and simulation, optimal control and planning of unmanned vehicles. Her email address is ebesada@ucm.es.

JOSÉ M. GIRÓN-SIERRA is an Emeritus Professor of Systems Engineering and Automation at University Complutense of Madrid. He holds a PhD in in Physics (Electronics) from the same university. His research interests include unmanned vehicles, marine robotics, spatial systems, intelligent systems and optimisation.. His email address is gironsi@ucm.es.

JUAN JIMÉNEZ is an Associate Professor of Systems Engineering and Automation at University Complutense of Madrid. He holds a PhD in Computer Systems from Universidad Nacional de Educación a Distancia (UNED). His research interests include unmanned vehicles and cooperative control for multiagent systems. His email address is juan.jimenez@fis.ucm.es.

JOSÉ A. LOPEZ-OROZCO is a Full Professor in the University Complutense of Madrid. He holds a Ph.D. in Physics from the same University. His research interests include multisensor data fusion, control and planning of unmanned vehicles, and robotics. His email address is jalo@ucm.es.