

Cooperative Motion Control of Aerial and Marine Vehicles for Environmental Applications

Marcelo Jacinto, António Pascoal and Rita Cunha

Laboratory of Robotics and Systems in Engineering and Science (LARSyS),

ISR/IST, University of Lisbon, Lisbon, Portugal.

Email: marcelo.jacinto@tecnico.ulisboa.pt

Abstract—This work addresses the problem of formation control of a quadrotor and one (or more) marine vehicles operating at the surface of the water with the end goal of encircling the boundary of a chemical spill. Firstly, the mathematical models of the Medusa class of marine robots, and quadrotor aircrafts are introduced, followed by the design of single vehicle motion controllers that allow these vehicles to follow a parameterized path individually. Inner-outer loop schemes coupled with Lyapunov based techniques are used for control design. At a second stage, a distributed coordination controller using event triggered communications is introduced, enabling the vehicles to perform cooperative path following missions according to a pre-defined geometric formation. In the next step, a real time path planning algorithm is developed that makes use of a camera sensor, installed on-board the quadrotor. This sensor enables the detection in the image of which pixels encode parts of a chemical spill boundary and use them to generate and update in real time a set of smooth B-spline based paths for all the vehicles to follow cooperatively. The performance of the complete system is evaluated by resorting to 3-D simulation software, making it possible to simulate visually a chemical spill. Results from real water trials are also provided for parts of the system, where two Medusa vehicles are required to perform a static lawn-mowing path following mission cooperatively at the surface of the water.

Index Terms—Quadrotor control, Autonomous Surface Vehicle control, Cooperative Path Following, Environmental Boundary Following

I. INTRODUCTION

The ocean covers around 361M km² of the Earth's surface [1] and there is evidence that it was at its bottom that the first primordial cells have formed, about 3.8 to 4 billion years ago [2]. It also plays a key roll in our modern society, being a source of food and sustainable energy that powers millions of homes [3]. Unfortunately, this vast habitat is also known for environmental disasters, some as a direct consequence of human behaviour, such as oil spills or ocean waste disposal and others as an indirect consequence, such as global warming and the rise of seawater levels. These catastrophes represent a major threat to wild life, and as a consequence a threat to humans. In the case of oil spills or waste disposals, surveillance as well as cleanup missions must be carried out

This work was partially supported by the H2020 EU Marine Robotics Research Infrastructure Network (Project ID 731103) and, the H2020-EU.1.2.2 FET Proactive RAMONES project (Grant agreement ID: 101017808). Funded was also provided by FCT, FEDER, and ERDF through the LARSyS UIDB/50009/2020, LISBOA-01-0145-FEDER-031411, and EMSO-PT projects.

in order to restore these environments to their previous states. These operations are expensive to conduct and require the use of huge vessels with specialized staff on board to conduct them. In the case of oil spills, these operations usually resort to skimmers used of off boats in order to "skim" the oil from the sea surface, together with chemical dispersants to break up the oil molecules.

Recent years have seen a huge development in computing power and miniaturization of sensors which have enabled the development of very efficient robots that can sweep through the sea at a relative small cost when compared to the current alternatives - large ships that are loud and very disruptive. These robots are usually known as Autonomous Underwater Vehicles (AUV) or/and Autonomous Surface Vehicle (ASV) when working only at the surface of water. In addition to these, there has been recently a growing interest on the development of miniaturized aircrafts denominated Unmanned Aerial Vehicles (UAV) which are usually equipped with camera sensors allowing them to have a top-down view of the environment.

Motivated by the all of those factors, the aim of this work is to develop a set of tools that allow an UAV-quadrotor and multiple ASVs to perform a surveillance mission cooperatively, where the main goal is to detect and follow closely an environmental boundary¹, such as an oil spill, at the surface of the water, according to Figure 1.

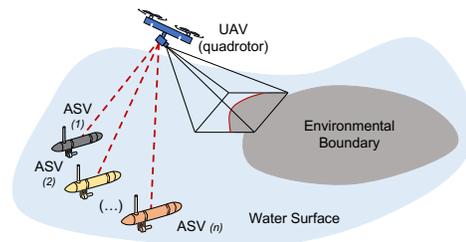


Fig. 1. Cooperative path following on an environmental boundary

II. VEHICLE MODELS

The terminology regarding coordinates and reference frames adopted for both vehicles is depicted in Figure 2. The *Inertial Reference Frame* $\{U\}$ follows the North-East-Down (NED)

¹The term environmental boundary used in this context denotes any hazardous spread of contaminants, pollutants, etc. that generate anisotropic changes in the environment, for which a clear perimeter can be defined.

convention and the *Body Reference Frame* $\{B\}$ is attached to each vehicle's center of mass.

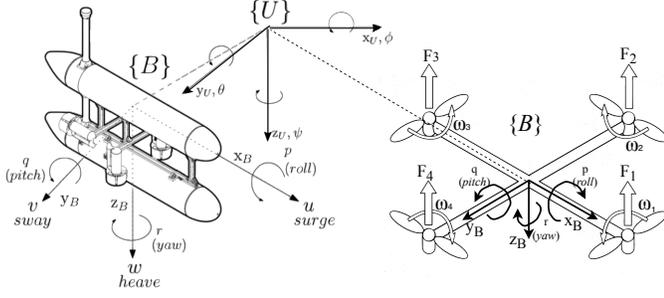


Fig. 2. Adopted reference frames (adapted from Teixeira et al. [4] and Luukkonen T. [5])

A. ASV Equations of Motion

The kinematic equations of motion for the ASV are given by

$$\begin{aligned} \dot{\mathbf{p}} &= {}^U_B R(\psi) \mathbf{v} + \mathbf{v}_c \\ \dot{\psi} &= r, \end{aligned} \quad (1)$$

where $\mathbf{p} := [x, y]^T$ denotes the ASV position expressed in $\{U\}$, $\mathbf{v} := [u, v]^T$ the body-velocity vector, ${}^U_B R(\psi) \in \mathbb{R}^{2 \times 2}$ rotation matrix and $\mathbf{v}_c := [v_c, v_y]^T$ the ocean current expressed in $\{U\}$. The dynamics equations are given by

$$\begin{aligned} m_u \dot{u} - m_v vr + d_u u &= \tau_u \\ m_v \dot{v} + m_u ur + d_v v &= 0 \\ m_r \dot{r} - m_{uv} uv + d_r r &= \tau_r \end{aligned} \quad (2)$$

where τ_u is the external force in surge (common mode), τ_r is the external torque about the Z-axis (differential mode), $m_u := m - X_{\dot{u}}$, $m_v := m - Y_{\dot{v}}$, $m_r := I_z - N_{\dot{r}}$ and $m_{uv} := m_u - m_v$ represent the mass and hydrodynamic added mass and $d_u := -X_u - X_{|u|u}|u|$, $d_v := -Y_v - Y_{|v|v}|v|$ and $d_r := -N_r - N_{|r|r}|r|$ the hydrodynamic damping effects.

B. Quadrotor Equations of Motion

An UAV-quadrotor can be described by the kinematics equations:

$$\dot{\mathbf{p}} := \dot{\boldsymbol{\eta}}_1 = {}^U_B R(\boldsymbol{\eta}_2) \mathbf{v}_1 \quad (3)$$

$$\dot{\boldsymbol{\eta}}_2 = Q(\boldsymbol{\eta}_2) \mathbf{v}_2 \quad (4)$$

where $\mathbf{p} := \boldsymbol{\eta}_1 = [x, y, z]^T$ denotes the quadrotor's position expressed in $\{U\}$, $\boldsymbol{\eta}_2 := [\phi, \theta, \psi]^T$ denotes the orientation vector of $\{B\}$ expressed in $\{U\}$. The linear velocity $\mathbf{v}_1 \in \mathbb{R}^3$ and angular velocity velocity $\mathbf{v}_2 \in \mathbb{R}^3$ are expressed in $\{B\}$. The rotation matrix is denoted by ${}^U_B R(\boldsymbol{\eta}_2) \in \mathbb{R}^{3 \times 3}$ and the angular transformation matrix by $Q(\boldsymbol{\eta}_2)$. The dynamic equations are given by

$$\ddot{\mathbf{p}} := \ddot{\boldsymbol{\eta}}_1 = g \mathbf{e}_3 - \frac{1}{m} {}^U_B R(\boldsymbol{\eta}_2) Z \mathbf{e}_3 \quad (5)$$

$$\dot{\mathbf{v}}_2 = -\mathbf{J}^{-1}(\mathbf{v}_2 \times \mathbf{J} \mathbf{v}_2) + \mathbf{J}^{-1} N_{RB} \quad (6)$$

where $\mathbf{e}_3 = [0, 0, 1]^T$, g the gravitational acceleration, m the mass of the vehicle, $\mathbf{J} \in \mathbb{R}^{3 \times 3}$ the inertia matrix, $Z \in \mathbb{R}$ the total thrust force and $N_{RB} \in \mathbb{R}^3$ the torque vector.

III. INNER-LOOP DESIGN

A. ASV Surge Speed and Yaw-rate Control

Problem III.1. Consider the ASV with dynamics described by (2) and let $\mathbf{u}_d^{\dagger} = [u_d, r_d]^T \in \mathbb{R}^2$ denote the desired surge speed and yaw-rate respectively. Design a linear control law for the force and torque τ_u and τ_r such that $\mathbf{u}^{\dagger} = [u, r]$ converges to a desired set of surge and yaw-rate references \mathbf{u}_d^{\dagger} .

Assumption III.1. The sway-motion of the ASV is negligible, i.e. $v \approx 0$.

The proposed inner-loop controllers are given by Proportional Integral (PI) control laws with feed-forward terms, according to:

$$\begin{aligned} \tau_u &= d_u u + m_u \left[-k_p e(t) - k_i \int_0^t e(\tau) d\tau \right], \\ \tau_r &= d_r r + m_r \left[-k_p e(t) - k_i \int_0^t e(\tau) d\tau \right], \end{aligned} \quad (7)$$

with $k_p, k_i > 0$ proportional and integral constants.

B. Ocean Currents Observer

Problem III.2. Consider that the ASV is equipped with a Doppler Velocity Logger (DVL) capable of providing the vehicle's relative velocity to the water, expressed in $\{B\}$ and a DGPS unit which provides measurements of the position of the vehicle \mathbf{p}_m , expressed in $\{U\}$. Furthermore, consider that it is possible to express the velocities provided by the DVL in $\{U\}$, by resorting to the rotation matrix $R(\psi)$, as \mathbf{v}_m . Develop an estimator for the ocean-currents \mathbf{v}_c expressed in $\{U\}$, assumed to be constant and irrotational, i.e. $\dot{\mathbf{v}}_c = 0$.

In order to solve this problem, we borrow the results from Pascoal et al. [6] and Sanches et al. [7] where the authors propose a time-varying complementary filter structure.

Proposition III.1. Consider the process model \mathcal{M}_p given by

$$\mathcal{M}_p := \begin{cases} \dot{\mathbf{p}} = R(\psi) \mathbf{v} + \mathbf{v}_c \\ \mathbf{v}_m = R(\psi) \mathbf{v} \\ \mathbf{p}_m = \mathbf{p} \end{cases} \quad (8)$$

and the candidate complementary filter model described by

$$\mathcal{F} := \begin{cases} \dot{\hat{\mathbf{p}}} = k_1(\mathbf{p}_m - \hat{\mathbf{p}}) + \mathbf{v}_m + \hat{\mathbf{v}}_c \\ \dot{\hat{\mathbf{v}}}_c = k_2(\mathbf{p}_m - \hat{\mathbf{p}}) \end{cases} \quad (9)$$

with $k_1, k_2 > 0$ constants. The proposed complementary filter is asymptotically stable and solves problem III.2.

C. Quadrotor Inner-loop Control

Problem III.3. Consider the AUV-quadrotor with rotational dynamics given by (6) and let $\boldsymbol{\eta}_{2d} = [\phi_d, \theta_d, \psi_d] \in \mathbb{R}^3$ denote the desired roll, pitch and yaw angles respectively. Design a linear control law for the external torque N_{RB} such that $\boldsymbol{\eta}_2$ converges to desired set of angle references $\boldsymbol{\eta}_{2d}$.

Assumption III.2. The drone's inertia matrix \mathbf{J} is diagonal.

Assumption III.3. The vehicle is working near its hover state, where $\phi \approx \theta \approx 0^\circ$.

Near the hover state, the quadrotor's linearized angular acceleration is described by

$$\ddot{\boldsymbol{\eta}}_2 = J^{-1} N_{RB}, \quad (10)$$

such that $N_{RB} = J \cdot \mathbf{u}_d^\ddagger$, where \mathbf{u}_d^\ddagger represents the control input. The proposed Proportional Derivative (PD) control law is given by

$$\mathbf{u}_d^\ddagger(t) = -K_p(\boldsymbol{\eta}_2(t) - \boldsymbol{\eta}_{2d}(t)) - K_d(\dot{\boldsymbol{\eta}}_2(t) - \dot{\boldsymbol{\eta}}_{2d}(t)), \quad (11)$$

with $K_p, K_d > 0$ the proportional and derivative gain matrices.

D. Generating angle references from accelerations

Consider the double integrator model of the vehicle given by (5) where $\tilde{\mathbf{p}} = \mathbf{u}^\circ$ is a desired control input given by the auto-pilot/outer-loop. It is necessary to develop a sub-system capable of computing the total thrust Z and the attitude associated with the matrix $R(\boldsymbol{\eta}_2)$ from the desired input of the auto-pilot. Consider expanding equation (5):

$$\mathbf{u}^\circ = -\frac{1}{m} R_z(\psi_{des}) [R_y(\theta) R_x(\phi) Z \mathbf{e}_3] + g \mathbf{e}_3. \quad (12)$$

Furthermore, consider \mathbf{u}^* to be given by

$$\mathbf{u}^* := R_y(\theta) R_x(\phi) Z \mathbf{e}_3, \quad (13)$$

where $\mathbf{u}^* = [u_1^*, u_2^*, u_3^*]^T$. Replacing (13) in (12) yields the relation

$$\mathbf{u}^* = -m R_z^T(\psi_{des}) (\mathbf{u}^\circ - g \mathbf{e}_3). \quad (14)$$

From (14) it is possible to compute \mathbf{u}^* , from the desired output of the outer-loop. It is also possible to infer that the total thrust to be given by $Z := \|\mathbf{u}^*\|$ and

$$\phi_{des} = \arcsin(-u_2^*/Z), \quad (15)$$

$$\theta_{des} = \arctan(u_1^*/u_3^*), \quad (16)$$

with u_3^* assumed to be different than zero.

IV. PATH FOLLOWING

Consider, for both vehicles that there exist virtual targets moving along a desired path for which the vehicles must converge to. The speed profile for a virtual target that moves along a desired path is given by

$$v_d(\gamma, t) := v_L(\gamma) + v_{coord}(t), \text{ with } |v_L(\gamma)| \leq v_L^{max}, \quad (17)$$

where $v_L(\gamma)$ is a desired speed profile as a function of the path, v_L^{max} a pre-defined speed upper-bound and $v_{coord}(t)$ a speed coordination term that will be used later for enabling Cooperative Path Following (CPF).

A. ASV Path Following

Problem IV.1. Let $\mathbf{p}_d(\gamma) : [0, \infty) \rightarrow \mathbb{R}^2$ denote the desired path parameterised by a continuous variable $\gamma \in \mathbb{R}$ and $v_d(\gamma, t) \in \mathbb{R}$ be a desired speed profile for a virtual target moving along the desired path. Furthermore, consider $\mathbf{p}_d(\gamma)$ to be C^2 and have its first and second derivatives with respect to γ bounded. Moreover, the vehicle is equipped with an inner-loop controller that given a desired surge speed and yaw-rate $\mathbf{u}_d = [u_d, r_d]^T$, assumed to be bounded, computes a set of desired thrust and torque to apply to the vehicle. Design a control law for surge u_d , yaw-rate r_d and virtual target $\dot{\gamma}$ such that:

- the vehicle's position converges to a tube around the desired position that can be made arbitrarily small, i.e.

$\|\mathbf{p}(t) - \mathbf{p}_d(\gamma)\|$ converges to a neighbourhood of the origin;

- the speed of the virtual target moving along the path converges to the desired speed profile, i.e. $|\dot{\gamma} - v_d(\gamma, t)| \rightarrow 0$ as $t \rightarrow \infty$.

Following the approach proposed by Aguiar et al. [8], [9] and [10], consider the global diffeomorphic coordinate transformation which expresses the position error defined in the body-frame of the vehicle $\{B\}$ as

$$\mathbf{e}_p(t) := \frac{B}{U} R(\psi)(\mathbf{p}(t) - \mathbf{p}_d(\gamma)), \quad (18)$$

and let the speed-tracking error be defined as

$$e_\gamma := \dot{\gamma} - v_d(\gamma, t). \quad (19)$$

The body-fixed position error dynamics can be given by:

$$\begin{aligned} \dot{\mathbf{e}}_p(t) &= \frac{B}{U} \dot{R}(\psi)(\mathbf{p}(t) - \mathbf{p}_d(\gamma)) + \frac{B}{U} R(\psi)(\dot{\mathbf{p}}(t) - \dot{\mathbf{p}}_d(\gamma)) \\ &= -S(r) \frac{B}{U} \mathbf{e}_p(t) + \mathbf{v} + \mathbf{v}_c - \frac{B}{U} R(\psi) \frac{\partial \mathbf{p}_d(\gamma)}{\partial \gamma} \dot{\gamma} \\ &= -S(r)(\mathbf{e}_p - \boldsymbol{\delta}) + \Delta \mathbf{u} + \begin{bmatrix} 0 \\ v \end{bmatrix} + \mathbf{v}_c - \frac{B}{U} R(\psi) \frac{\partial \mathbf{p}_d(\gamma)}{\partial \gamma} \dot{\gamma} \end{aligned} \quad (20)$$

with

$$S(r) = \begin{bmatrix} 0 & -r \\ r & 0 \end{bmatrix}, \quad (21)$$

$\boldsymbol{\delta} = [0, \delta]^T$, $\Delta = \text{diag}(1, -\delta)$ with $\delta < 0$ and $\mathbf{u} = [u, r]^T$ the system input. Consider the ocean currents estimate and estimation error expressed in the body frame, according to:

$$\hat{\mathbf{v}}_c := \frac{B}{U} R(\psi) \hat{\mathbf{v}}_c \text{ and } \tilde{\mathbf{v}}_c := \mathbf{v}_c - \hat{\mathbf{v}}_c, \quad (22)$$

and inner-loop tracking error given by

$$\tilde{\mathbf{u}} := \mathbf{u} - \mathbf{u}_d. \quad (23)$$

Proposition IV.1. Consider the system error dynamics described by equations (19) and (20), along with virtual target dynamics proposed in (25). Furthermore, consider the control laws given by

$$\mathbf{u}_d := \Delta^{-1} \left(-K_p \boldsymbol{\sigma}(\mathbf{e}_p - \boldsymbol{\delta}) - \begin{bmatrix} 0 \\ v \end{bmatrix} - \hat{\mathbf{v}}_c + \frac{B}{U} R(\psi) \frac{\partial \mathbf{p}_d(\gamma)}{\partial \gamma} v_d \right), \quad (24)$$

$$\dot{\gamma} := -k_\gamma e_\gamma + \dot{v}_d(\gamma, t) + (\mathbf{e}_p - \boldsymbol{\delta})^T \frac{B}{U} R(\psi) \frac{\partial \mathbf{p}_d(\gamma)}{\partial \gamma}, \quad (25)$$

where $K_p \geq 0$, $k_\gamma > 0$ and $\boldsymbol{\sigma}(\mathbf{e}_p)$ is a saturation function. The closed-loop system is ISS with respect to $\Delta \tilde{\mathbf{u}} + \tilde{\mathbf{v}}_c$.

Proof. Consider a candidate Lyapunov function:

$$V(\mathbf{e}_p, e_\gamma) = \frac{1}{2} (\mathbf{e}_p - \boldsymbol{\delta})^T (\mathbf{e}_p - \boldsymbol{\delta}) + e_\gamma^2. \quad (26)$$

Take the first derivative of the candidate Lyapunov function:

$$\begin{aligned} \dot{V} &= -(\mathbf{e}_p - \boldsymbol{\delta})^T K_p \boldsymbol{\sigma}(\mathbf{e}_p - \boldsymbol{\delta}) - k_\gamma e_\gamma + (\mathbf{e}_p - \boldsymbol{\delta})^T (\Delta \tilde{\mathbf{u}} + \tilde{\mathbf{v}}_c) \\ &\leq -(1 - \theta + \theta) (\mathbf{e}_p - \boldsymbol{\delta})^T K_p \boldsymbol{\sigma}(\mathbf{e}_p - \boldsymbol{\delta}) - k_\gamma |e_\gamma|^2 \\ &\quad + \|\mathbf{e}_p - \boldsymbol{\delta}\| \|\Delta \tilde{\mathbf{u}} + \tilde{\mathbf{v}}_c\|, \end{aligned} \quad (27)$$

where $0 < \theta < 1$. The term

$$-\theta (\mathbf{e}_p - \boldsymbol{\delta})^T K_p \boldsymbol{\sigma}(\mathbf{e}_p - \boldsymbol{\delta}) + \|\mathbf{e}_p - \boldsymbol{\delta}\| \|\Delta \tilde{\mathbf{u}} + \tilde{\mathbf{v}}_c\| \quad (28)$$

will be ≤ 0 if

$$\theta \lambda_{\min}(K_p) \boldsymbol{\sigma}(\|\mathbf{e}_p - \boldsymbol{\delta}\|) \geq \|\Delta \tilde{\mathbf{u}} + \tilde{\mathbf{v}}_c\|, \quad (29)$$

which in turn implies that

$$\|\mathbf{e}_p - \boldsymbol{\delta}\| \geq \boldsymbol{\sigma}^{-1} \left(\frac{1}{\theta \lambda_{\min}(K_p)} \|\Delta \tilde{\mathbf{u}} + \tilde{\mathbf{v}}_c\| \right), \quad (30)$$

and

$$\dot{V}_2 \leq -(1 - \theta)(\mathbf{e}_p - \boldsymbol{\delta})^T K_p \boldsymbol{\sigma}(\mathbf{e}_p - \boldsymbol{\delta}) - k_\gamma |e_\gamma|^2, \quad (31)$$

as the right side of inequality (30) can be made arbitrarily small by controlling the gain matrix K_p . \square

Problem IV.2. Consider a quadrotor-UAV described by

$$\ddot{\mathbf{p}} := \mathbf{u}^\diamond + \mathbf{d}, \quad (32)$$

where $\mathbf{d} \in \mathbb{R}^3$ represents unmeasured external constant disturbances acting on the vehicle, such that

$$\|\mathbf{d}\| \leq d_{max}, \quad (33)$$

where d_{max} is a known, positive constant. Let $\mathbf{p}_d(\gamma) : [0, \infty) \rightarrow \mathbb{R}^3$ denote the desired path parameterised by a continuous variable $\gamma \in \mathbb{R}$ and $v_d(\gamma, t) \in \mathbb{R}$ be a desired speed profile for a virtual target moving along the desired path. Furthermore, consider $\mathbf{p}_d(\gamma)$ to be C^2 and have its first and second derivatives with respect to γ bounded. Design a control law for the quadrotor acceleration and virtual target such that:

- the vehicle's position converges to a tube around the desired position that can be made arbitrarily small, i.e. $\|\mathbf{p}(t) - \mathbf{p}_d(\gamma)\|$ converges to a neighbourhood of the origin;
- the speed of the virtual target moving along the path converges to the desired speed profile, i.e. $|\dot{\gamma} - v_d(\gamma, t)| \rightarrow 0$ as $t \rightarrow \infty$.

Following a similar approach to the one proposed by F. Vanni and P. Aguiar, consider the position and velocity errors, but this time defined in the inertial frame $\{U\}$ as

$$\mathbf{e}_p := \mathbf{p}(t) - \mathbf{p}_d(\gamma), \quad (34)$$

$$\mathbf{e}_v := \dot{\mathbf{p}} - \frac{\partial \mathbf{p}_d}{\partial \gamma} v_d, \quad (35)$$

respectively and a virtual target speed tracking error to be defined by (19). Moreover, consider the speed tracking error (19) and a new auxililar error \mathbf{z} defined as

$$\mathbf{z} := \mathbf{e}_v + K_1 \mathbf{e}_p, \text{ with } K_1 \succeq 0. \quad (36)$$

The position and velocity error dynamics can be given by

$$\dot{\mathbf{e}}_p = \dot{\mathbf{p}} - \frac{\partial \mathbf{p}_d}{\partial \gamma} \dot{\gamma}, \quad (37)$$

$$\dot{\mathbf{e}}_v = \ddot{\mathbf{p}} - \frac{d}{dt} \left(\frac{\partial \mathbf{p}_d}{\partial \gamma} v_d \right). \quad (38)$$

Furthermore, consider the time derivative introduced in (38), the desired virtual target speed function (17) and virtual target

speed tracking error function (19), then we can expand the expression as

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial \mathbf{p}_d}{\partial \gamma} v_d \right) &= \frac{d}{dt} \left(\frac{\partial \mathbf{p}_d}{\partial \gamma} \right) v_d + \frac{\partial \mathbf{p}_d}{\partial \gamma} \frac{d}{dt} v_d \\ &= \underbrace{\left[\frac{\partial^2 \mathbf{p}_d}{\partial \gamma^2} v_d + \frac{\partial \mathbf{p}_d}{\partial \gamma} \frac{\partial v_d(\gamma)}{\partial \gamma} \right]}_{h(\gamma)} (e_\gamma + v_d) + \frac{\partial \mathbf{p}_d}{\partial \gamma} \dot{v}_{coord} \\ &= h(\gamma)(e_\gamma + v_d(\gamma, t)) + \frac{\partial \mathbf{p}_d}{\partial \gamma} \dot{v}_{coord}. \end{aligned} \quad (39)$$

Replacing (32) and (39) in (38) yields

$$\dot{\mathbf{e}}_v = \mathbf{u}^\diamond + \mathbf{d} - h(\gamma)(e_\gamma + v_d) - \frac{\partial \mathbf{p}_d}{\partial \gamma} \dot{v}_{coord}. \quad (40)$$

Let the disturbance estimation error be given by:

$$\tilde{\mathbf{d}} := \mathbf{d} - \hat{\mathbf{d}}. \quad (41)$$

In Cabecinhas et al. [11] the authors propose the use of a simple observer based on a smooth projection operator for disturbance rejection on a quadrotor. Consider the following disturbance observer:

$$\dot{\hat{\mathbf{d}}} := K_d \text{Proj}(\mathbf{z}, \hat{\mathbf{d}}) \text{ with } K_d \succeq 0, \quad (42)$$

where $\text{Proj}(\cdot)$ denotes a smooth projection operator given by

$$\text{Proj}(\boldsymbol{\mu}, \hat{\boldsymbol{\theta}}) = \boldsymbol{\mu} - \frac{\eta_1 \eta_2}{4(\varepsilon^2 + 2\varepsilon\theta_0)^{n+1} \theta_0^2} \nabla \mathbf{p}_d(\hat{\boldsymbol{\theta}}), \quad (43)$$

where

$$p_d(\hat{\boldsymbol{\theta}}) = \hat{\boldsymbol{\theta}}^T \hat{\boldsymbol{\theta}} - \theta_0^2, \quad (44)$$

$$\eta_1 = \begin{cases} p_d^{n+1}(\hat{\boldsymbol{\theta}}), & \text{if } p_d(\hat{\boldsymbol{\theta}}) \geq 0 \\ 0, & \text{otherwise} \end{cases}, \quad (45)$$

$$\eta_2 = \frac{1}{2} \hat{\boldsymbol{\theta}}^T \boldsymbol{\mu} + \sqrt{\left(\frac{1}{2} \hat{\boldsymbol{\theta}}^T \boldsymbol{\mu} \right)^2 + \delta^2}, \quad (46)$$

and $\boldsymbol{\mu}(t) \in \mathbb{R}^p$ is a known, n times continuously differentiable (C^n) variable, ε and δ are arbitrary positive constants. This projection operator enjoys the following property:

$$\tilde{\boldsymbol{\theta}}^T \text{Proj}(\boldsymbol{\mu}, \hat{\boldsymbol{\theta}}) \geq \tilde{\boldsymbol{\theta}}^T \boldsymbol{\mu}. \quad (47)$$

The estimation error dynamics are given by

$$\dot{\tilde{\mathbf{d}}} = -\tilde{\mathbf{d}}. \quad (48)$$

Moreover, consider an inner-loop tracking error given by

$$\tilde{\mathbf{u}}^\diamond := \mathbf{u}^\diamond - \mathbf{u}_d^\diamond, \quad (49)$$

assumed to be bounded.

Proposition IV.2. Consider the double integrator system described by (32) with outer-loop control laws given by

$$\mathbf{u}_d^\diamond := -\hat{\mathbf{d}} + h(\gamma)v_d(\gamma, t) + \frac{\partial \mathbf{p}_d}{\partial \gamma} \dot{v}_{coord} - K_1 \mathbf{e}_v - \mathbf{e}_p - K_2 \mathbf{z}. \quad (50)$$

$$\ddot{\gamma} := -k_\gamma e_\gamma + \dot{v}_d(\gamma, t) + \mathbf{e}_p^T \frac{\partial \mathbf{p}_d}{\partial \gamma} + \mathbf{z}^T \left(h(\gamma) + K_1 \frac{\partial \mathbf{p}_d}{\partial \gamma} \right), \quad (51)$$

where $K_1, K_2 \succeq 0$ and k_γ a positive gain. The proposed control laws solve the problem IV.2.

Proof. Consider the candidate Lyapunov function

$$V = \frac{1}{2} \mathbf{e}_p^T \mathbf{e}_p + \frac{1}{2} \mathbf{z}^T \mathbf{z} + \frac{1}{2} e_\gamma^2 + \frac{1}{2} \tilde{\mathbf{d}}^T K_d^{-1} \tilde{\mathbf{d}}. \quad (52)$$

Take its time derivative along with the previously introduced equations. Then

$$\dot{V} = -\mathbf{e}_p^T K_1 \mathbf{e}_p - \mathbf{z}^T K_2 \mathbf{z} - k_\gamma e_\gamma^2 + \underbrace{\tilde{\mathbf{d}}^T (\mathbf{z} - \text{Proj}(\mathbf{z}, \hat{\mathbf{d}}))}_{\leq 0} + \mathbf{z}^T \tilde{\mathbf{u}}^\circ. \quad (53)$$

Making use of the property of the smooth projection operator, it is possible to derive a bound for the derivative

$$\dot{V} \leq -W(\mathbf{e}_p, \mathbf{e}_v, e_\gamma) + \mathbf{z}^T \tilde{\mathbf{u}}^\circ, \quad (54)$$

From combining the relation (5), (49) and (50), the input is bounded by

$$\tilde{\mathbf{u}}^\circ \leq \|\tilde{\mathbf{r}}_3\| \left(\|\hat{\mathbf{d}}\| + \|\mathbf{h}(\gamma) v_d(\gamma, t)\| + \left\| \frac{\partial \mathbf{p}_d}{\partial \gamma} \dot{v}_{coord}(t) \right\| + \|K_1\| \|\mathbf{e}_v\| + \|\mathbf{e}_p\| + \|K_2\| \|\mathbf{z}\| + g \right). \quad (55)$$

It is only possible to conclude that as long as the position and velocity errors are small, and the inner-loop of the system is much faster than the outer-loop (guaranteeing that $\|\tilde{\mathbf{r}}_3\|$ is small), then the system is able to converge to a neighborhood of the desired references. \square

In order to make the designed control law \mathbf{u}_d more readable, consider the following algebraic manipulation:

$$\mathbf{u}_d^\circ = -\hat{\mathbf{d}} + \underbrace{h(\gamma) v_d + \frac{\partial \mathbf{p}_d}{\partial \gamma} \dot{v}_{coord}(t)}_{\text{acceleration term}} - \mathbf{e}_v \underbrace{(K_1 + K_2)}_{K_v} - \mathbf{e}_p \underbrace{(I + K_1 K_2)}_{K_p}. \quad (56)$$

V. COOPERATIVE PATH FOLLOWING

Consider a group of $N \in \mathbb{R}^+ \setminus \{1\}$ vehicles in a network that can be described by a digraph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$, consisting on N vertices \mathcal{V} , a set of directed edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, where the edge ε_{ij} represents the flow of information from agent i to agent j , and a weighted adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$. Furthermore, each vehicle i is able to receive information from its neighbours in \mathcal{N}_i^{in} and send information to its neighbours in \mathcal{N}_i^{out} . Let the state vector of the system be composed by the path parameter of each individual vehicle $\gamma = [\gamma_1, \dots, \gamma_N]^T$. The CPF problem is formulated in problem V.1.

Problem V.1. For each agent i , with $i = 1, \dots, N$ derive a consensus protocol for the speed correction term $\mathbf{v}^{coord} = [v_1^{coord}, \dots, v_N^{coord}]^T$ such that $\lim_{t \rightarrow \infty} |\gamma_i - \gamma_j| = 0, \forall j \in \mathcal{N}_i^{in}$, and the formation of vehicles achieves the desired speed assignment $\mathbf{v}_L(\gamma)$ as $t \rightarrow \infty$.

In order to solve problem V.1 the following simplifying assumption is taken:

Assumption V.1. The communication topology of the vehicles is fixed, i.e. the Laplacian matrix L associated to the graph \mathcal{G} is constant. \mathcal{G} is also undirected, i.e. $\mathcal{N}_i^{in} = \mathcal{N}_i^{out}$, and connected.

Assumption V.2. The speed progression of a virtual targets along the desired paths is always assumed to be modelled by a single integrator system $\dot{\gamma} = \mathbf{v}_d = \mathbf{v}_L(\gamma) + \mathbf{v}^{coord}$.

Let the synchronization error vector be defined as $\varepsilon = [\varepsilon_1, \dots, \varepsilon_N]^T$, such that

$$\varepsilon_i := \sum_{j \in \mathcal{N}_i^{in}} a_{ij} (\gamma_i - \gamma_j), \quad (57)$$

where ε_i denotes the coordination error between vehicle i and its neighbours. This can also be expressed in it's vectorial form as

$$\varepsilon := L\gamma \text{ with } L = D - \mathcal{A}, \quad (58)$$

where L is the Laplacian matrix of graph \mathcal{G} obtained according to:

$$D = \text{diag}(d_i^{in}), \text{ with } d_i^{in} = \sum_{j \in \mathcal{N}_i^{in}} a_{ij}. \quad (59)$$

such that $L\mathbf{1} = \mathbf{0}$ and L has a simple eigenvalue at zero associated with eigen vector $\mathbf{1}$. In this section a distributed control scheme with Event-Triggered Communications (ETC) is presented, based on previous work developed by A. Aguiar and A. Pascoal [12] and N. Hung and F. Rego [13]. In their work, the authors propose a scheme where each agent i has a set of estimators $\hat{\gamma}_j, j \in \mathcal{N}_i^{in}$ for the true state of each in-neighbour virtual target γ_j . In addition, each agent i has an estimator for its own state $\hat{\gamma}_i$ which is reset whenever vehicle i broadcasts its true state γ_i . The other estimators are reset whenever agent i receives the true state from its in-neighbours $j \in \mathcal{N}_i^{in}$. In their research paper, the authors propose a time-dependent triggering/broadcasting condition.

Proposition V.1. Consider the distributed control law given by

$$v_i^{coord} := -k_\varepsilon \sum_{j \in \mathcal{N}_i^{in}} a_{ij} (\gamma_i - \hat{\gamma}_j), \quad (60)$$

where k_ε is still a sufficiently large positive constant and $\hat{\gamma}_j$ is vehicle's i estimate of vehicle's j real state/path parameter. Consider also that the bank of estimators that each vehicle i is running follows the dynamics equation

$$\dot{\hat{\gamma}}_i := \mathbf{v}_L(\hat{\gamma}_i). \quad (61)$$

Based on the broadcast/triggering condition defined previously, at any time instant t , under negligible transmission delays, the vehicle's j self-state estimate $\hat{\gamma}_j$ is equal to vehicle's i estimate of $\hat{\gamma}_j$, which allows us to express the new distributed control law and estimator dynamics using vectorial notation as

$$\dot{\hat{\gamma}} := \mathbf{v}_L(\hat{\gamma}), \quad (62)$$

where, $\hat{\gamma} = [\hat{\gamma}_1, \dots, \hat{\gamma}_N]^T$ is the vector with the self-estimate of the state. It is possible to define \mathbf{v}^{coord} according to:

$$\begin{aligned} \mathbf{v}^{coord} &:= -k_\varepsilon [D\gamma - \mathcal{A}\hat{\gamma}] \\ &= -k_\varepsilon (\varepsilon + \mathcal{A}\tilde{\gamma}). \end{aligned} \quad (63)$$

Consider as well a triggering function used to define when to broadcast the state of each vehicle, defined as

$$\begin{cases} \delta_i(t) := |\tilde{\gamma}_i(t)| - g_i(t) \\ \tilde{\gamma}_i(t) = \hat{\gamma}_i(t) - \gamma_i(t) \end{cases}, \quad (64)$$

where $\tilde{\gamma}_i(t)$ is the local estimation error of agent i and $g_i(t)$ is a threshold function that is time dependent, such that if the estimation error exceeds this threshold, i.e. $\delta_i(t) \geq 0$, vehicle i broadcasts its state to the out-neighbours $\mathcal{N}_i^{\text{out}}$ and resets its local estimator. Furthermore, consider $g_i(t)$ to belong to a class of non-negative functions, given by

$$g_i(t) = c_i + b_i e^{-\alpha_i t}, \quad (65)$$

with c_i , b_i and α_i positive constants and $\mathbf{g}(t) = [g_1, \dots, g_N]^T$ the collection of functions for each individual vehicle. Then, the system is ISS with respect to the error vector $\boldsymbol{\varepsilon}$ and the inputs $\mathbf{v}_L(\boldsymbol{\gamma})$ and $\mathbf{h}(t)$, under the assumptions V.1 and V.2.

Proof. Consider the time derivative:

$$\begin{aligned} \dot{\boldsymbol{\varepsilon}} &= L(\mathbf{v}_L(\boldsymbol{\gamma}) - k_\varepsilon(\boldsymbol{\varepsilon} + \mathcal{A}\tilde{\boldsymbol{\gamma}})) \\ &= v_L \mathcal{L} \mathbf{1} + L \tilde{\mathbf{v}}_L - k_\varepsilon L(\boldsymbol{\varepsilon} + \mathcal{A}\tilde{\boldsymbol{\gamma}}) \\ &= -k_\varepsilon L(\boldsymbol{\varepsilon} + \mathbf{d}) \text{ with } \mathbf{d} = \frac{\tilde{\mathbf{v}}_L}{k_\varepsilon} + \mathcal{A}\tilde{\boldsymbol{\gamma}}, \end{aligned} \quad (66)$$

with $\mathbf{v}_L(\boldsymbol{\gamma}) = v_L \mathbf{1} + \tilde{\mathbf{v}}_L$. Consider now, the Jordan form $L = V^{-1} \Lambda V$. Moreover, consider the change of variables

$$\tilde{\boldsymbol{\varepsilon}} = -k_\varepsilon \Lambda(\boldsymbol{\varepsilon} + \bar{\mathbf{d}}), \text{ with } \bar{\mathbf{d}} = V^{-1} \mathbf{d}. \quad (67)$$

which is decomposable in

$$\begin{bmatrix} \dot{\tilde{\boldsymbol{\varepsilon}}}_1 \\ \dot{\tilde{\boldsymbol{\varepsilon}}}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -k_\varepsilon \Lambda_2(\tilde{\boldsymbol{\varepsilon}}_2 + \bar{\mathbf{d}}_2) \end{bmatrix}. \quad (68)$$

Consider now the candidate Lyapunov function

$$V_{\tilde{\boldsymbol{\varepsilon}}_2} = \frac{1}{2} \tilde{\boldsymbol{\varepsilon}}_2^T \tilde{\boldsymbol{\varepsilon}}_2, \quad (69)$$

and its time derivative given by

$$\dot{V}_{\tilde{\boldsymbol{\varepsilon}}_2} = -(1 - \theta) k_\varepsilon \tilde{\boldsymbol{\varepsilon}}_2^T \Lambda_2 \tilde{\boldsymbol{\varepsilon}}_2 - \theta k_\varepsilon \tilde{\boldsymbol{\varepsilon}}_2^T \Lambda_2 \bar{\mathbf{d}}_2 - k_\varepsilon \tilde{\boldsymbol{\varepsilon}}_2^T \Lambda_2 \bar{\mathbf{d}}_2 \quad (70)$$

where $0 < \theta < 1$. The term

$$-\theta k_\varepsilon \tilde{\boldsymbol{\varepsilon}}_2^T \Lambda_2 \bar{\mathbf{d}}_2 - k_\varepsilon \tilde{\boldsymbol{\varepsilon}}_2^T \Lambda_2 \bar{\mathbf{d}}_2 \quad (71)$$

will be ≤ 0 if

$$\|\tilde{\boldsymbol{\varepsilon}}_2\| \geq \frac{1}{\theta} \|\bar{\mathbf{d}}_2\| \quad (72)$$

and

$$\dot{V}_{\tilde{\boldsymbol{\varepsilon}}_2} \leq -(1 - \theta) k_\varepsilon \tilde{\boldsymbol{\varepsilon}}_2^T \Lambda_2 \tilde{\boldsymbol{\varepsilon}}_2. \quad (73)$$

The norm $\|\mathbf{d}\|$ can be made arbitrarily small and so does $\|\bar{\mathbf{d}}_2\|$ by proper choice of k_ε and triggering condition. From the application of 4.19 in H. Khalil [14], the system is ISS with respect to the error vector $\boldsymbol{\varepsilon}$ and inputs $\tilde{\boldsymbol{\gamma}}$ and $\tilde{\mathbf{v}}_L$. \square

VI. PATH PLANNING

Problem VI.1. Consider an UAV (quadrotor) flying over a body of water at a pre-defined fixed altitude, equipped with a camera sensor pointing downwards with a fixed pitch angle relative to the vehicle's body reference frame $\{B\}$. Consider also that the vehicle is capable of detecting environmental boundaries in the 2-D image provided by the camera. Furthermore, consider that at the surface of the water, one or more ASV vehicles are required to follow the quadrotor according to a pre-defined vehicle formation.

As the quadrotor detects an environmental boundary in the 2-D image, generate a set of smooth and planar reference

paths for each individual vehicle (quadrotor and ASVs), such that they encircle the boundary according to the pre-defined formation.

To solve problem VI.1 an online path planning algorithm is proposed. Given a stream of pixels corresponding to a boundary to be followed, the proposed algorithm:

- 1) uses the data provided by the quadrotor navigation system to convert the pixels to a 2D point cloud;
- 2) does some pre-processing on the point cloud;
- 3) fits the data with open B-splines, by formulating an online optimization problem;
- 4) generates a path for each vehicle to follow based on the generated B-spline;
- 5) repeats the process once new data is available.

A few simplifying assumptions are made:

Assumption VI.1. The environmental boundary is located at the ocean's surface assumed to be a 2D plane at $Z = 0$ in the inertial frame of reference $\{U\}$.

Assumption VI.2. The quadrotor has a navigation system that can track the vehicle's pose with a "good enough" accuracy.

Assumption VI.3. The quadrotor has a limited vision of the environment, i.e., the camera sensor might not be able to capture the entire boundary, but rather sections of it, according to Figure 1.

Assumption VI.4. The detection of the pixels that encode the boundary in the image frame is a sub-system that we assume to be already developed and readily available.

A. Camera Model

In order to convert pixels in an image frame to a 2D point cloud expressed in the inertial frame $\{U\}$, we must first introduce the camera model adopted (Figure 3).

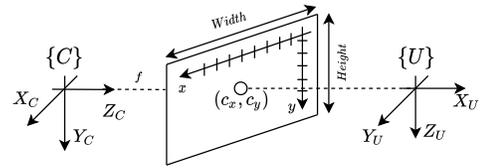


Fig. 3. Camera model and reference frames

A camera can be characterized by: i) a set of extrinsic parameters ${}^C_U[R|T]$, which model the conversion between coordinates expressed in the world/inertial reference frame $\{U\}$ and the camera reference frame $\{C\}$; ii) intrinsic parameters K which describe how a set of points in $\{C\}$ are represented in the image frame.

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} fsx & 0 & c_x \\ 0 & fsy & c_y \\ 0 & 0 & 1 \end{bmatrix}}_K \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{{}^C_U[R|T]} \begin{bmatrix} X_U \\ Y_U \\ Z_U \\ 1 \end{bmatrix}, \quad (74)$$

where (x, y) denote the coordinates in the image frame, f denotes the focal distance, (s_x, s_y) are scale factors in the X

and Y-axis respectively, (c_x, c_y) correspond to the offset of the focal point in the image plane and λ is a scale factor. Furthermore, we can aggregate the intrinsic and extrinsic parameters in a matrix Ω according to

$$\Omega = K \cdot \underset{U}{C} [R|T]. \quad (75)$$

Taking into consideration assumption VI.1 we can define the linear system:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{\lambda} \underbrace{\begin{bmatrix} \Omega_1 & \Omega_2 & \Omega_3 & \Omega_4 \\ \Omega_5 & \Omega_6 & \Omega_7 & \Omega_8 \\ \Omega_9 & \Omega_{10} & \Omega_{11} & \Omega_{12} \end{bmatrix}}_{\Omega} \begin{bmatrix} X_U \\ Y_U \\ 0 \\ 1 \end{bmatrix}. \quad (76)$$

Making use of assumption VI.2 we can assume that the linear system of equations is well defined and can be inverted such that for each pixel, X_U and Y_U are extracted from

$$\frac{1}{\lambda} \begin{bmatrix} X_U \\ Y_U \\ 1 \end{bmatrix} = \begin{bmatrix} \Omega_1 & \Omega_2 & \Omega_4 \\ \Omega_5 & \Omega_6 & \Omega_8 \\ \Omega_9 & \Omega_{10} & \Omega_{12} \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (77)$$

This methodology is far from perfect and small estimation errors in the vehicle attitude estimation can lead to errors of several meters in the generated point cloud. Synchronizing the orientation data provided by the positioning system and the camera images can also prove challenging when the quadrotor's orientation is changing very fast. In order to avoid data synchronization problems, a naive step is added to the process, such that if for a given planning iteration $\|\mathbf{v}_2\| \geq \omega_{max}$, the image data is discarded and no point cloud data is produced.

B. Pre-processing Point Cloud Data

In this section, the problem of developing an online path planning algorithm that given a set of points generates a desired path for the quadrotor (the leader vehicle) to follow is addressed. Start by considering the example in Figure 4 where the vision system of the quadrotor produces a point cloud, representing the boundary to be followed, at an arbitrary time-step t_k . In the point cloud, some points represent the environmental boundary in a region close to the vehicle (with outliers) - the region of interest. Other points represent regions of the boundary that were partially occluded and, therefore, seem disconnected from the main cluster of points. The goal is for the vehicle to follow a path (depicted in red) which fits only points in the region close to itself. This requires that at the pre-processing stage the cluster of points that are further away from the vehicle, as well as outliers, are ignored.

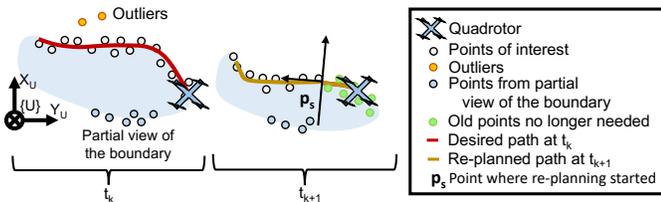


Fig. 4. Pre-processing stage

Consider that at time-step t_{k+1} the vision system produces a new point cloud in which some of the points overlap a region

where the quadrotor has already flown by. Since this section of the path should not be re-planned, those points should be discarded as well as other points that are "behind" \mathbf{p}_s (an arbitrary point further ahead of the vehicle).

1) **Remove unused points:** Consider $\mathbf{p}_s \in \mathbb{R}^2$ to be the point at which the path re-planning starts (to be defined later on), arbitrarily further ahead of the vehicle's position on the current path. In order to remove the points that are "behind" \mathbf{p}_s , consider that ψ_s is the tangent angle to the current path at \mathbf{p}_s . A coordinate transformation can be applied to the new points $\mathbf{X} := \{\mathbf{X}_m\}_{m=1}^M \in \mathbb{R}^2$, such that in a new reference frame, points that are behind \mathbf{p}_s (points that should be ignored) have a negative X-coordinate. This coordinate transformation is given by

$$\mathbf{X}_m^\circ = R(\psi_s) \cdot (\mathbf{X}_m - \mathbf{p}_s), \forall m = 1, \dots, M \quad (78)$$

where $\mathbf{X}_m^\circ = [\mathbf{X}_m^{\circ x}, \mathbf{X}_m^{\circ y}]^T$. Each point \mathbf{X}_m is discarded if $\mathbf{X}_m^{\circ x} < 0$. The points that belong to set \mathbf{X} and are not discarded, should be saved in a new set $\mathbf{X}^* := \{\mathbf{X}_j\}_{j=1}^J \in \mathbb{R}^2$ with $J \leq M$.

2) **Order a set of points and remove outliers:** In order to fit a set of points with a parametric curve, it is necessary to infer some natural ordering from the data. Unlike most path planning problems, we lack the knowledge of which points represent an end position goal, as well as in which order should the vehicle pass near each point.

To arrange the points in a consistent manner, the authors in [15] propose the construction of an Euclidean Minimum Spanning Tree (EMST) from a set of unordered points. To compute the EMST associated to the data we must first construct a graph from the set of points \mathbf{X}^* such that each vertex \mathcal{V} in the graph represents a point \mathbf{X}_i , and each edge \mathcal{E} , with its associated weight A_{ij} , represents the euclidean distance between each pair of points i and j . For computing a Minimum Spanning Tree (MST) one can resort to Kruskal's algorithm - a greedy algorithm with a computational complexity of $\mathcal{O}(|\mathcal{E}| \log |\mathcal{V}|)$ [16].

If we consider that each vertex is connected to each other, the construction of the graph itself will have a computational complexity of approximately $\mathcal{O}(|\mathcal{V}|^2)$ due to the necessity of having to compute the euclidean norm between each pair of points. This is not suitable for real time applications nor to use with Kruskal's algorithm as the resulting graph will be dense. In order to simplify this problem we can consider that each point is only related to its nearest set of points.

To find the nearest neighbours for each point, we can resort to a very popular unsupervised learning data structure proposed by Jon Bentley, the KDTree [17]. By defining a threshold distance N_j for the neighbours of each point, the computational cost of this nearest neighbour search for each individual point is on average $\mathcal{O}(2|\mathcal{V}| \log(|\mathcal{V}|))$ [18]. Repeating this operation for all J points, we construct a sparse graph where each point has a limited set of neighbours. From there we can use Kruskal's algorithm to generate the EMST.

To get rid of outliers and define a coarse path to follow, Breadth-First Search (BFS) can be applied to the points that

form the MST, starting from \mathbf{p}_s . The resulting ordered list of points that forms the path with the highest number of points should be saved in a new ordered set $\mathbf{X}^\dagger := \{\mathbf{X}_k\}_{k=1}^K \in \mathbb{R}^2$ with $K \leq J$.

C. Fitting data with a parametric curve

In order to have a suitable representation of a path that PF controllers can track, we are required to have parametric curves that are both smooth and at least \mathcal{C}^2 . For this work, we resort to uniform cubic B-Splines.

1) **Define the number of segments:** Consider now the ordered sequence of K points obtained via the application of the previously described steps. In order to fit the ordered sequence of points with a parametric curve we are required to attribute to each point $\mathbf{X}_k \in \mathbb{R}^2$ a corresponding γ_k in the target curve. This problem could be formulated as a nonlinear optimization problem (computationally demanding to solve for real-time applications). A good approximation proposed in [19] is to consider D_X to be the total distance between the points, given by

$$D_X := \sum_{k=2}^K \|\mathbf{X}_k - \mathbf{X}_{k-1}\|, \quad (79)$$

and the vector of parametric values $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_K]^T$ associated to the set of points being fitted, given by

$$\begin{cases} \gamma_1 = 0 \\ \gamma_k = \gamma_{k-1} + \frac{\|\mathbf{X}_k - \mathbf{X}_{k-1}\|}{D_X} \gamma_{max}, k = 2, \dots, K \end{cases}, \quad (80)$$

where γ_{max} is the maximum parameter value of the parametric curve. For cubic B-splines this number depends directly on the number of control points that the target curve will have. The number of control points also dictates how many spline sections are actually used for the fitting problem. Since uniform cubic B-splines must have at least 4 control points in order to define one segment, a dynamic way of defining the number of control points N_C used is by taking:

$$N_C := \max \left\{ \left\lfloor \frac{D_X}{\rho} \right\rfloor, 4 \right\}, \quad (81)$$

where $(1/\rho) > 0$ denotes a control points density (a tuning parameter) and $\lfloor D_X/\rho \rfloor$ denotes the rounding to the nearest integer value. For a smaller ρ , the higher the number of control points used.

Before fitting the points with a new B-spline with N_C segments, it is important to define which segments from the previously planned path should be discarded and which should be kept. For that it is now necessary to define the point \mathbf{p}_s at which the new planning starts.

2) **Fit the points with a B-spline:** Consider a 2-dimensional uniform cubic B-spline definition expressed in vectorial form, according to:

$$\mathbf{C}(\boldsymbol{\gamma}) = \mathbf{B}(\boldsymbol{\gamma}) \cdot \mathbf{P}, \quad (82)$$

where $\mathbf{B}(\boldsymbol{\gamma})$ denotes a matrix of basis functions evaluated at a given parameter $\boldsymbol{\gamma}$ and \mathbf{P} the vector of control points where the X and Y-coordinates are concatenated. Each spline section is only defined by four local control points - local support property.

A naive approach to the define the re-planning point \mathbf{p}_s (undefined until now) is to consider the quadrotor's current virtual target γ_{t_k} and define the starting point for re-planning according to

$$\mathbf{p}_s := \mathbf{C}(\lceil \gamma_{t_k} \rceil), \quad (83)$$

where $\lceil \gamma_{t_k} \rceil$ denotes the ceiling of γ_{t_k} (the vehicle's virtual target value at instant of re-planning, t_k). To remove sections of the curve to be re-planned corresponding to $\gamma > \lceil \gamma_{t_k} \rceil$, all the control points with indexes $i \geq \lceil \gamma_{t_k} \rceil + 3$ should be discarded.

To generate new curve sections, an optimization problem is solved such that the ordered sequence of points \mathbf{X}^\dagger is fitted with a new B-spline dictated by N_C control points. Making use of the local support property, it is known that \mathcal{C}^2 continuity between the old curve and the new one can be guaranteed as long as the first 3 new control points are equal to the 3 last control points in the previous curve. Consider the following optimization strategy:

$$\begin{aligned} \mathbf{P}^{new} = \underset{\mathbf{P}^{new}}{\operatorname{argmin}} \quad & \underbrace{\|B(\boldsymbol{\gamma})\mathbf{P}^{new} - \mathbf{X}\|^2}_{\text{goal}} + \underbrace{\lambda \mathbf{P}^{new T} R_1 \mathbf{P}^{new}}_{\text{regularizer}} \\ & + \underbrace{\beta \mathbf{P}^{new T} R_2 \mathbf{P}^{new}}_{\text{regularizer}} \end{aligned} \quad (84)$$

$$\text{subject to} \quad \begin{bmatrix} P_0^x \\ P_1^x \\ P_2^x \\ P_0^y \\ P_1^y \\ P_2^y \end{bmatrix}^{new} = \begin{bmatrix} P_{\lceil \gamma_{t_k} \rceil}^x \\ P_{\lceil \gamma_{t_k} \rceil + 1}^x \\ P_{\lceil \gamma_{t_k} \rceil + 2}^x \\ P_{\lceil \gamma_{t_k} \rceil}^y \\ P_{\lceil \gamma_{t_k} \rceil + 1}^y \\ P_{\lceil \gamma_{t_k} \rceil + 2}^y \end{bmatrix}$$

where $\mathbf{P}^{new} = [P_0^{x, new}, \dots, P_{N_C-1}^{x, new}, P_0^{y, new}, \dots, P_{N_C-1}^{y, new}]^T$ defines the new vector of control points (with X and Y-coordinates concatenated), and $\mathbf{P}^{old} = [P_0^x, P_1^x, \dots, P_{\lceil \gamma_{t_k} \rceil}^x, P_{\lceil \gamma_{t_k} \rceil + 1}^x, P_{\lceil \gamma_{t_k} \rceil + 2}^x, P_0^y, P_1^y, \dots, P_{\lceil \gamma_{t_k} \rceil}^y, P_{\lceil \gamma_{t_k} \rceil + 1}^y, P_{\lceil \gamma_{t_k} \rceil + 2}^y]^T$ denotes the vector of control points used to represent the initial curve, $\lambda, \beta \geq 0$ are regularization constants. Matrix R_1 results from the integral of the L_2^2 norm of the first derivative of the basis functions, and penalizes the total length of the curve. The matrix R_2 results from the integral of the L_2^2 norm of the second derivative of the basis function, and penalizes bends in the path.

Remark: For the first iteration the same algorithms are used with the following key differences:

- The position \mathbf{p}_s and angle ψ_s are given by the vehicle's position and orientation (and not by a virtual target);
- The linear constraint in (84) is now given by $B(0)\mathbf{P}^{new} = \mathbf{p}_s$, such that the initial path starts at the vehicle's position;

D. Multi Path Coordination

To generate individual paths for the ASVs, consider a formation vector denominated $\mathbf{d}_i \in \mathbb{R}^3$, different for each vehicle, with each distance defined with respect to a tangential reference frame $\{T\}$ to the virtual target's position in the path. Then, the desired position for ASV i can be given by

$$\mathbf{p}_{Fi}(\gamma_i) = \mathbf{p}_d(\gamma_i) + \mathbf{U}_T^T R(\gamma_i) \mathbf{d}_i, \quad (85)$$

where $\mathbf{U}_T^T R(\gamma_i)$ encodes a rotation from a tangential frame in the leader path to the inertial frame.

VII. RESULTS

A. CPF with ETC between Quadrotor and 2 Medusa ASVs (simulation)

For the first experiment a CPF mission was performed where the quadrotor was required to follow a lawn-mowing trajectory with two Medusa ASVs, at a desired speed of 0.5m/s, according to a triangle formation (Figure 5). In this experiment there was bi-directional communication between the pairs of vehicles: (quadrotor, Medusa 1) and (quadrotor, Medusa 2).

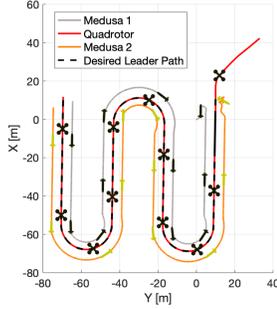


Fig. 5. XY-view

The aircraft was required to fly at an altitude of 30m and the formation vectors for the marine vehicles were given by $\mathbf{d}_1 = [-5, 5, 0]^T$ m and $\mathbf{d}_2 = [-5, -5, 0]^T$ m. From the results obtained in Figure 6, it is observable that the vehicles converge to their desired formation at around second 20. After that period of time, the position error converges to zero and the virtual target speeds converge to their desired value. As a consequence, the number of communication events between the vehicles drops as the bank of observers in each vehicle can more accurately track the state of the virtual target of their peers.

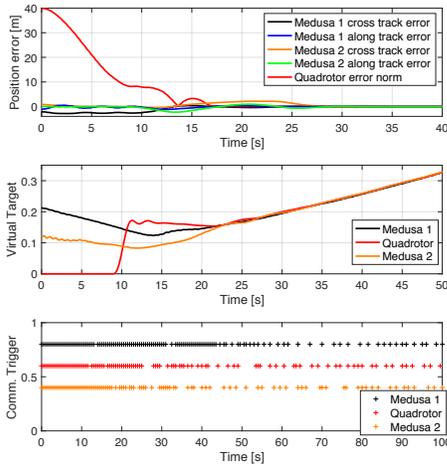


Fig. 6. CPF between quadrotor and 2 medusas (simulation)

B. Boundary Tracking with Quadrotor and a Medusa ASV (simulation)

For the next experiment, the quadrotor was required to start a lawn-mowing CPF mission with one Medusa ASV, and as soon as it detects the environmental boundary, start re-planning in real time the desired path to follow, at a pre-defined height of 30m with a desired constant speed of 0.5m/s (Figure 7). The quadrotor was equipped with a fixed camera with a pitch angle of -45° , pointing downwards. The quadrotor is required to

align itself with the tangent to the path to guarantee a constant overview of the boundary being followed.

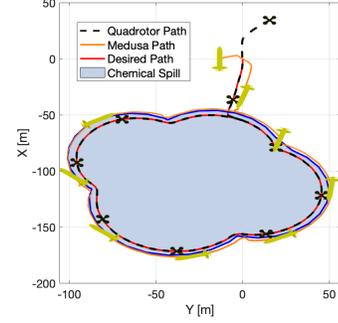


Fig. 7. 2D-view

It was desirable for the marine vehicle to always follow the aerial vehicle from behind and never in front of it, to guarantee that the path further ahead can be generated. For that matter, a formation of vector $\mathbf{d} = [-5, 5, 0]^T$ m was once again picked. There was bi-directional communication between both vehicles. In Figure 8 the evolution of the boundary distance to each vehicle is provided, along with the path following position tracking error.

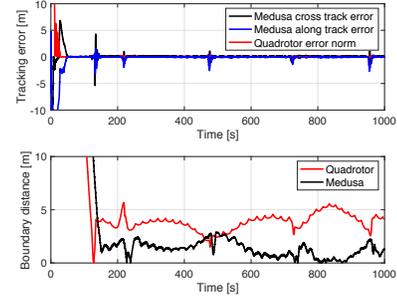


Fig. 8. Performance metrics

Remark: A demonstration video is available online [20].

C. CPF with ETC between 2 Medusa Vehicles (real trials)

For the real trial, it was requested that two Medusa vehicles performed a lawn-mowing mission cooperatively (Figure 9).

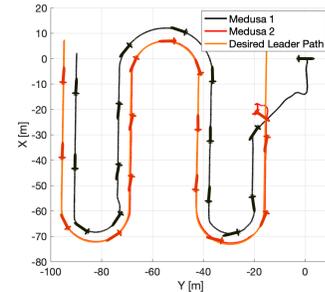


Fig. 9. Medusa cooperative path following (real trial)

The black vehicle (Medusa 1) was required to follow a formation dictated by $\mathbf{d} = [-5, -5, 0]$, with respect to the leader's path. From the results in Figure 10 it is observable that the virtual targets of each vehicle rapidly increase to a value of approximately 0.5 such that they get as close as possible to the original position of their respective vehicles. Some oscillations are also observed in the beginning on the trial which result from the virtual targets finding an agreement

between inter-vehicle alignment and intra-vehicle position error minimization.

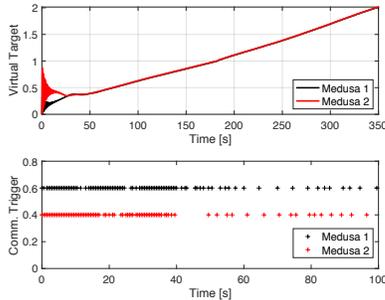


Fig. 10. Triggering condition

After approximately 50s, the vehicles align themselves into the required formation and as a consequence, the rate of information exchange decreases after this period of time.

VIII. CONCLUSION

This dissertation addressed the problem of tracking and following an environmental boundary caused by a chemical spill using a team of robots composed of an aerial quadrotor and marine vehicles.

In the vehicle modelling section, the notation and reference frames adopted for both the ASV and UAV were introduced. In the next section, the problem of vehicle inner-loop control was formulated and a set of linear control schemes were derived for both vehicles. Next, the PF problem was introduced, and a non-linear control law derived for the ASV, according to the proposal by P. Aguiar and F. Vanni. Inspired by this control law, a new one was derived for a quadrotor following the same methodology with some key differences due to the nature of the aircraft.

For the section that followed, the CPF problem was formulated and a proposal to solve the problem was presented, such that the synchronization controller was distributed and the same for all vehicles (aerial and marine). Borrowing from the works of A. Pascoal, N. Hung and F. Rego this controller was devised such that information exchange between vehicles would only be carried using even-triggered communications.

For the following chapter, a new real-time path planning algorithm was developed. This algorithm made use of the camera sensor onboard of the quadrotor to have a local view of the boundary and generate a point cloud expressed in the inertial frame. This data was then used to solve an optimization problem which generates a B-spline based path that grows dynamically as the vehicle moves along the boundary and acquires more data. This path is then shared among all the ASV vehicles in the network.

Finally, the proposed algorithms were implemented in four main toolbox by resorting to ROS, C++ and python. Moreover, a 3-D virtual scenario that resembles Doca dos Olivais was also generated, allowing for realistic simulations of the proposed algorithms. In the end it was also possible to test the CPF algorithm using two real Medusa vehicles.

A. Future Work

In this work some problems were left unsolved. Some notable work that could be addressed includes:

- Considering event-triggered communication for CPF under network changing topologies and communication delays;
- Make the height at which the quadrotor operates dynamic;
- Introducing obstacle avoidance into the path planning problem.

REFERENCES

- [1] Introduction to oceanography. [Online]. Available: <https://rwu.pressbooks.pub/webboceanography/chapter/1-1-overview-of-the-oceans/>
- [2] Life's origins by land or sea - debate gets hot. [Online]. Available: <https://www.scientificamerican.com/article/lifes-origins-by-land-or-sea-debate-gets-hot/>
- [3] Geoblueplanet. Ocean resources. [Online]. Available: <https://geoblueplanet.org/ocean-resources/>
- [4] D. Teixeira, "Sensor-based cooperative control of multiple autonomous marine vehicles," M.Sc. thesis, Instituto Superior Técnico, 11 2019.
- [5] T. Luukkonen, "Modelling and control of quadcopter," *Independent research project in applied mathematics, Espoo*, vol. 22, 8 2011.
- [6] A. Pascoal, I. Kaminer, and P. Oliveira, "Navigation system design using time-varying complementary filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 4, pp. 1099–1114, 2000.
- [7] G. Sanches, "Sensor-based formation control of autonomous marine robots," M.Sc. thesis, Instituto Superior Técnico, 10 2015.
- [8] A. P. Aguiar and J. P. Hespanha, "Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1362–1379, 2007.
- [9] F. Vanni, A. P. Aguiar, and A. M. Pascoal, "Cooperative path-following of underactuated autonomous marine vehicles with logic-based communication," *Proceedings Volumes*, vol. 41, no. 1, pp. 107–112, 2008, 2nd IFAC workshop on navigation, guidance and control of underwater vehicles.
- [10] A. P. Aguiar, R. Ghabcheloo, A. M. Pascoal, and C. Silvestre, "Coordinated Path-Following Control of Multiple Autonomous Underwater Vehicles," vol. All Days, 07 2007, iSOPE-I-07-006.
- [11] D. Cabecinhas, R. Cunha, and C. Silvestre, "A nonlinear quadrotor trajectory tracking controller with disturbance rejection," *Control Engineering Practice*, vol. 26, p. 1–10, 05 2014.
- [12] A. P. Aguiar and A. M. Pascoal, "Coordinated path-following control for nonlinear systems with logic-based communication," in *2007 46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 1473–1479.
- [13] N. T. Hung, F. C. Rego, and A. M. Pascoal, "Event-triggered communications for the synchronization of nonlinear multi agent systems on weight-balanced digraphs," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 2713–2718.
- [14] H. Khalil, *Nonlinear systems*, 3rd ed., ser. Always Learning. Pearson Education Limited, 2013.
- [15] Y. Liu, H. Yang, and W. Wang, "Reconstructing B-spline curves from point clouds—a tangential flow approach using least squares minimization," in *International Conference on Shape Modeling and Applications 2005 (SMI' 05)*. IEEE, 2005, pp. 4–12.
- [16] Hackerearth - minimum spanning tree. [Online]. Available: <https://www.hackerearth.com/practice/algorithms/graphs/minimum-spanning-tree/tutorial/>
- [17] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, p. 509–517, Sep. 1975.
- [18] Scikit learn - unsupervised learning, section 1.6.4.2. k-d tree. [Online]. Available: <https://scikit-learn.org/stable/modules/neighbors.html>
- [19] M. Liu, S. Huang, G. Dissanayake, and S. Kodagoda, "Towards a consistent SLAM algorithm using B-splines to represent environments," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2065–2070.
- [20] M. Jacinto. Thesis demo video. [Online]. Available: <https://youtu.be/ax8q3wf5MYM>