2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Integrated vector field and backstepping control for quadcopters

Arthur H. D. Nunes, Guilherme V. Raffo, and Luciano C. A. Pimenta

Abstract— **In this work, we present an Integrated Guidance and Controller (IGC) scheme to drive quadcopters in path-following tasks with obstacle avoidance and constant uncertainty rejection. This scheme is based on the combination of a time-varying artificial vector field and Backstepping with integral action control. The vector field switches between two behaviors: (i) path-following; and (ii) obstacle circumnavigation to allow collision avoidance. This vector field is then integrated into a non-linear controller designed via Backstepping with Integral Action to deal with the quadcopter vehicle dynamics and reject constant uncertainties. The considered vehicle model is based on quaternion algebra. The control inputs are considered to be the total thrust and torques. Stability is proved by using Lyapunov's Theory and Matrosov's Theorem.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) are of great interest in the fields of robotics and control. These vehicles can accomplish several tasks such as surveillance [1], monitoring [2], target tracking [3], search and rescue [4], package delivery [5], autonomous race [6], among many others.

Several controllers for those vehicles have already been proposed in the literature [7], [8], [9], [10], [11], [12], [13]. In this work, we propose a novel quadcopter controller to allow path-following and obstacle avoidance in the presence of constant uncertainties, such as external disturbances or inaccurate model parameters. Depending on the task, path-following strategies overcome trajectory-tracking ones, because they have smoother convergence to the path and less demand for control efforts [13]. Our method is based on the integration of an artificial vector field guidance strategy [14], [15] and a Backstepping control with Integral Action [16].

Vector field navigation strategies have been discussed in many works [17], [18], [19], [20], [21], [22]. In [17], the authors have proposed a vector field for static curve following in n dimensions represented by a parametric equation. The authors have extended the n-dimensional space to $n+1$ dimensions and applied the method from [18]. Thus, the resulting vector field is singularity-free. In [19], [20], obstacle avoidance has been considered. However, time-varying paths and generic-shaped obstacles have not been considered in these works.

A constructive vector field for time-varying path-following in n dimensions has been previously proposed by our group in [14]. In this recent method, the field is obtained constructively with the use of the Euclidean distance function to the curve. It uses a parametric representation of the target curve, an advantage over the intersection of zero-level sets required in [18]. Several characteristics and advantages of the new approach have been presented in [14], including formal proofs. Nevertheless, the presence of obstacles has not been considered in that work. Then, in [15], we have extended this field to allow obstacle avoidance in the field formulation by considering obstacle circumnavigation when an imminent collision is detected. The circumnavigation is achieved by using the obstacle closest point to the vehicle.

Guidance methods are usually used as an outer control layer, and then, it is necessary to combine them with low-level controllers to be able to drive real aerial vehicles such as quadcopters, usually in a cascade scheme. In a previous work [23], we have proposed a cascade path-following control for quadcopters, in which the vector field from [18] is used in a guidance layer. The controller has not considered collision avoidance or uncertainty rejection. The quadcopter model has been considered to receive the total thrust and angular velocities as inputs, assuming an acro-rate control mode.

Now, in this work, we further extend our previous developments by incorporating the new collision-free vector field in a Backstepping control. As opposed to the widely used cascade strategies where the guidance is a separate layer from the rest of the controller (SGC), we propose an Integrated Guidance and Controller (IGC) scheme.

Backstepping is a well-known non-linear control technique [24] which has advantages over linear approaches, such as PID and LQR [13], [25], [26], [27], and has been successfully applied to quadcopters [28], [29], [30], [31], [32]. Unlike previous works, here we use Backstepping as a path controller and also allow obstacle avoidance. Both are accomplished by integrating the guidance in the controller. Furthermore, we use integral action to allow constant uncertainty rejection [16].

In summary, the main novelties and contributions of this work are: (i) integration between the field and the low-level controller, giving rise to a robust IGC strategy that is able to reject constant uncertainties and to avoid collisions; (ii) stability proof using Lyapunov's Theory and Matrosov's Theorem; (iii) consideration of torques as control inputs instead of angular velocities [23]; (iv) application of a novel collision-free vector field strategy from [14], [15]; and (v) algebraic formulation based on quaternions, [33], [34], [35].

The authors are with the Graduate Program in Electrical Engineering (PPGEE) of the Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, MG, 31270-901, Brazil (emails: {arthurhdn,raffo,lucpim}@ufmg.br).

This work was in part supported by the project INCT (National Institute of Science and Technology) under the grant CNPq (Brazilian National Research Council) 465755/2014-3 and FAPESP (São Paulo Research Foundation) 2014/50851-0. This work was also supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) (Finance Codes 001 and 88887.136349/2017-00), CNPq (grant numbers 315258/2020-9, 315695/2020-0, and 407063/2021-8), and FAPEMIG, under grant APQ-03090-17.

II. System Modeling

In this section, we aim to describe the system and propose a model for it using quaternion algebra. Before presenting the system and its model, we first show a quick algebra review.

A. Quaternion algebra review

We first start defining the imaginary identities $\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = \hat{i}\hat{j}\hat{k} = -1$. Then, let $\mathbb{H}$ be the quaternion set given by

$$\mathbb{H} = \{\boldsymbol{h} = h_0 + \hat{i}h_1 + \hat{j}h_2 + \hat{k}h_3 : h_0, h_1, h_2, h_3 \in \mathbb{R}\}. \quad (1)$$

The real part of a quaternion is defined as $\mathrm{Re}(\boldsymbol{h}) = h_0$; the imaginary as $\mathrm{Im}(\boldsymbol{h}) = \hat{i}h_1 + \hat{j}h_2 + \hat{k}h_3$; and the conjugate as $\boldsymbol{h}^* = \mathrm{Re}(\boldsymbol{h}) - \mathrm{Im}(\boldsymbol{h})$.

The set of quaternions with null real parts is called pure quaternions set. Let $\mathbb{H}_p = \{\boldsymbol{h} \in \mathbb{H} : \mathrm{Re}(\boldsymbol{h}) = 0\}$ be this set. It is similar to the $\mathbb{R}^3$ set. Therefore, we can define the cross and dot products between two pure quaternions as $\boldsymbol{u} \times \boldsymbol{v} = \frac{\boldsymbol{uv} - \boldsymbol{vu}}{2}$; $\boldsymbol{u} \cdot \boldsymbol{v} = -\frac{\boldsymbol{uv} + \boldsymbol{vu}}{2}$, respectively. See [33], [34], [35] for more details.

B. Quadcopter system

The controlled system in this work consists of a helicopter with four rotors symmetrically arranged in a plane. Its movement is a result of the four individual rotors' velocities. Each rotor generates a force $f_i, i \in \{1, 2, 3, 4\}$. In steady state, each rotor's force can be approximated by $f_i = b\Omega_i^2$, where b is the thrust coefficient, and Ω_i is each rotor's angular velocity around its axis [36].

Fig. 1 shows a schematic for the vehicle, where its center is a blue cube with the center of mass in the geometric center. A reference frame rigidly attached to the body is plotted in RGB. The forces f_i are defined as pointing in direction $\hat{\boldsymbol{k}}_B$. It is considered that rotors 1 and 3 spin in $\hat{\boldsymbol{k}}_B$'s negative direction, while rotors 2 and 4 spin in $\hat{\boldsymbol{k}}_B$'s positive direction.

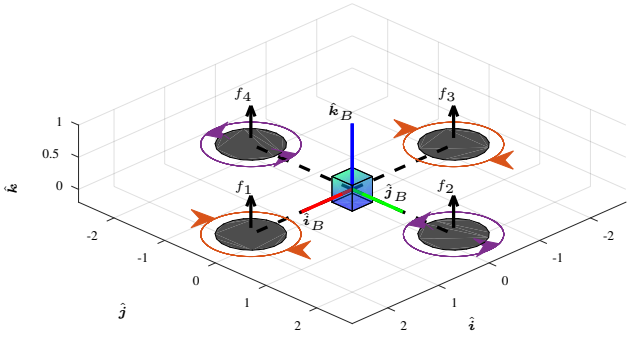

Fig. 1. Schematic representation of the quadcopter vehicle.

The control inputs are considered to be the total thrust $u_T \in \mathbb{R}$ and the torques $\boldsymbol{u}_\tau \in \mathbb{H}_p$, which are given by

$$\begin{aligned} u_T &= \textstyle\sum_{i=1}^{4} f_i \in \mathbb{R}, \\ \boldsymbol{u}_\tau &= \hat{\boldsymbol{i}}_B \ell(f_2 - f_4) + \hat{\boldsymbol{j}}_B \ell(f_3 - f_1) \\ &+ \hat{\boldsymbol{k}}_B \tfrac{k_r}{b}(f_1 - f_2 + f_3 - f_4) \in \mathbb{H}_p, \end{aligned} \quad (2)$$

where k_r is an opposite force to the torsion effort constant, and ℓ is the distance between the rotor axis and the center of mass.

C. Modeling

In order to define the system states and dynamics, let $\boldsymbol{p} = \hat{\boldsymbol{i}}p_x + \hat{\boldsymbol{j}}p_y + \hat{\boldsymbol{k}}p_z \in \mathbb{H}_p$ denote the three-dimensional position. Let $\boldsymbol{o} = \cos(\frac{\phi}{2}) + \boldsymbol{n}\sin(\frac{\phi}{2}) \in \mathbb{H}$, $\|\boldsymbol{o}\| = 1$ denote the three-dimension orientation in terms of a $\phi \in \mathbb{S}$ rotation around an axis $\boldsymbol{n} \in \mathbb{H}_p$, $\|\boldsymbol{n}\| = 1$. Also, let $\boldsymbol{v} = \hat{\boldsymbol{i}}v_x + \hat{\boldsymbol{j}}v_y + \hat{\boldsymbol{k}}v_z \in \mathbb{H}_p$ and $\boldsymbol{\omega} = \hat{\boldsymbol{i}}\omega_x + \hat{\boldsymbol{j}}\omega_y + \hat{\boldsymbol{k}}\omega_z \in \mathbb{H}_p$ denote, respectively, the linear and angular velocities.

Given the states $\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{v}$, and $\boldsymbol{\omega}$, a dynamic model for the vehicle is given by

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \quad (3)$$

$$\dot{\boldsymbol{v}} = -g\hat{\boldsymbol{k}} + m^{-1}u_T\boldsymbol{o}\hat{\boldsymbol{k}}\boldsymbol{o}^* + m^{-1}\boldsymbol{A}_v, \quad (4)$$

$$\dot{\boldsymbol{o}} = \frac{1}{2}\boldsymbol{\omega}\boldsymbol{o}, \quad (5)$$

$$\dot{\boldsymbol{\omega}} = -J^{-1}(\boldsymbol{\omega} \times J\boldsymbol{\omega}) + J^{-1}\boldsymbol{u}_\tau + J^{-1}\boldsymbol{A}_\omega, \quad (6)$$

where $m \in \mathbb{R}$ is the vehicle's mass, $J \in \mathbb{R}^{3\times3}$ the vehicle's inertia tensor, and $\boldsymbol{A}_v, \boldsymbol{A}_\omega \in \mathbb{H}_p$ are additive uncertainties.

The considered inputs and model are similar to other works such as [28], [37], [36], [38]. The main difference here is the usage of quaternion algebra [33], [34], [35].

III. Problem Statement

The problem addressed in this work is defined by the following: considering the vehicle dynamic model (3)-(6), design a non-linear controller that drives the vehicle with constant speed $v_r \in \mathbb{R}$ to converge to and follow a given time-varying curve $\boldsymbol{\mathcal{C}}(t) \subset \mathbb{H}_p$ with parametric representation $\boldsymbol{r}(s,t) : \mathbb{R} \times \mathbb{R}_+ \to \mathbb{H}_p$, under Assumptions 1 and 2. The designed controller must: (i) allow obstacle avoidance; and (ii) be robust to constant uncertainties.

Assumption 1: The target time-varying curve $\boldsymbol{\mathcal{C}}(t)$ is either: (i) an open curve, *i.e.* an unbounded curve homeomorphic to a straight line; or (ii) a closed curve, *i.e.* a bounded curve homeomorphic to a circle.

Assumption 2: The parametric representation $\boldsymbol{r}(s,t)$ for the curve $\boldsymbol{\mathcal{C}}(t)$ has both the following properties: (i) it is twice differentiable in s for all t; and (ii) it is differentiable in t for all s.

If obstacles are sensed between the system and the curve, the controller must prioritize the avoidance over the curve following, i.e. the system is allowed to move away from the curve to accomplish the obstacle deviation and then return to it once the avoidance is finished.

A secondary control objective can be considered to control the yaw angle of the vehicle as it will be a free parameter as shown next in the methodology.

IV. Proposed Solution

We propose a backstepping controller with integral action to solve the stated problem. The integral action will allow the constant uncertainty rejection. The first step of the backstepping will consist of a vector field [15], as reviewed next, to accomplish the path-following with obstacle avoidance task.

A. Vector Field Review

In this part, we review the vector field from [14], [15] considering a three-dimensional space and using quaternion algebra.

The field is designed to control the first-order integrator model (3), where the control input is considered as $\boldsymbol{v}$.

1) Composite Field: The goal is to find $\boldsymbol{\mathcal{F}}(\boldsymbol{p},t) : \mathbb{H}_p \times \mathbb{R}_+ \to \mathbb{H}_p$, such that if $\dot{\boldsymbol{p}} = \boldsymbol{v} = \boldsymbol{\mathcal{F}}(\boldsymbol{p},t)$ the system (3) follows the target path $\boldsymbol{\mathcal{C}}(t)$ with constant speed v_r, while also avoiding obstacles.

To accomplish both path-following and obstacle avoidance, $\boldsymbol{\mathcal{F}}(\boldsymbol{p},t)$ will switch between: (i) a field that only follows the target curve, let us call it $\boldsymbol{\Phi}(\boldsymbol{p},t)$; and (ii) another field that only circumnavigates the closest obstacle, $\boldsymbol{\Psi}(\boldsymbol{p},t)$, at a fixed predefined safety distance λ. Both fields $\boldsymbol{\Phi}(\boldsymbol{p},t)$ and $\boldsymbol{\Psi}(\boldsymbol{p},t)$ will be reviewed soon.

The idea is to make $\boldsymbol{\mathcal{F}}(\boldsymbol{p},t) = \boldsymbol{\Phi}(\boldsymbol{p},t)$ when it is safe and make $\boldsymbol{\mathcal{F}}(\boldsymbol{p},t) = \boldsymbol{\Psi}(\boldsymbol{p},t)$ when an imminent collision is detected. The latter effectively means two conditions: (i) the curve field direction points to a direction in which the obstacle distance decreases; and (ii) the distance between the system and the closest obstacle, let us call it $\boldsymbol{D}_o(\boldsymbol{p},t)$, has norm smaller than a predefined threshold $D_{in} > \lambda$.

To avoid abrupt field transitions, an intermediate state is also considered to accomplish a smooth switching. Let $D_{in0} > D_{in}$ be another threshold value. Then, this intermediate state occurs when $D_{in} < \|\boldsymbol{D}_o(\boldsymbol{p},t)\| < D_{in0}$. In this state, $\boldsymbol{\mathcal{F}}(\boldsymbol{p},t)$ is a combination between both fields weighted by

$$\Theta(\boldsymbol{p},t) = \frac{\|\boldsymbol{D}_o(\boldsymbol{p},t)\| - D_{in}}{D_{in0} - D_{in}} \in [0,1]. \tag{7}$$

Then, $\boldsymbol{\mathcal{F}}(\boldsymbol{p},t)$ is computed according to

$$\boldsymbol{\mathcal{F}} = \begin{cases} \boldsymbol{\Psi}, & \boldsymbol{D}_o \cdot \boldsymbol{\Phi} < 0,\ \|\boldsymbol{D}_o\| < D_{in}, \\ \boldsymbol{\Phi}, & \boldsymbol{D}_o \cdot \boldsymbol{\Phi} \geq 0 \text{ or } \|\boldsymbol{D}_o\| > D_{in0}, \\ v_r \frac{\Theta\boldsymbol{\Phi} + (1-\Theta)\boldsymbol{\Psi}}{\|\Theta\boldsymbol{\Phi} + (1-\Theta)\boldsymbol{\Psi}\|}, & \text{otherwise.} \end{cases} \tag{8}$$

2) Curve Field: The curve field $\boldsymbol{\Phi}(\boldsymbol{p},t)$ follows the procedure from [14]. First, consider the curve parametric representation $\boldsymbol{r}(s,t)$ and then the definitions of the curve's closest point to the system

$$s_*(\boldsymbol{p},t) = \arg\min_s \|\boldsymbol{p} - \boldsymbol{r}(s,t)\|, \tag{9}$$

$$\boldsymbol{r}_*(\boldsymbol{p},t) = \boldsymbol{r}(s_*(\boldsymbol{p},t),t). \tag{10}$$

Next, the distance between the curve and the system, $\boldsymbol{D}(\boldsymbol{p},t)$, and the tangent component, $\boldsymbol{T}(\boldsymbol{p},t)$, of the curve at s_* are defined as

$$\boldsymbol{D}(\boldsymbol{p},t) = \boldsymbol{p} - \boldsymbol{r}_*(\boldsymbol{p},t), \tag{11}$$

$$\boldsymbol{T}(\boldsymbol{p},t) = \left.\frac{\partial \boldsymbol{r}(s,t)}{\partial s}\right|_{s=s_*}. \tag{12}$$

Those two components are weighted by scalar gains $G(\|\boldsymbol{D}(\boldsymbol{p},t)\|) = \frac{2}{\pi}\tan^{-1}(k_G\|\boldsymbol{D}(\boldsymbol{p},t)\|)$, $k_G > 0$, and $H(\|\boldsymbol{D}(\boldsymbol{p},t)\|) = \pm\sqrt{1 - G(\|\boldsymbol{D}(\boldsymbol{p},t)\|)^2}$.

Then, we compute a static unit field, $\boldsymbol{\Phi}_S$, and a feed-forward component, $\boldsymbol{\Phi}_T$, as in

$$\begin{aligned} \boldsymbol{\Phi}_S(\boldsymbol{p},t) &= -G(\|\boldsymbol{D}(\boldsymbol{p},t)\|)\frac{\boldsymbol{D}(\boldsymbol{p},t)}{\|\boldsymbol{D}(\boldsymbol{p},t)\|} \\ &+ H(\|\boldsymbol{D}(\boldsymbol{p},t)\|)\frac{\boldsymbol{T}(\boldsymbol{p},t)}{\|\boldsymbol{T}(\boldsymbol{p},t)\|}, \\ \boldsymbol{\Phi}_T(\boldsymbol{p},t) &= -\Pi_T(\boldsymbol{p},t)\frac{\partial \boldsymbol{D}}{\partial t}, \end{aligned} \tag{13}$$

where Π_T is the null space projector of $\frac{\boldsymbol{T}(\boldsymbol{p},t)}{\|\boldsymbol{T}(\boldsymbol{p},t)\|}$.

To weight the sum of the components defined above, another scalar gain $\eta(\boldsymbol{p},t) \in [0, v_r]$ is computed to ensure $\|\boldsymbol{\Phi}(\boldsymbol{p},t)\| = v_r$. This gain is given by

$$\eta = -\boldsymbol{\Phi}_S \cdot \boldsymbol{\Phi}_T + \sqrt{(\boldsymbol{\Phi}_S \cdot \boldsymbol{\Phi}_T)^2 + v_r^2 - \|\boldsymbol{\Phi}_T\|^2}, \tag{14}$$

where the dependencies $(\boldsymbol{p},t)$ were omitted.

Finally, the field $\boldsymbol{\Phi}(\boldsymbol{p},t)$ is given by

$$\boldsymbol{\Phi}(\boldsymbol{p},t) = \eta(\boldsymbol{p},t)\boldsymbol{\Phi}_S(\boldsymbol{p},t) + \boldsymbol{\Phi}_T(\boldsymbol{p},t). \tag{15}$$

3) Obstacle Field: The field $\boldsymbol{\Psi}(\boldsymbol{p},t)$ from [15] follows a similar procedure. First, consider the obstacle set $\boldsymbol{\mathcal{O}}(t)$. In this field, the obstacle's closest point will be used similar to the curve's closest point in the curve field. Let $\boldsymbol{p}_{o*} \in \boldsymbol{\mathcal{O}}(t)$ be this closest obstacle point, which is given by

$$\boldsymbol{p}_{o*}(\boldsymbol{p},t) = \arg\min_{\boldsymbol{p}_o \in \boldsymbol{\mathcal{O}}(t)} \|\boldsymbol{p} - \boldsymbol{p}_o\|. \tag{16}$$

This field's goal is to make the system circumnavigate the obstacles at a fixed distance $\lambda > 0$. Then, the distances are defined as

$$\boldsymbol{D}_o(\boldsymbol{p},t) = \boldsymbol{p} - \boldsymbol{p}_{o*}(\boldsymbol{p},t), \tag{17}$$

$$\boldsymbol{D}_\lambda(\boldsymbol{p},t) = \boldsymbol{D}_o(\boldsymbol{p},t) - \lambda\frac{\boldsymbol{D}_o(\boldsymbol{p},t)}{\|\boldsymbol{D}_o(\boldsymbol{p},t)\|}. \tag{18}$$

As described in [15], for the tangent component $\boldsymbol{T}_\lambda(\boldsymbol{p},t)$, there are infinite possibilities to contour the obstacle since we are in a three-dimensional space. Thus, $\boldsymbol{T}_\lambda(\boldsymbol{p},t)$ is chosen as the curve field removing its component on direction $\boldsymbol{D}_\lambda(\boldsymbol{p},t)$ as in

$$\boldsymbol{T}_\lambda(\boldsymbol{p},t) = \Pi_{D_\lambda}(\boldsymbol{p},t)\boldsymbol{\Phi}(\boldsymbol{p},t). \tag{19}$$

This will cause the vehicle to contour the obstacle in the closest direction to the curve.

Next, let $\boldsymbol{\Psi}_S(\boldsymbol{p},t)$ be the unit static component and $\boldsymbol{\Psi}_T(\boldsymbol{p},t)$ be the feed-forward component given by

$$\begin{aligned} \boldsymbol{\Psi}_S(\boldsymbol{p},t) &= -G_o(\|\boldsymbol{D}_\lambda(\boldsymbol{p},t)\|)\frac{\boldsymbol{D}_\lambda(\boldsymbol{p},t)}{\|\boldsymbol{D}_\lambda(\boldsymbol{p},t)\|} \\ &+ H_o(\|\boldsymbol{D}_\lambda(\boldsymbol{p},t)\|)\frac{\boldsymbol{T}_\lambda(\boldsymbol{p},t)}{\|\boldsymbol{T}_\lambda(\boldsymbol{p},t)\|}, \\ \boldsymbol{\Psi}_T(\boldsymbol{p},t) &= -\Pi_{T_\lambda}\frac{\partial \boldsymbol{D}_\lambda}{\partial t}. \end{aligned} \tag{20}$$

Finally the field $\boldsymbol{\Psi}(\boldsymbol{p},t)$ is

$$\boldsymbol{\Psi}(\boldsymbol{p},t) = \eta_o(\boldsymbol{p},t)\boldsymbol{\Psi}_S(\boldsymbol{p},t) + \boldsymbol{\Psi}_T(\boldsymbol{p},t). \tag{21}$$

For more details see [14], [15].

B. Backstepping with Integral Action

In this part, we describe the design of the proposed quadcopter controller using backstepping with integral action in four steps.

1) Step 1: In the first step, we consider the first-order integrator model (3) and design a virtual control law for the velocity. The field $\boldsymbol{\mathcal{F}}(\boldsymbol{p},t)$ (8) will be then this control law.

Let $\boldsymbol{z}_p \equiv \boldsymbol{D}(\boldsymbol{p},t) = \boldsymbol{p} - \boldsymbol{r}_*$ be the position error for the primary goal of following the target curve.

We can assure that $\boldsymbol{z}_p = 0$ will be a uniform global asymptotic stable (UGAS) equilibrium point only when no imminent collision is detected. Otherwise, the system can be forced to increase $\boldsymbol{z}_p$ to avoid collisions. For the sake of simplicity, consider that no obstacles are in an imminent collision and let $V_p : \mathbb{H}_p \to \mathbb{R}$ be a candidate Lyapunov function, given by $V_p(\boldsymbol{z}_p) = \frac{1}{2}\boldsymbol{z}_p \cdot \boldsymbol{z}_p$, from which, using (3) its time derivative yields

$$\dot{V}_p(\boldsymbol{z}_p) = \boldsymbol{z}_p \cdot (\boldsymbol{\mathcal{F}} - \dot{\boldsymbol{r}}_*). \tag{22}$$

Applying (8) and the results from [14], the asymptotic stability is ensured by

$$\dot{V}_p(\boldsymbol{z}_p) = -\eta G\|\boldsymbol{z}_p\| < 0,\ \forall \boldsymbol{z}_p \neq 0. \tag{23}$$

2) Step 2: In the second step, let $\boldsymbol{z}_v = \boldsymbol{v} - \boldsymbol{\mathcal{F}}(\boldsymbol{p},t)$ be the velocity error and let $\boldsymbol{z}_{vI} = \int_0^t \boldsymbol{z}_v dt$ be its integral. Then, consider the system with (8)

$$\dot{\boldsymbol{z}}_p = \boldsymbol{\mathcal{F}} - \dot{\boldsymbol{r}}_* + \boldsymbol{z}_v, \tag{24}$$

$$\dot{\boldsymbol{z}}_{vI} = \boldsymbol{z}_v, \tag{25}$$

$$\dot{\boldsymbol{z}}_v = -g\hat{\boldsymbol{k}} + \frac{u_T}{m}\hat{\boldsymbol{k}}_B - \dot{\boldsymbol{\mathcal{F}}} + m^{-1}\boldsymbol{A}_v. \tag{26}$$

The virtual control law in this step will be considered as $\boldsymbol{a} \equiv \frac{u_T}{m}\hat{\boldsymbol{k}}_B$. To find this control law, let $V_{p,v} : \mathbb{H}_p^3 \to \mathbb{R}$ be a control Lyapunov function given by $V_{p,v}(\boldsymbol{z}_p, \boldsymbol{z}_{vI}, \boldsymbol{z}_v) = k_p V_p + \frac{1}{2}k_{vI}\boldsymbol{z}_{vI} \cdot \boldsymbol{z}_{vI} + \frac{1}{2}\boldsymbol{z}_v \cdot \boldsymbol{z}_v$, with $k_p > 0, k_{vI} > 0$. Its time derivative yields

$$\begin{aligned} \dot{V}_{p,v} &= k_p \boldsymbol{z}_p \cdot (\boldsymbol{\mathcal{F}} + \boldsymbol{z}_v - \dot{\boldsymbol{r}}_*) + k_{vI}\boldsymbol{z}_{vI} \cdot \boldsymbol{z}_v \\ &+ \boldsymbol{z}_v \cdot (-g\hat{\boldsymbol{k}} + \boldsymbol{a} - \dot{\boldsymbol{\mathcal{F}}} + m^{-1}\boldsymbol{A}_v). \end{aligned} \tag{27}$$

From (23), we know that $\boldsymbol{z}_p \cdot (\boldsymbol{\mathcal{F}} - \dot{\boldsymbol{r}}_*) = -\eta G\|\boldsymbol{z}_p\|$ is already negative definite. Then, the following virtual control law $\boldsymbol{a}$ is proposed

$$\boldsymbol{a} = g\hat{\boldsymbol{k}} + \dot{\boldsymbol{\mathcal{F}}} - k_v\boldsymbol{z}_v - k_{vI}\boldsymbol{z}_{vI} - k_p\boldsymbol{z}_p, \tag{28}$$

ensuring that, for every constant uncertainty $\boldsymbol{A}_v$, the system has only one equilibrium point at $(\boldsymbol{z}_p, \boldsymbol{z}_{vI}, \boldsymbol{z}_v) = (0, k_{vI}^{-1}m^{-1}\boldsymbol{A}_v, 0)$.

Theorem 1: Applying (28), the system (24)-(26) has a UGAS equilibrium point at $(\boldsymbol{z}_p, \boldsymbol{z}_{vI}, \boldsymbol{z}_v) = (0, k_{vI}^{-1}m^{-1}\boldsymbol{A}_v, 0),\ \forall \boldsymbol{A}_v$ constant.

Proof: Consider the following closed-loop system given by (24)-(26) with (8) and (28):

$$\dot{\boldsymbol{z}}_p = \boldsymbol{\mathcal{F}} - \dot{\boldsymbol{r}}_* + \boldsymbol{z}_v, \tag{29}$$

$$\dot{\boldsymbol{z}}_{vI} = \boldsymbol{z}_v, \tag{30}$$

$$\dot{\boldsymbol{z}}_v = -k_p\boldsymbol{z}_p - k_{vI}\boldsymbol{z}_{vI} - k_v\boldsymbol{z}_v + m^{-1}\boldsymbol{A}_v. \tag{31}$$

From (27) and (28), when $\boldsymbol{A}_v = 0$, the equilibrium point $(\boldsymbol{z}_p, \boldsymbol{z}_{vI}, \boldsymbol{z}_v) = (0,0,0)$ is uniformly globally stable (UGS). The proof of asymptotic stability follows the extended Matrosov's Theorem [39]. Define $\phi_1 = \boldsymbol{z}_p$, $W_1 = V_{p,v}$, $W_2 = \boldsymbol{z}_{vI} \cdot \boldsymbol{z}_v$, $Y_1 = -k_p\eta G\|\boldsymbol{z}_p\| - k_v\boldsymbol{z}_v \cdot \boldsymbol{z}_v$ and $Y_2 = \boldsymbol{z}_v \cdot \boldsymbol{z}_v - \boldsymbol{z}_{vI} \cdot \boldsymbol{z}_p - k_v\boldsymbol{z}_{vI} \cdot \boldsymbol{z}_v - k_{vI}\boldsymbol{z}_{vI} \cdot \boldsymbol{z}_{vI}$.

As $\boldsymbol{r}_*$ is bounded, $\boldsymbol{p}$ is continuous for $(\boldsymbol{z}_p, \boldsymbol{z}_{vI}, \boldsymbol{z}_v)$ bounded, then ϕ_1, W_1 and W_2 are bounded. Besides that, if $Y_1 = 0$ then $Y_2 = -k_{vI}\boldsymbol{z}_{vI} \cdot \boldsymbol{z}_{vI} \leq 0$. If $Y_1 = Y_2 = 0$, then $(\boldsymbol{z}_p, \boldsymbol{z}_{vI}, \boldsymbol{z}_v) = (0,0,0)$. Finally, this equilibrium point is UGAS for $\boldsymbol{A}_v = 0$.

When $\boldsymbol{A}_v \neq 0$ is constant, the system (24)-(26) will have a new equilibrium point at $(\boldsymbol{z}_p, \boldsymbol{z}_{vI}, \boldsymbol{z}_v) = (0, k_{vI}^{-1}m^{-1}\boldsymbol{A}_v, 0)$. After performing a variable change $\tilde{\boldsymbol{z}}_{vI} = \boldsymbol{z}_{vI} - k_{vI}^{-1}m^{-1}\boldsymbol{A}_v$, the system (29)-(31) is rewritten as

$$\dot{\boldsymbol{z}}_p = \boldsymbol{\mathcal{F}} - \dot{\boldsymbol{r}}_* + \boldsymbol{z}_v, \tag{32}$$

$$\dot{\tilde{\boldsymbol{z}}}_{vI} = \boldsymbol{z}_v, \tag{33}$$

$$\dot{\boldsymbol{z}}_v = -k_p\boldsymbol{z}_p - k_{vI}\tilde{\boldsymbol{z}}_{vI} - k_v\boldsymbol{z}_v, \tag{34}$$

which has the same dynamics as (29)-(31) for $\boldsymbol{A}_v = 0$. Finally, (29)-(31) has a UGAS equilibrium point at $(\boldsymbol{z}_p, \boldsymbol{z}_{vI}, \boldsymbol{z}_v) = (0, k_{vI}^{-1}m^{-1}\boldsymbol{A}_v, 0)$ for all constant $\boldsymbol{A}_v$. ■

After finding $\boldsymbol{a}$ (28), we decompose it in the total thrust control input, u_T, and in a reference orientation, $\boldsymbol{o}_d$. The idea is: find $\boldsymbol{o}_d$ such that $\hat{\boldsymbol{k}}_B \| \boldsymbol{a}$ and find u_T such that $\|\dot{\boldsymbol{v}}\| = \|\boldsymbol{a}\|$. The input u_T is given by

$$u_T = m(\boldsymbol{a} \cdot \hat{\boldsymbol{k}}_B). \tag{35}$$

Next, $\boldsymbol{o}_d$ is given by the following procedure:

$$\boldsymbol{n}_z = \hat{\boldsymbol{k}} \times \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|},\ \phi_z = \cos^{-1}\left(\hat{\boldsymbol{k}} \cdot \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|}\right), \tag{36}$$

$$\boldsymbol{o}_z = \cos(\frac{\phi_z}{2}) + \boldsymbol{n}_z \sin(\frac{\phi_z}{2}), \tag{37}$$

$$\boldsymbol{o}_{\psi d} = \cos(\frac{\psi_d}{2}) + \hat{\boldsymbol{k}}_B \sin(\frac{\psi_d}{2}), \tag{38}$$

$$\boldsymbol{o}_d = \boldsymbol{o}_{\psi d}\boldsymbol{o}_z. \tag{39}$$

To put the vehicle in the desired orientation, it is necessary that its axis $\hat{\boldsymbol{k}}_B$ points in $\boldsymbol{a}$ direction. This is achieved via $\boldsymbol{o}_z$. Besides that, when finding $\boldsymbol{o}_d$ there is a free angle corresponding to the vehicle's yaw. In order to choose this parameter, a rotation ψ_d is computed around axis $\hat{\boldsymbol{k}}_B$, represented by $\boldsymbol{o}_{\psi d}$. This rotation is chosen to accomplish the secondary control goal.

The control law u_T already appears in (35). This strictly means that the Backstepping is finished given the theory [16], [24], [39]. To find a control law for the other input $\boldsymbol{u}_\tau$, we add two more steps, similar to [37].

3) Step 3: In this step, we define $\tilde{\boldsymbol{o}} = \boldsymbol{o}\boldsymbol{o}_d^* = \cos(\frac{\tilde{\phi}}{2}) + \tilde{\boldsymbol{n}}\sin(\frac{\tilde{\phi}}{2})$ as the unit quaternion orientation error. The goal of this step is to find a virtual control law for $\boldsymbol{\omega}$, let us call it $\boldsymbol{\varphi}$, such that $\boldsymbol{o} \to \boldsymbol{o}_d \iff \tilde{\boldsymbol{o}} \to 1 \iff \tilde{\phi} \to 0$ given the first-order orientation system (5).

Using $\boldsymbol{o}_d$, it is possible to find its angular velocity given the relation $\boldsymbol{\omega}_d = 2\frac{d}{dt}\ln(\boldsymbol{o}_d)$.

Let $\boldsymbol{z}_o = 2\ln(\tilde{\boldsymbol{o}}) = \tilde{\boldsymbol{n}}\tilde{\phi}$ be another representation for the orientation error expressed as a pure quaternion. Also let $V_o(\boldsymbol{z}_o) = \frac{1}{2}\boldsymbol{z}_o \cdot \boldsymbol{z}_o, V_o : \mathbb{H}_p \to \mathbb{R}$, be a Lyapunov control function with time derivative equals to

$$\dot{V}_o(\boldsymbol{z}_o) = \boldsymbol{z}_o \cdot (\boldsymbol{\varphi} - \boldsymbol{\omega}_d). \tag{40}$$

The proposed virtual control law $\boldsymbol{\varphi}$ is given by

$$\boldsymbol{\varphi} = \boldsymbol{\omega}_d - k_o \boldsymbol{z}_o, \ k_o > 0. \tag{41}$$

Applying (41) in (40) yields $\dot{V}_o(\boldsymbol{z}_o) = -k_o \boldsymbol{z}_o \cdot \boldsymbol{z}_o < 0$, $\forall \boldsymbol{z}_o \neq 0$.

4) Step 4: For the last step, consider the virtual law $\boldsymbol{\varphi}$ (41), let $\boldsymbol{z}_\omega = \boldsymbol{\omega} - \boldsymbol{\varphi}$ be the angular velocity error and $\boldsymbol{z}_{\omega I} = \int_0^t \boldsymbol{z}_\omega dt$ its integral. This yields the system

$$\dot{\boldsymbol{z}}_o = -k_o \boldsymbol{z}_o + \boldsymbol{z}_\omega, \tag{42}$$

$$\dot{\boldsymbol{z}}_{\omega I} = \boldsymbol{z}_\omega, \tag{43}$$

$$\dot{\boldsymbol{z}}_\omega = -J^{-1}(\boldsymbol{\omega} \times J\boldsymbol{\omega}) + J^{-1}\boldsymbol{u}_\tau - \dot{\boldsymbol{\varphi}} + J^{-1}\boldsymbol{A}_\omega. \tag{44}$$

Using a static feedback linearization, $\boldsymbol{u}_\tau$ is chosen as

$$\boldsymbol{u}_\tau = J(\boldsymbol{u}_{\tau,aux} + J^{-1}(\boldsymbol{\omega} \times J\boldsymbol{\omega})), \tag{45}$$

such that $\dot{\boldsymbol{\omega}} = \boldsymbol{u}_{\tau,aux} \iff \dot{\boldsymbol{z}}_\omega = \boldsymbol{u}_{\tau,aux} - \dot{\boldsymbol{\varphi}}$ for $\boldsymbol{A}_\omega = 0$, and $\boldsymbol{u}_{\tau,aux}$ is an auxiliary control law.

Let $V_{o,\omega} : \mathbb{H}_p^3 \to \mathbb{R}$ be a control Lyapunov function given by $V_{o,\omega}(\boldsymbol{z}_o, \boldsymbol{z}_{\omega I}, \boldsymbol{z}_\omega) = k_o V_o + \frac{1}{2} k_{\omega I} \boldsymbol{z}_{\omega I} \cdot \boldsymbol{z}_{\omega I} + \frac{1}{2} \boldsymbol{z}_\omega \cdot \boldsymbol{z}_\omega$, from which its time derivative yields

$$\begin{aligned} \dot{V}_{o,\omega} = {} & k_o(-k_o \boldsymbol{z}_o + \boldsymbol{z}_\omega) \cdot \boldsymbol{z}_o + k_{\omega I} \boldsymbol{z}_{\omega I} \cdot \boldsymbol{z}_\omega \\ & + \boldsymbol{z}_\omega \cdot (\boldsymbol{u}_{\tau,aux} - \dot{\boldsymbol{\varphi}} + J^{-1}\boldsymbol{A}_\omega), \ k_o, k_{\omega I} > 0. \end{aligned} \tag{46}$$

The control law $\boldsymbol{u}_{\tau,aux}$ is chosen as

$$\boldsymbol{u}_{\tau,aux} = \dot{\boldsymbol{\varphi}} - k_\omega \boldsymbol{z}_\omega - k_{\omega I} \boldsymbol{z}_{\omega I} - k_o \boldsymbol{z}_o. \tag{47}$$

Theorem 2: Applying (45) and (47) in the system (42)-(44), the point $(\boldsymbol{z}_o, \boldsymbol{z}_{\omega I}, \boldsymbol{z}_\omega) = (0, k_{\omega I}^{-1} J^{-1} \boldsymbol{A}_\omega, 0)$ becomes a UGAS equilibrium point.

Proof: Given (45) and (47), the closed-loop system is

$$\dot{\boldsymbol{z}}_o = -k_o \boldsymbol{z}_o + \boldsymbol{z}_\omega, \tag{48}$$

$$\dot{\boldsymbol{z}}_{\omega I} = \boldsymbol{z}_\omega, \tag{49}$$

$$\dot{\boldsymbol{z}}_\omega = -k_o \boldsymbol{z}_o - k_{\omega I} \boldsymbol{z}_{\omega I} - k_\omega \boldsymbol{z}_\omega + J^{-1}\boldsymbol{A}_\omega. \tag{50}$$

From (46) and (47), when $\boldsymbol{A}_\omega = 0$, the equilibrium point $(\boldsymbol{z}_o, \boldsymbol{z}_{\omega I}, \boldsymbol{z}_\omega) = (0, 0, 0)$ is UGS. The proof of asymptotic stability follows the extended Matrosov's Theorem [39]. Define $\phi_2 = \boldsymbol{z}_o$, $W_3 = V_{o,\omega}$, $W_4 = \boldsymbol{z}_{\omega I} \cdot \boldsymbol{z}_\omega$, $Y_3 = -k_o^2 \boldsymbol{z}_o \cdot \boldsymbol{z}_o - k_\omega \boldsymbol{z}_\omega \cdot \boldsymbol{z}_\omega$ and $Y_4 = \boldsymbol{z}_\omega \cdot \boldsymbol{z}_\omega - \boldsymbol{z}_{\omega I} \cdot \boldsymbol{z}_o - k_\omega \boldsymbol{z}_{\omega I} \cdot \boldsymbol{z}_\omega - k_{\omega I} \boldsymbol{z}_{\omega I} \cdot \boldsymbol{z}_{\omega I}$.

As $\boldsymbol{o}_d$ is bounded and $\boldsymbol{o}$ is continuous for bounded $(\boldsymbol{z}_o, \boldsymbol{z}_{\omega I}, \boldsymbol{z}_\omega)$, then ϕ_2, W_3, and W_4 are bounded. Besides that, if $Y_3 = 0$ then $Y_4 = -k_{\omega I} \boldsymbol{z}_{\omega I} \cdot \boldsymbol{z}_{\omega I} \leq 0$. If $Y_3 = Y_4 = 0$, then $(\boldsymbol{z}_o, \boldsymbol{z}_{\omega I}, \boldsymbol{z}_\omega) = (0, 0, 0)$. Therefore, this equilibrium point is UGAS for $\boldsymbol{A}_\omega = 0$.

For constant $\boldsymbol{A}_\omega \neq 0$, the system will have a new equilibrium point $(\boldsymbol{z}_o, \boldsymbol{z}_{\omega I}, \boldsymbol{z}_\omega) = (0, k_{\omega I}^{-1} J^{-1} \boldsymbol{A}_\omega, 0)$. After performing a variable change $\tilde{\boldsymbol{z}}_{\omega I} = \boldsymbol{z}_{\omega I} - k_{\omega I}^{-1} J^{-1} \boldsymbol{A}_\omega$, the system (48)-(50) can be rewritten as

$$\dot{\boldsymbol{z}}_o = -k_o \boldsymbol{z}_o + \boldsymbol{z}_\omega, \tag{51}$$

$$\dot{\tilde{\boldsymbol{z}}}_{\omega I} = \boldsymbol{z}_\omega, \tag{52}$$

$$\dot{\boldsymbol{z}}_\omega = -k_o \boldsymbol{z}_o - k_{\omega I} \tilde{\boldsymbol{z}}_{\omega I} - k_\omega \boldsymbol{z}_\omega. \tag{53}$$

which has the same dynamics as (48)-(50) for $\boldsymbol{A}_\omega = 0$. Finally, the system (48)-(50) has a UGAS equilibrium point at $(\boldsymbol{z}_o, \boldsymbol{z}_{\omega I}, \boldsymbol{z}_\omega) = (0, k_{\omega I}^{-1} J^{-1} \boldsymbol{A}_\omega, 0)$ for constant $\boldsymbol{A}_\omega$. ■

C. Summary

The controller is designed using constructive methods. It incorporates the vector fields from [15] in a backstepping with integral action.

Each step had its asymptotically convergence demonstrated. Considering (29)-(31), with (8), and (48)-(50), the closed-loop system can be written as follows

$$\dot{\boldsymbol{z}}_p = \boldsymbol{\mathcal{F}} - \dot{\boldsymbol{r}}_* + \boldsymbol{z}_v, \tag{54}$$

$$\dot{\boldsymbol{z}}_{vI} = \boldsymbol{z}_v, \tag{55}$$

$$\dot{\boldsymbol{z}}_v = -k_p \boldsymbol{z}_p - k_{vI} \boldsymbol{z}_{vI} - k_v \boldsymbol{z}_v + m^{-1}\boldsymbol{A}_v, \tag{56}$$

$$\dot{\boldsymbol{z}}_o = -k_o \boldsymbol{z}_o + \boldsymbol{z}_\omega, \tag{57}$$

$$\dot{\boldsymbol{z}}_{\omega I} = \boldsymbol{z}_\omega, \tag{58}$$

$$\dot{\boldsymbol{z}}_\omega = -k_o \boldsymbol{z}_o - k_{\omega I} \boldsymbol{z}_{\omega I} - k_\omega \boldsymbol{z}_\omega + J^{-1}\boldsymbol{A}_\omega. \tag{59}$$

Theorem 3: The system (54)-(59) has a UGAS equilibrium point at $(\boldsymbol{z}_p, \boldsymbol{z}_{vI}, \boldsymbol{z}_v, \boldsymbol{z}_o, \boldsymbol{z}_{\omega I}, \boldsymbol{z}_\omega) = (0, k_{vI}^{-1} \frac{\boldsymbol{A}_v}{m}, 0, 0, k_{\omega I}^{-1} J^{-1} \boldsymbol{A}_\omega, 0)$ for all constant $\boldsymbol{A}_v, \boldsymbol{A}_\omega$.

Proof: Let $V : \mathbb{H}_p^6 \to \mathbb{R}$, $V(\boldsymbol{z}_p, \boldsymbol{z}_{vI}, \boldsymbol{z}_v, \boldsymbol{z}_o, \boldsymbol{z}_{\omega I}, \boldsymbol{z}_\omega) = V_{p,v} + V_{o,\omega}$ be a candidate Lyapunov function for the system (54)-(59). Considering null uncertainties, $\boldsymbol{A}_v = \boldsymbol{A}_\omega = 0$, its time derivative yields

$$\begin{aligned} & \dot{V} = -k_p \eta G \|\boldsymbol{z}_p\| - k_v \boldsymbol{z}_v \cdot \boldsymbol{z}_v - k_r \boldsymbol{z}_o \cdot \boldsymbol{z}_o - k_\omega \boldsymbol{z}_\omega \cdot \boldsymbol{z}_\omega, \\ & \dot{V} \leq 0, \ \forall (\boldsymbol{z}_p, \boldsymbol{z}_{vI}, \boldsymbol{z}_v, \boldsymbol{z}_o, \boldsymbol{z}_{\omega I}, \boldsymbol{z}_\omega), \end{aligned} \tag{60}$$

ensuring that the origin is a UGS equilibrium point. When $\boldsymbol{A}_v, \boldsymbol{A}_\omega$ are not null, but constant, the system has a new equilibrium point at $(\boldsymbol{z}_p, \boldsymbol{z}_{vI}, \boldsymbol{z}_v, \boldsymbol{z}_o, \boldsymbol{z}_{\omega I}, \boldsymbol{z}_\omega) = (0, k_{vI}^{-1} \frac{\boldsymbol{A}_v}{m}, 0, 0, k_{\omega I}^{-1} J^{-1} \boldsymbol{A}_\omega, 0)$, which is UGAS considering the results from IV-B.2 and IV-B.4 via the extended Matrosov's Theorem [16]. ■

V. Results and Discussions

In order to illustrate the methodology, we show next computational simulations using MATLAB & SIMULINK and ROS. A detailed video of the results is available at youtu.be/r9jk025wkzQ.

A. Matlab & Simulink

In this simulation, we consider $m = 0.7\text{kg}$, $J = \text{diag}(1.2416, 1.2416, 1.2416)\text{kgm}^2$, $g = 9.81\text{m/s}^2$. The target curve parametric equation is $\boldsymbol{r}(s,t) = \boldsymbol{r}_i(s,t) + \boldsymbol{r}_j(s) + \boldsymbol{r}_k(s)$, with $\boldsymbol{r}_i(s,t) = \hat{\boldsymbol{i}}(2\sin(0.5t) + 0.05(10\cos(s))^2)\text{m}$, $\boldsymbol{r}_j(s) = \hat{\boldsymbol{j}}10\cos(s)\text{m}$ and $\boldsymbol{r}_k(s) = \hat{\boldsymbol{k}}10\sin(s)\text{m}$.

The system's initial conditions are $\boldsymbol{p}(0) = \hat{\boldsymbol{i}}20 - \hat{\boldsymbol{j}}10 + \hat{\boldsymbol{k}}5$, $\boldsymbol{o}(0) = 1 \iff 2\ln(\boldsymbol{o}(0)) = 0$, $\boldsymbol{v}(0) = 0$ and $\boldsymbol{\omega}(0) = 0$. The chosen desired yaw angle is $\psi_d = 0$. The controller gains are $v_r = 2\text{m/s}, k_G = 15\text{m}^{-1}, k_p = 0.1\text{s}^{-2}, k_{vI} = 2\text{s}^{-2}, k_v = 10\text{s}^{-1}, k_o = 100\text{s}^{-2}, k_{\omega I} = 10\text{s}^{-2}$, and $k_\omega = 200\text{s}^{-1}$.

This workspace is collision-free and it starts with null uncertainties. Then, at $t = 50\text{s}$, the system is disturbed with constant $\boldsymbol{A}_v = (\hat{\boldsymbol{i}}7 + \hat{\boldsymbol{j}}35 - \hat{\boldsymbol{k}}7)\text{N}$ and $\boldsymbol{A}_\omega = \hat{\boldsymbol{k}}5\text{Nm}$.

Fig. 2 shows the evolution of the states in time; Fig. 3 shows the error signals; and Fig. 4 shows the control signals.

The error signals converge to zero until $t = 50$s, when the system is disturbed and next rejects the constant disturbances returning to an error equal to zero. This means that the system's states and trajectory converged to the target path.

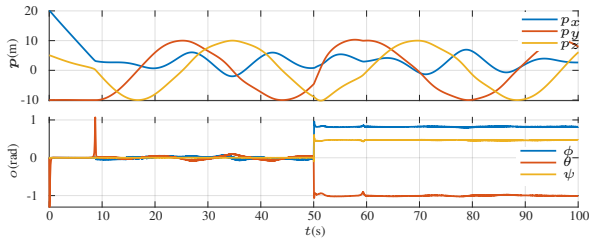

Fig. 2. Matlab & Simulink: Evolution in time of position and orientation.

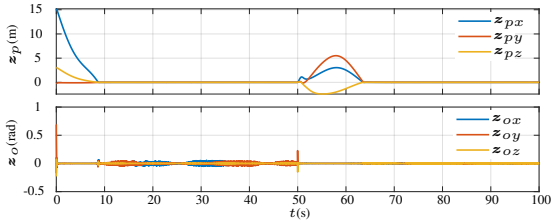

Fig. 3. Matlab & Simulink: Evolution in time of error signals.

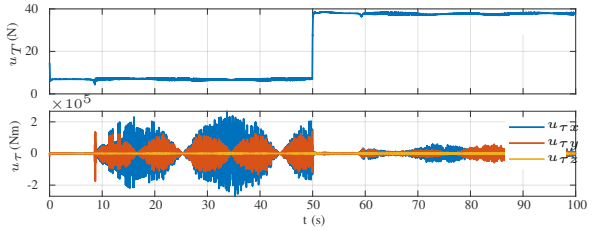

Fig. 4. Matlab & Simulink: Evolution in time of control signals.

B. ROS

In this ROS-based simulation, we used the vector_field stack simulator[1]. The quadcopter model in this simulator receives total thrust and angular velocities inputs, i.e. it only allows an acro-rate control mode. Then, it was controlled using the Backstepping controller with the first three steps, feeding the system directly with the found angular velocities.

The goal curve has the parametric representation $\mathbf{r}(s) = (\hat{\boldsymbol{i}}5\cos(s) + \hat{\boldsymbol{j}}2\sin(s) + \hat{\boldsymbol{k}}(\cos(4s) + 2))$m. Fig. 5 shows the target curve $\boldsymbol{\mathcal{C}}$ in black, the result trajectory $\boldsymbol{p}(t)$ in blue, and the present obstacles $\boldsymbol{\mathcal{O}}$ in red.

The avoidance parameters are: $\lambda = 0.6$ m, $D_{in} = 0.8$ m and $D_{in0} = 1.2$ m. Fig. 6 shows the distance $\|\boldsymbol{D}\|$ in blue and the distance $\|\boldsymbol{D}_o\|$ in red.

The controller gains are $v_r = 0.5\text{m/s}, k_G = 8\text{m}^{-1}, k_p = 0.01\text{s}^{-2}, k_{vI} = 0, k_v = 10\text{s}^{-1}$ and $k_o = 10\text{s}^{-2}$, with ψ_d aligned with the field direction, $\psi_d = \text{atan2}(\mathcal{F}_j, \mathcal{F}_i)$[2].

[1]Available at github.com/adrianomcr/vectorfield_stack [15].

[2]$\text{atan2}(\Delta_y, \Delta_x)$ is the arctangent, and $\mathcal{F}_j, \mathcal{F}_i$ are components of $\boldsymbol{\mathcal{F}}$ in directions $\hat{\boldsymbol{j}}, \hat{\boldsymbol{i}}$ respectively.

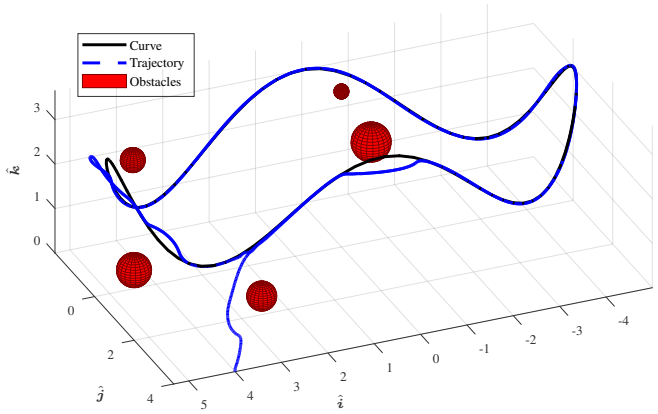

Fig. 5. ROS: System trajectory in blue, target curve in black, and obstacles in red.

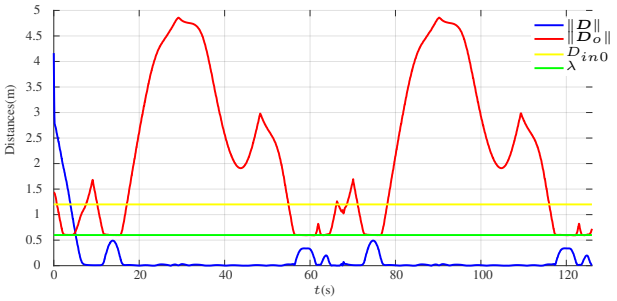

Fig. 6. ROS: Curve distance's norm in blue, obstacle distance's norm in red, D_{in0} in yellow, and λ in green.

After the simulation begins, the vehicle deviates from the first obstacle away from the curve. Then, after reaching the curve, it deviates from other obstacles. In all cases, the deviation is in the direction closest to the curve field.

Initially, the distance to the curve (blue) is approximately 4m. This rapidly decreases to and stays at approximately zero. It increases when the distance to the obstacle (red) is smaller than D_{in0} (yellow). In these stages, the obstacle is contoured at a fixed distance λ (green). When the deviation is finished, the distance to the curve decreases again.

VI. Conclusion and Future Works

In this work, we proposed a non-linear controller to drive a UAV quadcopter to follow and circulate a given time-varying curve. The controller incorporates a novel collision-free artificial vector field [14], [15] into a backstepping with integral actions [16], allowing the closed-loop system to deviate from obstacles and reject constant uncertainties in a path-following strategy. The main contributions of this work rely on the integration between the vector field guidance and the non-linear backstepping with stability proofs, the consideration of torques as control inputs[3], and the applications of the novel field.

For future work, we propose to evaluate the robustness over bounded and unknown uncertainties to assure stability and to perform real vehicle experiments.

[3]Even though the angular velocities are internally calculated and can be used as control input in acro-rate control mode.

REFERENCES

[1] A. Williams and O. Yakimenko, "Persistent mobile aerial surveillance platform using intelligent battery health management and drone swapping," in *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, pp. 237–246, 2018.

[2] V. Duggal, M. Sukhwani, K. Bipin, G. S. Reddy, and K. M. Krishna, "Plantation monitoring and yield estimation using autonomous quadcopter for precision agriculture," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5121–5127, 2016.

[3] M. Rabah, A. Rohan, S. A. S. Mohamed, and S.-H. Kim, "Autonomous moving target-tracking for a uav quadcopter based on fuzzy-pi," *IEEE Access*, vol. 7, pp. 38407–38419, 2019.

[4] S. E. Eid and S. Sham Dol, "Design and development of lightweight-high endurance unmanned aerial vehicle for offshore search and rescue operation," in *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, pp. 1–5, 2019.

[5] V. R. Miranda, A. Rezende, T. L. Rocha, H. Azpúrua, L. C. Pimenta, and G. M. Freitas, "Autonomous navigation system for a delivery drone," *Journal of Control, Automation and Electrical Systems*, vol. 33, no. 1, pp. 141–155, 2022.

[6] A. M. Rezende, V. R. Miranda, H. N. Machado, A. C. Chiella, V. M. Gonçalves, and G. M. Freitas, "Autonomous system for a racing quadcopter," in *2019 19th International Conference on Advanced Robotics (ICAR)*, pp. 1–6, IEEE, 2019.

[7] M. Iacono and A. Sgorbissa, "Path following and obstacle avoidance for an autonomous UAV using a depth camera," *Robotics and Autonomous Systems*, vol. 106, pp. 38–46, Aug. 2018.

[8] J. P. Wilhelm and G. Clem, "Vector Field UAV Guidance for Path Following and Obstacle Avoidance with Minimal Deviation," *Journal of Guidance, Control, and Dynamics*, vol. 42, pp. 1848–1856, Aug. 2019.

[9] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, pp. 2520–2525, May 2011.

[10] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 5420–5425, Dec 2010.

[11] D. Zhou and M. Schwager, "Vector field following for quadrotors using differential flatness," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6567–6572, May 2014.

[12] S. Zhao, X. Wang, Z. Lin, D. Zhang, and L. Shen, "Integrating vector field approach and input-to-state stability curved path following for unmanned aerial vehicles," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–8, 2018.

[13] B. Rubí, R. Pérez, and B. Morcego, "A survey of path following control strategies for uavs focused on quadrotors," *Journal of Intelligent & Robotic Systems*, vol. 98, no. 2, pp. 241–265, 2020.

[14] A. M. C. Rezende, V. M. Goncalves, and L. C. A. Pimenta, "Constructive time-varying vector fields for robot navigation," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 852–867, 2022.

[15] A. H. D. Nunes, A. M. C. Rezende, G. P. Cruz, G. M. Freitas, V. M. Gonçalves, and L. C. A. Pimenta, "Vector field for curve tracking with obstacle avoidance," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 2031–2038, 2022.

[16] R. Skjetne and T. I. Fossen, "On integral control in backstepping: Analysis of different techniques," in *Proceedings of the 2004 American control conference*, vol. 2, pp. 1899–1904, IEEE, 2004.

[17] W. Yao, H. G. de Marina, B. Lin, and M. Cao, "Singularity-Free Guiding Vector Field for Robot Navigation," *IEEE Transactions on Robotics*, vol. 37, pp. 1206–1221, Aug. 2021.

[18] V. M. Gonçalves, L. C. Pimenta, C. A. Maia, B. C. Dutra, and G. A. Pereira, "Vector fields for robot navigation along time-varying curves in n-dimensions," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 647–659, 2010.

[19] W. Yao, B. Lin, B. D. O. Anderson, and M. Cao, "Guiding vector fields for following occluded paths," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, pp. 4091–4106, 2022.

[20] W. Yao, B. Lin, and M. Cao, "Integrated Path Following and Collision Avoidance Using a Composite Vector Field," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, Dec. 2019.

[21] D. A. Lawrence, E. W. Frew, and W. J. Pisano, "Lyapunov Vector Fields for Autonomous Unmanned Aircraft Flight Control," *Journal of Guidance, Control, and Dynamics*, vol. 31, pp. 1220–1229, Sept. 2008.

[22] D. Nelson, D. Barber, T. McLain, and R. Beard, "Vector Field Path Following for Miniature Air Vehicles," *IEEE Transactions on Robotics*, vol. 23, pp. 519–529, June 2007.

[23] A. M. Rezende, V. M. Gonçalves, A. H. Nunes, and L. C. Pimenta, "Robust quadcopter control with artificial vector fields," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6381–6387, IEEE, 2020.

[24] H. K. Khalil, "Nonlinear systems third edition," *Patience Hall*, vol. 115, 2002.

[25] N. S. Özbek, M. Önkol, and M. Önder Efe, "Feedback control strategies for quadrotor-type aerial robots: a survey," *Transactions of the Institute of Measurement and Control*, vol. 38, no. 5, pp. 529–554, 2016.

[26] S. Bouabdallah, A. Noth, and R. Siegwart, "Pid vs lq control techniques applied to an indoor micro quadrotor," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2451–2456 vol.3, 2004.

[27] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke, "A prototype of an autonomous controller for a quadrotor uav," in *2007 European Control Conference (ECC)*, pp. 4001–4008, 2007.

[28] G. F. Salierno, "Whole-body backstepping control of a quadrotor UAV for trajectory tracking," Master's thesis, Universidade Federal de Minas Gerais, 2018.

[29] C. K. Tan, J. Wang, Y. C. Paw, and T. Y. Ng, "Tracking of a moving ground target by a quadrotor using a backstepping approach based on a full state cascaded dynamics," *Applied Soft Computing*, vol. 47, pp. 47–62, 2016.

[30] I. A. Raptis, K. P. Valavanis, and W. A. Moreno, "A novel nonlinear backstepping controller design for helicopters using the rotation matrix," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 2, pp. 465–473, 2011.

[31] M. Huang, B. Xian, C. Diao, K. Yang, and Y. Feng, "Adaptive tracking control of underactuated quadrotor unmanned aerial vehicles via backstepping," in *Proceedings of the 2010 American Control Conference*, pp. 2076–2081, 2010.

[32] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2247–2252, 2005.

[33] F. F. Silva, J. J. Quiroz-Omaña, and B. V. Adorno, "Dynamics of mobile manipulators using dual quaternion algebra," *Journal of Mechanisms and Robotics*, pp. 1–24, 2022.

[34] B. V. Adorno, "Robot kinematic modeling and control based on dual quaternion algebra—part i: Fundamentals.," 2017.

[35] H. Abaunza, P. Castillo, A. Victorino, and R. Lozano, "Dual quaternion modeling and control of a quad-rotor aerial manipulator," *Journal of Intelligent & Robotic Systems*, vol. 88, no. 2, pp. 267–283, 2017.

[36] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An integral predictive/nonlinear h_∞ control structure for a quadrotor helicopter," *Automatica*, vol. 46, no. 1, pp. 29–39, 2010.

[37] G. F. Salierno and G. V. Raffo, "Whole-body backstepping control with integral action of a quadrotor uav," in *Proceedings of the XIII Brazilian Symposium on Intelligent Automation (SBAI 2017), Porto Alegre, Brazil*, pp. 2157–2164, 2017.

[38] V. Mistler, A. Benallegue, and N. M'sirdi, "Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback," in *Proceedings 10th IEEE international workshop on robot and human interactive communication. Roman 2001 (Cat. no. 01th8591)*, pp. 586–593, IEEE, 2001.

[39] A. Loria, E. Panteley, D. Popovic, and A. R. Teel, "An extension of Matrosov's theorem with application to stabilization of nonholonomic control systems," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 2, pp. 1528–1533, IEEE, 2002.