Shape visual servoing of a cable suspended between two drones

Lev Smolentsev, Alexandre Krupa, François Chaumette

Abstract— **In this paper, we propose a shape visual servoing approach for manipulating a suspended cable attached between two quadrotor drones. A leader-follower control strategy is presented, where a human operator controls the rigid motion of the cable by teleoperating one drone (the leader), while the second drone (the follower) performs a shape visual servoing task to autonomously apply a desired deformation to the cable. The proposed shape visual servoing approach uses an RGB-D camera embedded on the follower drone and has the advantage to rely on a simple geometrical model of the cable that only requires the knowledge of its length. In the same time, our control strategy maintains the best visibility of the cable in the camera field of view. A robust image processing pipeline allows detecting and tracking in real-time the cable shape from the data provided by the onboard RGB-D camera. Experimental results demonstrate the effectiveness of the proposed visual control approach to shape a flexible cable into a desired shape. In addition, we demonstrate experimentally that such system can be used to perform an aerial transport task by grasping with the cable an object fitted with a hook, then moving and releasing it at another location.**

I. INTRODUCTION

The manipulation of Deformable Linear Objects (DLO) such as cables by robotic manipulators or drones is a topic of growing interest. There is an increasing industrial need for aerial manipulation of cables. For example, drones can be used to lay high-voltage transmission lines from one pylon to another [1], [2]. Performing such an operation on a pylon at high altitude presents a major risk to human operators. It is therefore safer and more efficient to replace human operators with a drone to carry out this type of task. In another context, drones can be used to search for and rescue people trapped in debris or collapsed buildings [3]. In these situations, it may be necessary to employ drones to manipulate DLO such as ropes and straps for removing debris. In a different field, there is an increasing need for a more efficient logistics and transportation industry and more particularly in the context of e-Commerce. One of the main challenges for logistics providers is the last mile delivery step (delivery of packages to the final customer from the e-Retailer hub) and the use of drones can significantly reduce its delivery times and costs [4]. In the literature, several research projects have been carried out for transporting objects with drones with the aim of increasing their possible payload. The solutions consist in a flying gripper made of a drone that is connected with a cable to a point load [5] or by a cable-suspended parallel robot actuated by multiple drones [6], [7], [8], [9]. However, in these works, the cables are assumed to be taut under the weight of an attached platform or a rigid beam. Another approach is the use of one drone that is equipped with a serial manipulator [10] or a delta robot [11] [12] serving as an onboard end-effector but whose weight reduces considerably the possible payload. In our work, we propose to use a flexible cable whose extremities are attached to a pair of drones for grasping and transporting an object. Using a light deformable cable allows the system to be energy efficient and the softness of the cable avoids breaking the drone or damaging the environment in case of collision, which are advantageous properties compared to previous approaches.

Despite all the benefits of using deformable cables or other soft bodies, controlling their shape remains an open problem. In [13], cable collision avoidance is achieved by controlling its lowest point using the catenary model and drone relative pose measurements, but this method does not manage the entire cable shape. In [14], the authors proposed a flying gripper made of two drones linked by a slack cable. The cable is again modelled with the catenary model whose parameters are estimated from the drone odometry. The main drawbacks of the proposed approach are the hard constraints on the motion of the drones which are forced to follow the same yaw angle and same altitude. In their next work the authors of [15] proposed a multi-catenary robot to fold knots forming a kind of net potentially convenient for objects transportation but only simulation results were presented. In opposite, a planning control strategy is proposed in [16] to avoid knotting of the power tether cables of a multi-drone system formation. Other methods have been proposed for controlling multiple drones carrying a flexible cable, either to follow setpoints [17] or to replicate a leader trajectory while maintaining constant distances [18]. However, these approaches do not control the cable shape and were only validated through simulations. Finally, a system composed of two quadrotors connected with a cable is proposed in [19]. A dynamic model was developed which is based on a discrete representation of a deformable and extensible cable. This model is composed of lumped masses connected by linear springs through passive spherical joints. The system produces a set of flat outputs that is used to design a control scheme to manipulate the cable. The main drawback of this approach, which was only tested through numerical simulations, is the increasing complexity when the number of segments used to model the cable increases, making the closed-form solutions more and more complex as the cable is divided into smaller

This work carried out in the MAMBO project has received a French government support granted to the Labex CominLabs excellence laboratory and managed by the National Research Agency in the "Investing for the Future" program under reference ANR-10-LABX-07-01.

Experiments presented in this paper were carried out thanks to a platform of the Robotex 2.0 French research infrastructure.

L. Smolentsev, A. Krupa, and F. Chaumette are with Inria, Univ Rennes, CNRS, IRISA, Campus de Beaulieu, 35042 Rennes, France Firstname.Name@inria.fr

parts.

In this paper we propose to control the shape of the cable by visual servoing unlike all the works mentioned above that did not use on-board cameras on their drones. Our visual servoing approach relies on a geometrical model of the suspended cable that we chose as simple as possible: a parabola. The modelling of the interaction matrix that relates the variation of the coefficients of the parabola to the velocities of the cable extremities was detailed in our previous work [20] where only one extremity of the cable was manipulated by a grounded robot arm. To the best of our knowledge, we propose in this paper the first onboard shape visual servo control approach for manipulating and deforming a cable to a desired shape configuration with quadrotor drones. The contributions of our work are:

- A shape visual servoing control method using a parabola model for a cable between two drones;
- A leader-follower strategy with one teleoperated drone and one autonomously adjusting the cable shape;
- An image processing method for real-time shape tracking and control using an RGB-D camera onboard the follower drone;
- Validation through real experiments, including object transport by two drones.

The rest of the paper is organised as follows: Section II presents the high-level controller of our bi-drone system that corresponds to the shape visual servoing part. Section III details the low-level controller that generates the speeds to be applied to the drone propellers from the control inputs provided by the shape visual servoing. Section IV presents our image processing pipeline to detect and track in real-time the shape of the cable from the data provided by the onboard RGB-D camera. Section V describes the experimental setup and shows the results obtained from real-flying experiences. Finally Section VI concludes this paper.

II. Shape visual servoing of the cable

In this section, we present our visual servoing approach for controlling the shape of a cable suspended between two drones.

A. Parabola model of the cable

Each extremity of the cable is attached to a drone as can be seen in Fig. 1. The attachment points $\mathbf{p}_f$ and $\mathbf{p}_l$ are considered as passive ball joints. We propose to model the cable that is subjected to gravity by a simple parabola expressed in the reference frame $\mathcal{F}_t$, whose origin coincides with the point $\mathbf{p}_f$. The orientation of this frame with respect to an arbitrary world frame $\mathcal{F}_\mathrm{w}$ whose z axis is vertical is given by the yaw angle α and the roll angle ϕ of the plane containing the cable. The coordinates in $\mathcal{F}_\mathrm{w}$ of any point ${}^\mathrm{w}\mathbf{p}$ of the cable are given by:

$$ {}^\mathrm{w}\mathbf{p} = {}^\mathrm{w}\mathbf{p}_f + \mathbf{R}(\phi)\mathbf{R}(\alpha)\,{}^t\mathbf{p} \tag{1} $$

where $\mathbf{R}(\phi)$ and $\mathbf{R}(\alpha)$ are the roll and pitch rotation matrices describing the orientation of the cable plane, and ${}^t\mathbf{p} = (0, {}^ty, a{}^ty^2 + b{}^ty)$ represents the parabola model of the cable

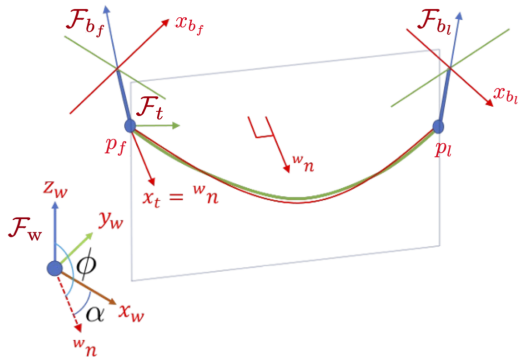


Fig. 1: Schematic illustration of the system consisting of a flexible cable attached to 2 drones. The parabola used to model the cable is depicted in red. $\mathcal{F}_t$ stands for the cable frame, $\mathcal{F}_{b_f}$ and $\mathcal{F}_{b_l}$ for the body frames of the 2 drones. The normal vector ${}^\mathrm{w}\mathbf{n}$ of the plane containing the cable, which also corresponds to the $\mathbf{x}$ axis of $\mathcal{F}_t$, is expressed in $\mathcal{F}_\mathrm{w}$ thanks to the orientation angles α and ϕ.

expressed in $\mathcal{F}_t$. To control the shape of the cable we define $\mathbf{s} = (a, b, \alpha)$ being the vector of visual features to regulate to a desired value $\mathbf{s}^* = (a^*, b^*, \alpha^*)$. In our previous work that used a serial robot for manipulating a cable [20], we assumed that the cable was lying in a perfect vertical plane, which means that ϕ was strictly zero and $\mathbf{R}(\phi)$ the identity matrix. However, this assumption no longer holds when the cable is manipulated by drones, since they generate rolling motion of the cable plane due to aerodynamic disturbances, even with the presence of passive ball joints in the cable attachment points. The method to estimate the parameters a, b, α and the rolling angle ϕ is described in Section IV.

For this non-vertical plane case, the relation between the velocities $\mathbf{v}_l = {}^\mathrm{w}\dot{\mathbf{p}}_l, \mathbf{v}_f = {}^\mathrm{w}\dot{\mathbf{p}}_f$ of the attachment points and the variation of visual features $\dot{\mathbf{s}}$ is formulated by:

$$ \mathbf{v}_l = \mathbf{v}_f + \mathbf{R}(\phi)\mathbf{M}\dot{\mathbf{s}} + \mathbf{N}\dot{\phi} \tag{2} $$

To keep it simple, we consider at this modeling step that the velocity $\dot{\phi}$ can be neglected and that ϕ always remains small. We will see in Section V that our system is robust to such approximations. In that case, matrix $\mathbf{M}$ is given by:

$$ \mathbf{M} = \begin{bmatrix} -k_1 \sin\alpha & -k_2 \sin\alpha & -D\cos\alpha \\ k_1 \cos\alpha & k_2 \cos\alpha & -D\sin\alpha \\ n_1 & n_2 & 0 \end{bmatrix} \tag{3} $$

with all its terms and properties developed and explained in [20]. It is worth recalling that D stands for the span of the cable (the horizontal distance between the attachment points) that can be estimated from the parabola coefficients a, b and the known length L of the cable, which is the only *a priori* knowledge required in the determination of $\mathbf{M}$.

B. Shape visual servoing control laws

To deform the cable to some desired shape $\mathbf{s}^*$, we propose a visual servoing control law generating the control velocity of the attachment point $\mathbf{p}_f$ to drive to zero the shape error

$\mathbf{e} = \mathbf{s} - \mathbf{s}^*$ of the cable. It is clear from (2) that the variation of $\mathbf{s}$ is achieved thanks to the relative velocity $\mathbf{v}_l - \mathbf{v}_f$ of the attachment points. This means that it is possible to carry out the shaping task with only the drone b_f, while the other performs another navigation task. In particular, we use a leader-follower control strategy in which the drone b_l, which we call the leader, is remotely controlled by a human operator and the drone b_f (the follower) is equipped with an onboard RGB-D camera and is controlled by visual servoing to autonomously apply a desired shape to the cable. This allows to autonomously maintain the desired shape of the cable while the operator controls its rigid movement by teleoperating the leader drone. In practice, we use 2 quadrotor drones each offering 4 degrees of freedom (DOF) (3 translations and one decoupled yaw rotation). To perform the shaping task of the cable by the follower drone, we propose to apply for its translational velocity the following control law:

$$\mathbf{v}_f^* = \widehat{\mathbf{v}}_l + \lambda \mathbf{R}(\phi)\mathbf{M}\mathbf{e} + \mu \mathbf{R}(\phi)\mathbf{M} \sum_i^N \mathbf{e}_i \tag{4}$$

where $\widehat{\mathbf{v}}_l$ is an estimation of the velocity of the leader drone, λ is the control gain of the visual servoing, and the right part corresponds to an integral term with gain μ computed from the past visual errors memorized in a sliding window of size N. In practice, we set $\widehat{\mathbf{v}}_l$ as the control velocity of the leader that is defined by the operator through a joystick. The integral term is added to compensate the presence of unmodelled flight disturbances and possible drift between the actual leader velocity and its teleoperation reference.

Since the yaw angle ψ_{b_f} of the follower drone is decoupled from the yaw angle α of the cable due to the passive ball joint, it is not used in the cable shaping task. We therefore propose to control it to autonomously maintain the best visibility of the cable in the center of the image with the control law:

$$\dot{\psi}_{b_f}^* = -\lambda(\psi_{b_f} - \psi_{b_f}^*) - \mu \sum_i^N (\psi_{b_{f_i}} - \psi_{b_f}^*) \tag{5}$$

where $\psi_{b_f}^* = \alpha^*$ is the desired yaw angle of the drone and a similar integral term is added. This secondary task that is fully decoupled from the shaping task aligns the camera optical axis on the cable plane. In this case, the cable projection seen on the RGB image corresponds to a straight line vertically centered in the image and not a parabola. This is not an issue since the method proposed in Section IV for tracking the features a, b and α uses the depth data provided by the onboard RGB-D camera.

III. Low-level controller of the follower drone

In this section, we detail how to generate the propeller velocities of the follower drone from the translational and yaw velocities provided by the visual control laws.

A. Drone modelling

The system is described by the body frames $\mathcal{F}_{b_f}$ and $\mathcal{F}_{b_l}$ of the follower and leader drones respectively. The origin of these frames are the center of gravity (COG) of each drone and their control velocities are expressed in the world frame $\mathcal{F}_\mathrm{w}$. The dynamics of the drone b_i with $i \in \{f, l\}$ holding a cable is described by the following system of equations:

$$\begin{aligned}
{}^\mathrm{w}\dot{\mathbf{p}}_i &= \mathbf{v}_i \\
\dot{\mathbf{v}}_i &= \frac{f_i}{m_i} \begin{bmatrix} 2(q_{w_i} q_{y_i} + q_{x_i} q_{z_i}) \\ 2(q_{y_i} q_{z_i} - q_{w_i} q_{x_i}) \\ 1 - 2(q_{x_i}^2 + q_{y_i}^2) \end{bmatrix} - g\mathbf{e}_3 + {}^\mathrm{w}\mathbf{f}_{c_i} \\
\dot{\mathbf{q}}_i &= \frac{1}{2} \begin{bmatrix} 0 \\ \boldsymbol{\omega}_i \end{bmatrix} \otimes \mathbf{q}_i \\
\dot{\boldsymbol{\omega}}_i &= \mathbf{I}_{b_i}^{-1}(\boldsymbol{\tau}_i - \boldsymbol{\omega}_i \times \mathbf{I}_{b_i}\boldsymbol{\omega}_i + {}^i\boldsymbol{\tau}_{c_i})
\end{aligned} \tag{6}$$

where f_i and $\boldsymbol{\tau}_i$ are the total thrust and body moment expressed in $\mathcal{F}_{b_i}$, $g\mathbf{e}_3$ is the gravity vector with $\mathbf{e}_3 = (0, 0, 1)$, m_i stands for the mass of the quadrotor b_i and $\mathbf{I}_{b_i}$ represents its inertia matrix. The unit quaternion $\mathbf{q}_i = (q_{w_i}, q_{x_i}, q_{y_i}, q_{z_i})$ is used to represent the orientation of the body frame $\mathcal{F}_{b_i}$ with respect to the world frame $\mathcal{F}_\mathrm{w}$. It can also be represented by the rotation matrix $\mathbf{R}_i(\phi_{b_i}, \theta_{b_i}, \psi_{b_i}) = \mathbf{f}(\mathbf{q}_i)$, with $(\phi_{b_i}, \theta_{b_i}, \psi_{b_i})$ being the drone roll, pitch, and yaw angles expressed in $\mathcal{F}_\mathrm{w}$. The vector $\boldsymbol{\omega}_i$ denotes the angular velocity of the body frame of the quadrotor. Finally, ${}^\mathrm{w}\mathbf{f}_{c_i}$ and ${}^i\boldsymbol{\tau}_{c_i}$ represent the cable tension wrench due to its weight and an eventual hanging load on the cable. In our work this wrench is processed as an unknown disturbance on the drone b_i. To map the total thrust f_i and the body torques $\boldsymbol{\tau}_i$ to the propeller velocities $\boldsymbol{\omega}_{prop_i}$, the inverse allocation matrix $\mathbf{A}_i^{-1}$ is used with $\mathbf{A}_i$ given by (under the assumption that the barycenter of the drone coincides with its COG):

$$\begin{pmatrix} f_i \\ \boldsymbol{\tau}_i \end{pmatrix} = k \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & l & 0 & -l \\ -l & 0 & l & 0 \\ c & -c & c & -c \end{bmatrix} \boldsymbol{\omega}_{prop_i} \tag{7}$$

where k and c are aerodynamic coefficients of the propellers that can be identified following the procedure described in [21], l is the distance between the COG of the quadrotor and the propellers. These equations (6) and (7) come from the well-known dynamics of drones [21], [22].

B. Follower drone controller

The main idea behind the control of the system (6) is to drive the quadrotor COG $\mathbf{p}_f$ to a desired position in $\mathcal{F}_\mathrm{w}$ to achieve the desired cable shape $\mathbf{s}^*$. It can be done by designing a thrust vector $f\mathbf{R}_f\mathbf{e}_3$ to track the desired translation $\mathbf{p}_f^*$ of the drone and its velocity $\mathbf{v}_f^*$. At the same time, the quadrotor torques $\boldsymbol{\tau}$ are controlled by aligning its body orientation with the desired one denoted $\mathbf{R}_f^*$ to apply the thrust in the right direction and drive the yaw angle ψ_{b_f} to the desired value $\psi_{b_f}^*$. Note from (6) that the quadrotor is underactuated and has only four controllable DOFs, which are ${}^\mathrm{w}\mathbf{p}_f$ and ψ_{b_f}. One common solution to

this problem is to apply the controller proposed in [23]. Furthermore, the integral term of this controller that reduces the position tracking error is used to compensate for the cable tension wrench and will be shown efficient in the experiments presented in Section V. Therefore, we choose the reference velocity tracking errors to be:

$$\mathbf{e}_{v_f} = \mathbf{v}_f - \mathbf{v}_f^* \text{ and } e_{\dot{\psi}_{b_f}} = \dot{\psi}_{b_f} - \dot{\psi}_{b_f}^* \tag{8}$$

where $\mathbf{v}_f$ and $\dot{\psi}_{b_f}$ are the actual translational and yaw angular velocities of the drone provided by state estimation. However, the challenge is to generate the trajectory for the quadrotor outputs $(\mathbf{p}_f^*(t), \psi_{b_f}^*(t))$ by observing the shape of the cable with a camera, whose rate is 20 Hz taking into account the time for tracking the visual features $\mathbf{s}$ and computing the desired control velocities $\mathbf{v}_f^*$ and $\dot{\psi}_{b_f}^*$. A straightforward solution is to interpolate the outputs each time a new visual servoing command is computed. It can be done by using some differentiable curves, such as splines respecting the boundary conditions on the outputs and its derivatives to achieve $\mathbf{v}_f^*$ and $\dot{\psi}_{b_f}^*$ as fast as possible. These conditions are important since the stability of the visual servoing controller (4)-(5) holds under the assumption that the control velocities $\mathbf{v}_f^*$ and $\dot{\psi}_{b_f}^*$ can be executed by the drone. Fortunately, this solution already exists, as proposed in [24] where a quadrotor steering method is provided to reconstruct the outputs trajectories with their derivatives. This solution generates a trajectory feasible by the drone if the derivatives of the outputs are correctly bounded.

In practice, we implemented the control architecture presented in Fig. 2. Our control laws (4),(5) are computed by the "Shape Visual Servoing & Visibility Task" block and we use an inner and outer control loop strategy to deal with the slow rate of $\mathbf{v}_f^*$ and the fast rate needed for $\boldsymbol{\omega}_{prop_f}^*$. The control outputs of the visual servoing are sent to the trajectory generator [24] (KDTP Maneuver block in Fig. 2) and the Lee controller [23] (Lee ctrl block in Fig. 2) is in charge of computing the control velocities of the propellers which lead the drone to follow this trajectory. An unscented Kalman filter (UKF block) is used to estimate the current state of the drone (position and orientation) from measurements provided by an onboard IMU sensor and an external motion capture system (Optitrack block). The control velocities of the propellers are applied to the drone motors thanks to the Rotorcraft-Genom3 middleware [25]. Note that since the cable model with hanging load (and without any load) has been derived using the static equilibrium assumption, the quadrotor can only move in near hovering mode in order to not destabilise the cable dynamics, and therefore its outputs derivatives remain correctly bounded during all the shape visual servoing task.

IV. Visual Features Estimation

In this section, we present the image processing pipeline (Visual Feature Tracking block in Fig. 2) that we propose for estimating in real time the visual features $\mathbf{s}$ from the data provided by a RGB-D camera. The camera is mounted on the

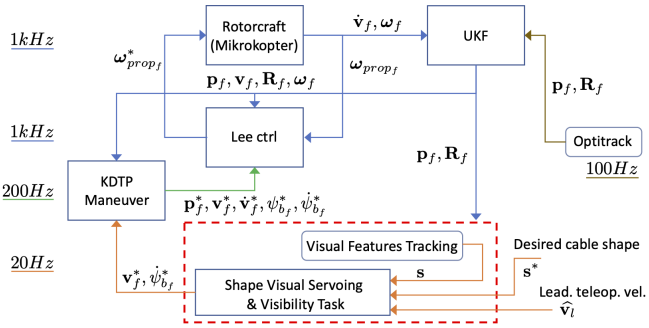


Fig. 2: Control architecture of the follower quadrotor. The visual feature tracking process and visual servoing blocks are enclosed in the red box. The "KDTP block" is in charge of updating at 200Hz the drone trajectory to follow the desired velocity emitted at 20Hz by our visual controller. The "Lee ctrl" block regulates at 1kHz the speeds of the propellers to perform the drone trajectory. The "Rotorcraft" module is the low-level controller of the drone that applies the desired propellers speeds to the motors and measures the drone acceleration and angular velocity thanks to an IMU. The "UKF" block is in charge of estimating the current state of the drone by fusing its internal measurement of acceleration and angular velocity with drone position and attitude measured at 100Hz by the "Optitrack" module.

follower drone to observe the cable (see Fig. 3). The method consists first in capturing the current point cloud and the RGB image provided by the RGB-D camera. A binarization is then applied on the RGB image based on a HSV threshold to segment the cable from its color (yellow in our case). The result of the binarization provides a mask that we apply on the point cloud to retain only the points of the cable. Then the PCL library [26] is used to downsample this segmented point cloud using a voxelized grid approach. For this, the VoxelGrid[1] algorithm from the PCL is employed to create a 3D voxel grid over the input point cloud data. The centroid of each voxel is then computed, and only its coordinates are used further. The constant transformation matrix between the camera frame $\mathcal{F}_c$ and the drone body frame $\mathcal{F}_{b_f}$ is determined off line with a classical hand-eye calibration method, which allows expressing the downsampled point cloud of the cable in $\mathcal{F}_{\mathrm{w}}$ at each iteration of the visual servoing controller thanks to the drone current pose. Then a RANSAC method (set with a maximum of 50 iterations to respect the real-time constraint) is used to estimate the cable plane that best contains all voxel centroids. This results in an estimate of its unitary normal ${}^{\mathrm{w}}\mathbf{n} = ({}^{\mathrm{w}}\mathbf{n}_x, {}^{\mathrm{w}}\mathbf{n}_y, {}^{\mathrm{w}}\mathbf{n}_z)$ and the set of inliers in the point cloud corresponding to the best-fit plane whose parameters ${}^{\mathrm{w}}\mathbf{n}$ and d are obtained by minimizing the following objective:

$$\begin{aligned} \min_{{}^{\mathrm{w}}\mathbf{n}, d} \quad & \textstyle\sum_i \left({}^{\mathrm{w}}\mathbf{n}^T \mathbf{x}_i + d\right)^2 \\ \text{s.t} \quad & \|{}^{\mathrm{w}}\mathbf{n}\| = 1 \end{aligned} \tag{9}$$

[1] https://pointclouds.org/documentation/tutorials/voxel_grid.html

where $\mathbf{x}_i = (x_i, y_i, z_i)$ are the coordinates of each voxel and d is the distance to the plane from the origin. In the case where the cable shape becomes a straight line, the minimization of (9) by the RANSAC algorithm results in a vertical plane. Since the resulted normal may have different signs pointing towards the cable plane or in the opposite direction, a Kalman filter could be used to reject these fluctuations, but this solution is highly dependent on the initialisation condition of the normal. Instead, we propose to exploit the property that the point cloud is sorted in a list by its z-coordinates and that the last point ${}^{\mathrm{w}}\mathbf{p}_e$ is the farthest point observed by the camera from the attachment point on the drone $\mathbf{b}_f$. The following relation gives a direction $\mathbf{d}_n$ for the cable in $\mathcal{F}_{\mathrm{w}}$ starting from its attachment point in $\mathbf{p}_f$:

$$\mathbf{d}_n = \frac{{}^{\mathrm{w}}\mathbf{p}_e - {}^{\mathrm{w}}\mathbf{p}_f}{||{}^{\mathrm{w}}\mathbf{p}_e - {}^{\mathrm{w}}\mathbf{p}_f||} \tag{10}$$

To ensure that the normal always points towards the cable we compute the cross-product:

$$\mathbf{c} = {}^{\mathrm{w}}\mathbf{n} \times \mathbf{d}_n \tag{11}$$

and check if the sign of its z-component $\mathbf{c}_z$ is positive. If $\mathbf{c}_z$ is negative, this means that the direction of the normal must be reversed in the opposite direction. In the next step, a noise reduction is performed by applying a standard Kalman filter on the values of the estimated cable plane parameters. A constant-velocity model has been considered for the evolution of these parameters, and we set the ratio of measurement variance to model variance at 30 to give greater confidence in the state model. After that, it is straightforward to find the yaw angle of the plane that corresponds to the feature:

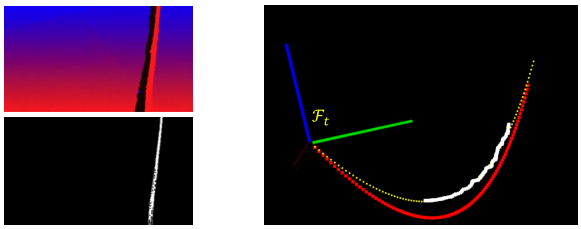
$$\alpha = \arctan \frac{{}^{\mathrm{w}}\mathbf{n}_y}{{}^{\mathrm{w}}\mathbf{n}_x} \tag{12}$$

and the roll angle ϕ of the plane that describes its deviation with respect to the gravity vector:

$$\phi = \arctan \frac{{}^{\mathrm{w}}\mathbf{n}_z}{\sqrt{{}^{\mathrm{w}}\mathbf{n}_x^2 + {}^{\mathrm{w}}\mathbf{n}_y^2}} \tag{13}$$

Finally, the visual features a and b are extracted by performing a robust least square fitting of the model parobola with the points corresponding to the intersection of the estimated plane with the segmented point cloud.

As mentioned previously, in practice, handling the cable with one or two drones has the effect of introducing a slight rolling motion of the cable plane due to the dynamics of the drones (the drone vibrates and needs to tilt slightly to change its position). The estimate ϕ given by (13) is thus used to compensate for this tilt directly in the control law (4) through $\mathbf{R}(\phi)$. This angle ϕ remains low in case of slow motion of the drones. On the contrary, when both drones move rapidly in the same direction, which is typical for transportation scenario, the cable plane rolls at a higher angle and is no more vertical. Note that ϕ is not controlled but it will naturally reduces when the drones reach a hovering configuration.



(a) (b)

Fig. 3: Extraction of the visual features using the onboard RGB-D camera. (a) Depth image on the top and segmented cable on the RGB image on the bottom. (b) 3D view of the extraction results: the white points correspond to the cable points used to estimate a, b, α and ϕ. The resulted fitted parabola is depicted in yellow. The red points represent an example of target parabola to reach with desired parameters $\mathbf{s}^* = (a^*, b^*, \alpha^*)$.

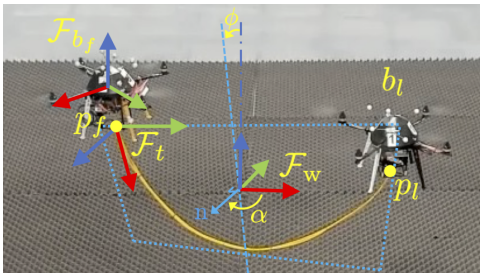


Fig. 4: Scenario 1: two drones holding a yellow cable. $\mathcal{F}_t$ stands for the cable frame, $\mathcal{F}_{b_f}$ for the body frame of the follower drone which is controlled by visual servoing and $\mathcal{F}_w$ for an arbitrary world frame. $\mathbf{p}_f$ is the cable attachment point on the follower drone and origin of $\mathcal{F}_t$, $\mathbf{p}_l$ is the attachment point on the teleoperated leader drone b_l. Dashed blue lines represent the cable plane with its yaw and roll angles α, ϕ and its normal $\mathbf{n}$.

V. Experimental Results

A. Experimental setup

A picture of the experimental setup is presented in Fig. 4 where the leader and follower quadrotor drones are handling the extremities of a flexible cable.

The hardware descriptions and software to build and control the quadrotors are provided by the *telekyb3* platform[2]. Each quadrotor is made of a *Mikrokopter*[3] autopilot board with Inertial Measurement Unit (IMU) and is connected to the electric motor speed controllers. This board is connected via a serial bus to a *NVIDIA Jetson TX2*[4] onboard computer, where the inner-loop (low-level) and outer-loop (visual servoing for the follower drone) control layers are running. The follower drone is equipped with an Intel *RealSense D435* depth camera with 848×480 resolution and 60 fps framerate. The state (pose) of each quadrotor is estimated using its IMU and an optical tracking system. The tracking system frame

[2] https://git.openrobots.org/projects/telekyb3
[3] https://www.mikrokopter.de
[4] https://developer.nvidia.com/embedded/jetson-tx2

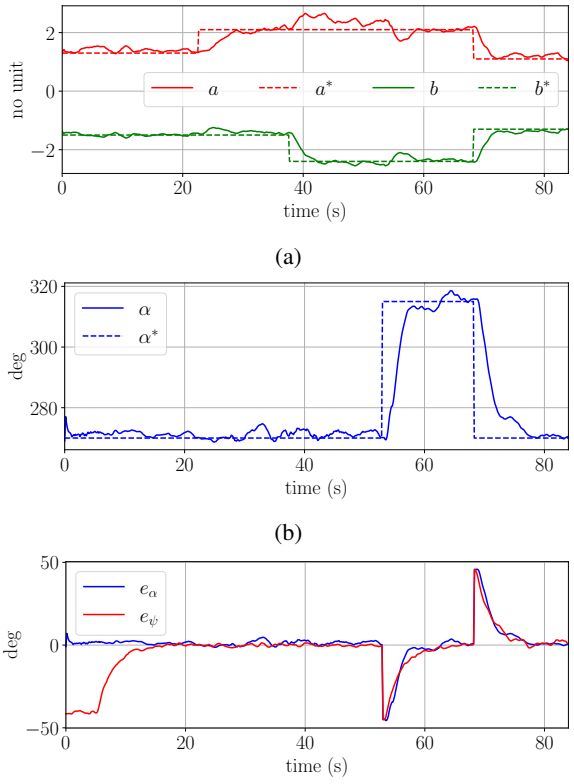


Fig. 5: Scenario 1: results of the cable shape visual servoing. (a) Desired and current parabola coefficients. (b) Desired and current yaw angle of the cable plane. (c) Error for α and ψ_{b_f} (visibility task).

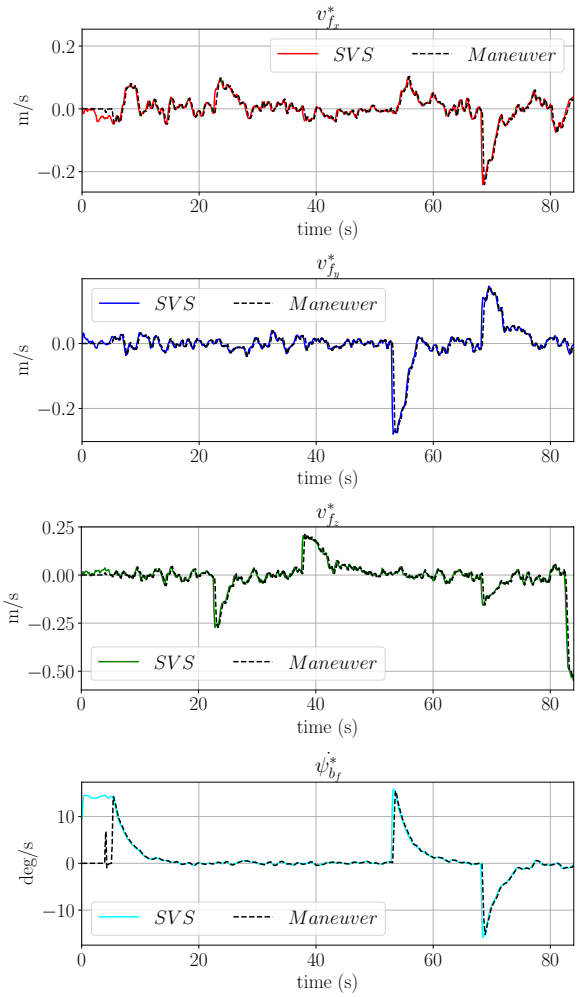


Fig. 6: Scenario 1: trajectories generated by the Maneuver module to track the shape visual servoing (SVS) control inputs.

is chosen as the world frame $\mathcal{F}_{\mathrm{w}}$, but it could be the frame of the onboard camera if this one was also used to perform visual odometry. The coordinates of the cable attachment point in the quadrotor body frame ${}^{b_f}\mathbf{p}_f = (0, 0, -0.13m)$ were measured manually. Note that the attachment point is not directly in the drone COG due to the presence of the battery and the *Jetson TX2* that balance the position of the COG closer to the barycenter of the drone. The cable is made of a rubber rope with a length chosen to be $L = 1.6$ m and diameter of 16 mm. In practice, a high camera frame rate and fast image processing are preferable for the visual servoing to be responsive. Therefore, processing the point cloud to extract the visual features and the plane angle ϕ becomes a challenging task for the embedded hardware on the small drone due to the processing delay between camera shot and tracking. To achieve real-time capability, a hardware acceleration of our image processing was implemented on the GPU of the *Jetson TX2* using the *CUDA* framework[5]. It allows us to track the visual features at a rate of 20 Hz.

B. Scenario 1: cable manipulation

In this scenario, we test the autonomous shaping task of the cable by the follower drone while the leader drone is maintained in a hovering mode. The results of this experiment are also presented in the accompanying video. The initial shape of the cable is such that $\mathbf{s}_0 = (1.3, -1.5, 270^o)$, the initial yaw angle involved in the cable visibility task is $\psi_{b_{f_0}} = 228^o$ and we set a sequence of target shapes (after the visual servoing starts at time $t = 5$ s) to reach successively: $\mathbf{s}_1^* = (2.1, -1.5, 270^o)$, $\mathbf{s}_2^* = (2.1, -2.4, 270^o)$, $\mathbf{s}_3^* = (2.1, -2.4, 315^o)$ and $\mathbf{s}_4^* = (1.1, -1.3, 270^o)$. The shape visual controller was set with control gain $\lambda = 0.25$, integral gain $\mu = 0.025$ and window size $N = 10$. Fig. 5(a)-(b) shows the convergence of the visual features $\mathbf{s}$ to their successive desired values. The evolution of the error $e_\alpha = \alpha - \alpha^*$ related to the cable yaw angle and the error of the visibility task $e_\psi = \psi_{b_f} - \alpha^*$ are presented in Fig. 5(c). We can note the convergence of the visibility task once the visual servoing starts. Then the visibility task is correctly maintained and reacts to the change of α^*. As shown in Table I, all visual features and the error of the visibility task converge to their desired values with fast convergence times relative to the large displacement the follower drone has

[5]https://docs.nvidia.com/cuda/doc/index.html

Targ.	t_{e_a}	$\overline{e_a}$	t_{e_b}	$\overline{e_b}$	t_{e_α}	$\overline{e_\alpha}$	t_{e_ψ}	$\overline{e_\psi}$
$\mathbf{s}_1^*$	6.03	0.098	-	-	-	-	-	-
$\mathbf{s}_2^*$	-	-	4.30	0.081	-	-	-	-
$\mathbf{s}_3^*$	-	-	-	-	4.62	1.86	8.81	0.83
$\mathbf{s}_4^*$	3.39	0.10	3.89	0.086	8.42	0.75	8.39	0.87

TABLE I: Convergence time in seconds t_{e_a}, t_{e_b}, t_{e_α}, t_{e_ψ} of each visual feature and the visibility task error for the 4 cable shape targets in Scenario 1. The convergence time is measured when the error falls below 5% of its initial value. The values $\overline{e_a}$ (no unit), $\overline{e_b}$ (no unit), $\overline{e_\alpha}$ (deg), $\overline{e_\psi}$ (deg) are the means of the absolute errors obtained after convergence.

to achieve. Furthermore, the low values of the mean errors obtained after convergence demonstrate the accuracy of the cable shaping task. As for the parameters a and b, we can see the complete rejection of the steady-state error but also the appearance of damped oscillations (see Fig. 5(a)), which influences the reference trajectory of the drone outputs shown in Fig. 6. Note that in this last figure, it is hard to notice any difference between the shape visual servoing (SVS) control inputs and the outputs generated by the Maneuver trajectory generator once the shape visual servoing task was started at time $t = 5$ s. This means that interpolation by splines is efficient. All these results demonstrate the efficiency and robustness of our cable shape visual servoing approach.

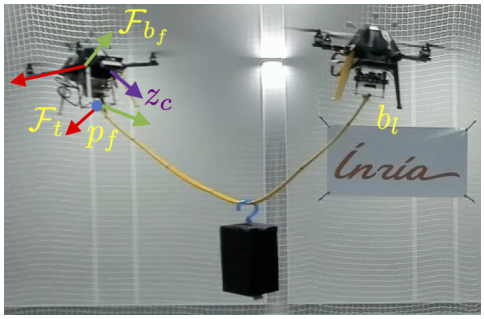


Fig. 7: Scenario 2: transportation of a box. The leader drone is b_l and the follower drone is b_f. The follower body frame and cable frame are illustrated together with the on-board camera optical axis $\mathbf{z}_c$.

C. Scenario 2: leader following, box grasping and transportation

In this second scenario, we first proceed as previously, but while the leader drone is teleoperated, and, in a second phase, we propose to manipulate the cable in order to pick up a box lying on the floor and transport it to another location. The results of this experiment are also presented in the accompanying video. For this experiment, the control law parameters of the follower drone were set as in the first scenario. At the beginning of the experiment, a box weighing $0.3kg$ is placed in a container attached to the floor with duct tape to prevent the box from blowing away by the air flow of the propellers. After showing that our control scheme is able to compensate for the motions of the leader, the system is used to grasp and manipulate this box (see Fig. 7). The follower drone b_f autonomously adjusts the shape of the cable to maintain a desired shape $\mathbf{s}^*$ while the leader drone is teleoperated by the user. The task consists of several actions, the first of which is the launch of the two drones, and the second the activation of the shape visual servoing controller at time $t = 0$ s that makes the follower drone b_f autonomous to maintain its initial shape $\mathbf{s}_0^* = (1.3, -1.5, 270^o)$. At time $t = 8$ s, a first arbitrary reference step is applied on the desired features with $\mathbf{s}_1^* = (2.1, -2.4, 315^o)$ and at time $t = 15$ s, the leader drone b_l is then teleoperated via a joystick to see how the system reaches and maintains the current desired cable shape while the leader drone is moving. Then a new shape reference is applied at time $t = 36$ s with $\mathbf{s}_2^* = (1.0, -1.3, 270^o)$ and the leader drone b_l is teleoperated towards the box with a hook attached on its upper side. When the cable passes under the hook (around time $t = 63$ s), a vertical velocity is sent to the drone b_l to fly up, therefore lifting the box. Finally, the release of the box is performed in another place of the room (around time $t = 97$ s). As can be seen in Fig. 8, the shape of the cable is well maintained. We can observe at different times dumped oscillations that are due to perturbations induced by aggressive control velocities (plotted in Fig. 8(d)) of the teleoperated drone. Anyway, the shape controller deals efficiently with the transportation task as demonstrated by the convergence of the visual features even in the presence of a point load on the cable that increases the tension in the cable. Moreover, this experiment showed the robustness of our shape visual control scheme to unmodelled disturbances that occur during grasping and lifting the box.

VI. CONCLUSIONS

In this work, we showed that it is possible to manipulate a flexible cable using a pair of drones. The experimental results demonstrate the validity of our visual servoing approach to deform the cable to some desired shapes while compensating for its tension by an integral action. Furthermore, experiments reveal the robustness of our shape visual servoing approach to uncertainties such as drone state estimation error and depth measurement noise of the RGB-D camera. Finally, we experimentally demonstrate that our method can be used in a leader-follower control strategy for autonomously maintaining the shape of the cable and its visibility during the grasping, transportation and release of an object by two drones.

ACKNOWLEDGMENTS

The authors would like to thank Fabien Spindler for his invaluable help in the achievement of the experimental results presented in this paper.

REFERENCES

[1] M. Malveiro and J. Cordeiro, "Overhead power line stringing with UAV," in *12th Int. Conf. on Live Maintenance*, 2017, pp. 1–2.

[2] K. Pawlak and D. Serek, "High voltage transmission line stringing operation. usage of unmanned aerial vehicles for installation of conductor and grounding wires with optical fibers," in *15th Int. Conf. on Electrical Machines, Drives and Power Systems*, 2017, pp. 32–37.

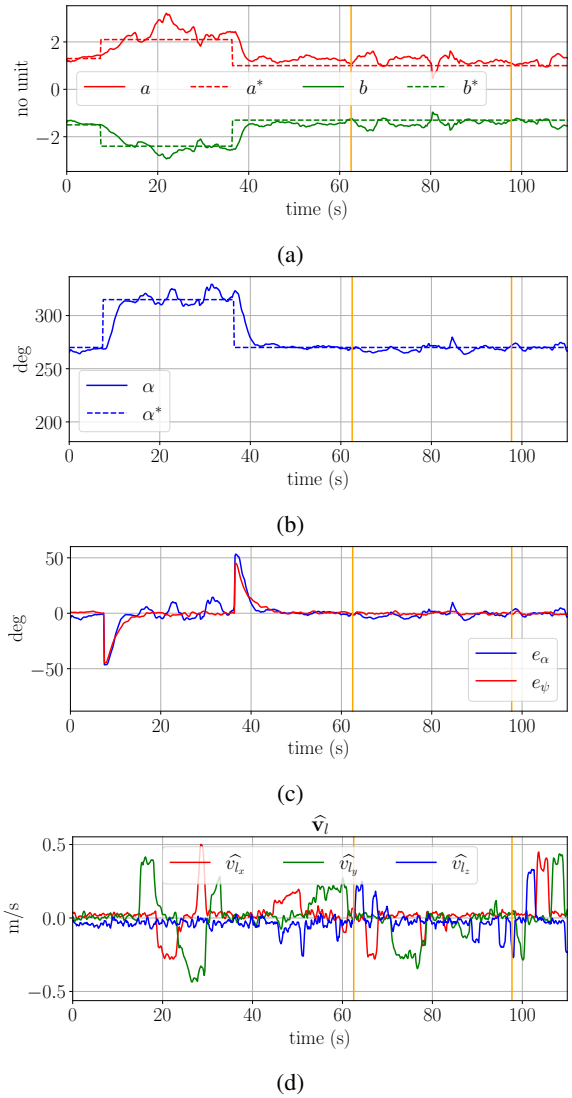


Fig. 8: Scenario 2: results of the cable shape visual servoing of the follower drone during leader following, box grasping, transportation and release. (a) Desired and current parabola coefficients with the grasp, transportation and release of an object that occur within the time period indicated by the orange box. (b) Desired and current yaw angle of the cable plane. (c) Error for α and ψ_{b_f} (visibility task) (d) Feedforward term used during the control of the follower drone b_f that corresponds to the teleoperated translational velocity $\widehat{\mathbf{v}}_l$ of the leader drone b_l.

[3] S. Kiribayashi, K. Yakushigawa, and K. Nagatani, "Position estimation of tethered micro unmanned aerial vehicle by observing the slack tether," *IEEE Int. Symp. on Safety, Security, and Rescue Robotics*, pp. 159–165, 2017.

[4] B. Madani and M. Ndiaye, "Hybrid truck-drone delivery systems: A systematic literature review," *IEEE Access*, vol. 10, pp. 92 854–92 878, 2022.

[5] G. Yu, W. Xie, D. Cabecinhas, R. Cunha, and C. Silvestre, "Adaptive control with unknown mass estimation for a quadrotor-slung-load system," *Int. Society of Automation Transactions*, vol. 133, pp. 412–423, 2023.

[6] J. Erskine, A. Chriette, and S. Caro, "Wrench capability analysis of aerial cable towed systems," in *ASME Int. Design Engineering Technical Conf. and Computers and Information in Engineering Conf.*, 2018.

[7] D. Sanalitro, M. Tognon, A. J. Cano, J. Cortés, and A. Franchi, "Indirect force control of a cable-suspended aerial multi-robot manipulator," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6726–6733, 2022.

[8] M. Tognon, C. Gabellieri, L. Pallottino, and A. Franchi, "Aerial Co-Manipulation with Cables: The Role of Internal Force for Equilibria, Stability, and Passivity," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2577 – 2583, 2018.

[9] A. E. Jiménez-Cano, D. Sanalitro, M. Tognon, A. Franchi, and J. Cortés, "Precise Cable-Suspended Pick-and-Place with an Aerial Multi-robot System," *Journal of Intelligent and Robotic Systems*, vol. 105, no. 3, p. 68, 2022.

[10] M. Tognon, H. A. T. Chávez, E. Gasparin, Q. Sablé, D. Bicego, A. Mallet, M. Lany, G. Santi, B. Revaz, J. Cortés, and A. Franchi, "A truly-redundant aerial manipulator system with application to push-and-slide inspection in industrial plants," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1846–1851, 2019.

[11] K. Bodie, M. Tognon, and R. Siegwart, "Dynamic end effector tracking with an omnidirectional parallel aerial manipulator," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8165–8172, 2021.

[12] T. W. Danko, K. P. Chaney, and P. Y. Oh, "A parallel manipulator for mobile manipulating uavs," in *IEEE Int. Conf. on Technologies for Practical Robot Applications*, 2015, pp. 1–6.

[13] S. Abiko, A. Kuno, and S. Narasaki, "Obstacle avoidance flight and shape estimation using catenary curve for manipulation of a cable hanged by aerial robots," in *Int. Conf. on Robotics and Biomimetics*, 2017, p. 2099–2104.

[14] D. S. D'Antonio, G. A. Cardona, and D. Saldana, "The catenary robot: Design and control of a cable propelled by two quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3857–3863, 2021.

[15] D. S. D'Antonio and D. Saldana, "Folding knots using a team of aerial robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2022, pp. 3372–3377.

[16] M. Cao, K. Cao, S. Yuan, T.-M. Nguyen, and L. Xie, "Neptune: Nonentangling trajectory planning for multiple tethered unmanned vehicles," *IEEE Trans. on Robotics*, vol. 39, no. 4, pp. 2786–2804, 2023.

[17] P. Kotaru and K. Sreenath, "Multiple quadrotors carrying a flexible hose: dynamics, differential flatness and control," in *IFAC PapersOnLine*, vol. 53, no. 2, 2020, p. 8832–8839.

[18] J. Estevez, J. M. Lopez-Guede, G. Garate, and M. Grana, "Hybrid modeling of deformable linear objects for their cooperative transportation by teams of quadrotors," *Applied Sciences*, vol. 12, no. 10, p. 5253, 2022.

[19] C. Gabellieri and A. Franchi, "Differential flatness and manipulation of elasto-flexible cables carried by aerial robots in a possibly viscous environment," in *Int. Conf. on Unmanned Aircraft Systems*, 2023, p. 963–968.

[20] L. Smolentsev, A. Krupa, and F. Chaumette, "Shape visual servoing of a tether cable from parabolic features," in *IEEE Int. Conf. on Robotics and Automation*, 2023, pp. 734–740.

[21] R. Spica, P. Robuffo Giordano, M. Ryll, H. H. Bülthoff, and A. Franchi, "An open-source hardware/software architecture for quadrotor uavs," in *2nd IFAC Workshop on Research, Education and Development of Unmanned Aerial Systems*, vol. 46, no. 30, 2013, pp. 198–205.

[22] A. Srour, A. Franchi, and P. Robuffo Giordano, "Controller and Trajectory Optimization for a Quadrotor UAV with Parametric Uncertainty," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Detroit, 2023, pp. 1–7.

[23] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," in *49th IEEE Conf. on Decision and Control*, 2010, pp. 5420–5425.

[24] A. Boeuf, J. Cortés, R. Alami, and T. Siméon, "Enhancing sampling-based kinodynamic motion planning for quadrotors," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 2447–2452.

[25] M. Foughali, F. Ingrand, and A. Mallet, "Genom3 templates: from middleware independence to formal models synthesis," *arXiv preprint arXiv:1807.10154*, 2018.

[26] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE Int. Conf. on Robotics and Automation*, Shanghai, 2011.