

Towards Vision-Based Impedance Control for the Contact Inspection of Unknown Generically-Shaped Surfaces with a Fully-Actuated UAV

Ramy Rashad, Davide Bicego, Ran Jiao, Santiago Sanchez-Escalonilla, and Stefano Stramigioli

Abstract—The integration of computer vision techniques for the accomplishment of autonomous interaction tasks represents a challenging research direction in the context of aerial robotics. In this paper, we consider the problem of contact-based inspection of a textured target of unknown geometry and pose. Exploiting state of the art techniques in computer graphics, tuned and improved for the task at hand, we designed a framework for the projection of a desired trajectory for the robot end-effector on a generically-shaped surface to be inspected. Combining these results with previous work on energy-based interaction control, we are laying the basis of what we call *vision-based impedance control paradigm*. To demonstrate the feasibility and the effectiveness of our methodology, we present the results of both realistic ROS/Gazebo simulations and preliminary experiments with a fully-actuated hexarotor interacting with heterogeneous curved surfaces whose geometric description is not available a priori, provided that enough visual features on the target are naturally or artificially available to allow the integration of localization and mapping algorithms.

Keywords: Vision-Based Impedance Control, Aerial Physical Interaction, Contact-Based Inspection, Aerial Robotics.

I. INTRODUCTION

The flourishing research on aerial robotics conducted in the last decade has progressively matured an increasing interest towards the study of unmanned aerial vehicles (UAVs) actively interacting with the surrounding environment [1], giving birth to the novel branch of aerial physical interaction, often also referred to as aerial manipulation. Apart from the scientific desire of solving new challenging theoretical problems, a relevant explanation for the springing of this new research focus is to be found in the exponentially growing demand in the market for aerial robots capable of accomplishing operations which are beyond typical passive visual inspection tasks, since they involve the exchange of energy between the robots and the environment. Examples of such operations include the contact-based inspection and maintenance of sensible sites and infrastructures, the transportation, assembling and decommissioning of structures, and tele-manipulation tasks with haptic feedback. Considered

R. Rashad, D. Bicego, and S. Sanchez-Escalonilla are with the Robotics and Mechatronics group, University of Twente, Enschede, The Netherlands. Emails: r.a.m.rashadhashem@utwente.nl, d.bicego@utwente.nl, s.sanchezescalonilla@plaza.utwente.nl

R. Jiao is with the School of Mechanical Engineering and Automation, Beihang University, Beijing, China and the Robotics and Mechatronics group, University of Twente, Enschede, The Netherlands. Email: jiaoran@buaa.edu.cn

S. Stramigioli is with the Robotics and Mechatronics group, University of Twente, and ITMO University, Saint Petersburg, Russia. Email: s.stramigioli@utwente.nl

This work has been funded by the cooperation program “INTERREG Deutschland-Nederland” as part of the SPECTORS project number 143081.

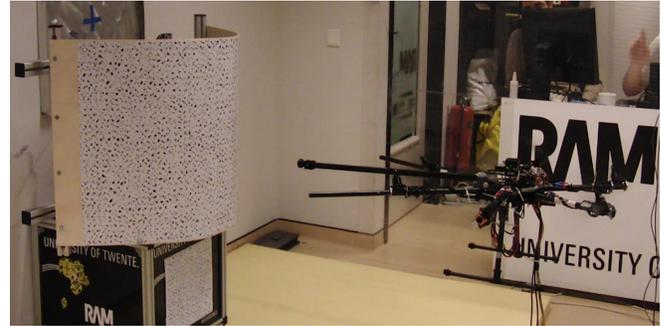


Fig. 1: Photo of our fully-actuated aerial robot scanning an unknown curved surface prior to interaction.

the relevance of such applications, many international synergies aimed to advance in this topic have been established. In the scope of the European Union, this gave birth to several collaborative projects like, e.g., Aeroarms¹, Aeroworks², Sectors³, and Aerial-core⁴, just to mention some.

In the view of endowing aerial robots with the manipulation capability and the dexterity needed to fulfill the particular interaction task at hand, different directions have been explored, which can be classified in two main approaches. The first one foresees to cope with the *underactuation* of typical UAVs with collinear rotors (like the well-known quadrotors, which can exert forces only along one direction w.r.t. the body frame), i.e., their inability to be commanded to follow arbitrary trajectories in their full (6D) configurations space, by equipping them with articulated robotic arms and exploiting the increased number of degrees of freedom (DoFs) to accomplish the task at hand. Paradigms of this kind have been fostered, e.g., in [2]–[4]. Depending on the number of the additional DoFs, the load may be manipulated independently from the motion of the platform. However, it is not possible for the robot to rapidly exert forces in arbitrarily-chosen directions of the space while keeping a pre-specified decoupled orientation.

On the other hand, the philosophy of the second paradigm, established in the more recent years, is to act on the actuator arrangement of the aerial platform rather than (exclusively) on its arm. By tilting the propellers from the classical collinear disposition in particular ways, it is possible to decouple, at least partially, the rotational dynamics from the

¹<http://www.aeroarms-project.eu>

²<http://www.aeroworks2020.eu/>

³<https://spectors.eu/>

⁴<https://aerial-core.eu/>

translational one, i.e., the robot gets the means to translate without the need to re-orient itself or, in a complimentary way, to hover while maintaining different non-flat orientations. This is achieved at the cost of an increased energy consumption, spent to generate the required internal forces to produce that motion. Systems of this kind are traditionally referred to as *fully-actuated*, and proved to be much more effective for physical interaction tasks compared to under-actuated ones, thanks to their improved dexterity and the partial decoupling of the generated forces and moments [5]–[8]. As a consequence, mechanical over-complications and additional weight in the arm design can be avoided and the interaction can take place also with simple rigid tools, following the *flying end-effector paradigm* fostered in [6].

A necessary step for an autonomous interactive aerial robot is to be able to perceive the environment it will interact with. Achieving consistent and repeatable interaction behavior requires a prior knowledge of the environment’s geometric properties, which is not suitable for autonomous robots in unstructured environments. A solution for that is combining computer-vision with interaction control techniques.

The topic of vision-based interaction control is already known in the literature of industrial ground manipulators and its study has led to the accomplishment of a wide spectrum of interesting tasks. In the view of automatizing labor-intensive works, the authors in [9] proposed a combination of vision-based and impedance control to perform polishing of flat surfaces like, e.g., walls. A similar approach has been fostered also in [10], where the authors achieved the tracking of a desired force on a non-planar surface of unknown geometry. Following this line of thoughts, a semi-autonomous robotic pen-drawing system has been presented in [11].

On the other hand, despite the compelling breakthroughs recently delivered by the research related to computer vision and state estimation (see [12] for a broad overview) considerable effort has still to be made to integrate these techniques with interaction control in the field of aerial robotics. As a matter of fact, impressive experimental results of aerial robots interacting with the environment have been demonstrated in [5], [6], [13], [14], which however rely on partial a priori knowledge of the interaction target. Towards the direction of vision-aided interaction, the authors in [8] exploit depth servoing to locally reconstruct the normal directions of a ceiling vault and use an impedance-controlled omni-directional UAV to perform punctual interaction tasks.

Driven by the aim to exploit vision algorithms in more global way w.r.t. the object and to extend the nice results presented in [8], in this paper we outline a novel vision-based interaction planner which allows to design a complete desired trajectory for the robot end-effector on the surface of unknown objects endowed, naturally or artificially, with features. Thanks to the successful integration of such method with a geometric impedance controller [7], we present numerical and preliminary experimental results of a fully-actuated UAV interacting with arbitrarily-shaped surfaces.

In this paper, we consider the challenging task of inspecting by contact the surface of an unknown physical

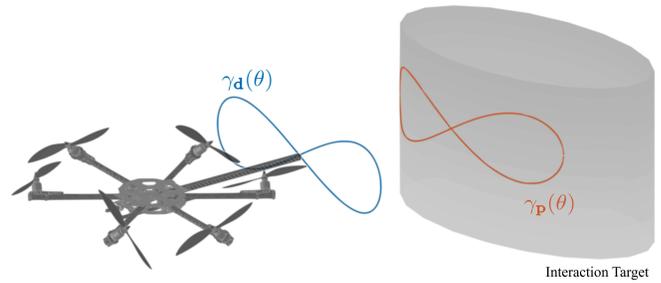


Fig. 2: Illustration of the employed curve projection idea, showing the desired curve $\gamma_d(\theta)$ in blue, and the projected curve $\gamma_p(\theta)$ in orange, where θ is the curve parameter.

object by means of a fully-actuated aerial robot, a hexarotor in the specific case, endowed with a simple rigid tool. The accomplishment of this operation represents a proof of concept towards the tackling of a wide set of meaningful use-case applications like, e.g., the remote detection of cracks on a gas pipeline or on the blade of a wind-mill, referring to an industrial scenario, or the autonomous cleaning/painting of tall buildings, in an civil context. Differently from other approaches in the field of aerial interaction, we do not require any a priori model of the object, apart from an initial guess of its position and bounding box. More specifically, the contributions presented in this paper are the following:

- the integration of a computer graphics algorithm for the projection of any 2D desired path specified by the user onto an unknown generic interaction surface without the need to analytically reconstruct it, and of an energy-based impedance controller for exerting a force;
- realistic numerical simulations in ROS/Gazebo, showing the effectiveness of our approach in different scenarios;
- preliminary experimental tests with a fully-actuated UAV interacting with planar and curved surfaces.

The paper outline is structured as follows: In Sec. II we provide the details of the vision based interaction methodology. To validate our approach, we present the results of both numerical simulations and real experiments in Sec. III and in Sec. IV, respectively. A brief description of the fully-actuated aerial robot used in the experimental validation is included in Sec. IV. Finally, in Sec. VI, we outline the conclusion and provide hints for future work.

II. VISION BASED INTERACTION METHOD

A. Problem Formulation

In this work, we consider the problem of an aerial robot physically interacting with a single target of unknown pose and geometry, as shown in Figs. 1–2. The interaction task of the aerial robot consists in sliding over the object surface. The task is defined in terms of a desired curve and wrench profile to be tracked by the UAV along the target surface. With reference to Fig. 2, this desired path is a parameterized curve γ_d defined in the end-effector’s frame. The goal is then to project this desired curve onto the target’s surface. The interaction planner then commands the UAV’s end-effector

to the projected curve γ_p on the surface and aligns the end-effector axis with the surface normal.

With the assumption that the target surface is endowed with enough visual features, we use images from an on-board camera to reconstruct the object, instead of relying on an a priori computer-aided design (CAD) model. To recover information about the 3D structure of the target, one could use simultaneous localization and mapping (SLAM)-algorithms to approximately gain a spatial representation of the target as well as an estimate of the UAV's pose relative to it. We choose as SLAM algorithm ORB-SLAM2 [15] due to its robust properties identified in the comparative study [16], and its support for stereo and RGB-D cameras. The ORB-SLAM2 algorithm extracts visual features (referred to as ORBs) from the camera images and locates their relative position with respect to the estimated camera pose. The output of ORB-SLAM2 is an estimated pose of the camera and a point-cloud, where each point corresponds to the ORB features identified on the surface of the target object.

The key problem now is projecting the desired curve onto the point-sampled surface of the target object, in addition to estimating the normal directions to the surface.

B. Proposed Method

The problem of projecting a point/curve onto a parametric or implicit surface is a well studied problem in the computer-graphics community. Using the point-cloud generated by ORB-SLAM2, one possible solution is to reconstruct the surface from the sampled points and then use one of the traditional techniques to project the desired curve onto the reconstructed surface and estimate its normals. However, there are several limitations for the implementation of the aforementioned solution on point clouds generated visually on an aerial robot. First, the errors introduced due to the surface reconstruction methods of the noisy point cloud will affect the projected curve. Second, traditional surface reconstruction methods often fail for large and complex point clouds. Moreover, it requires high computational power and memory space which are usually scarce in small-scale UAVs.

An alternative solution is to project the curve directly to the point cloud without reconstructing the surface. In fact, point-projection algorithms onto point clouds are used as a lower-level modules in some surface reconstruction algorithms [17]. We opt for this curve-projection method in our architecture, and use for that purpose the directed projection (DP) algorithm proposed by [18] with a slight modification for increasing computational efficiency for the considered interaction application, cf. Sec. III-A.

The overall interaction planning architecture is depicted in Fig. 3. The desired curve $\gamma_d(\theta) \in \mathbb{R}^2$ and desired normal force profile $f_d(\theta) \in \mathbb{R}$ are specified by the user, where $\theta \in [a, b]$ is the curve parameter for some $a, b \in \mathbb{R}$. The user also is required to specify the normal direction $\hat{n} \in S^2$ for which the curve is to be projected. Let $\mathbb{C}_N = \{\xi_i \in \mathbb{R}^3 | i \in \{1, \dots, N\}\}$ denote the set of unorganized data points representing the sampling of the target object's surface

generated through ORB-SLAM2. We suppose the point cloud \mathbb{C}_N to have a non-uniform distribution with possibly noise.

The curve γ_d is projected by discretizing it to a sequence of points N_c such that $\{\xi_{des,k}\} = \{\gamma_d(a), \dots, \gamma_d(b)\}$, for $k \in \{1, \dots, N_c\}$. The DP algorithm is then used to find the projection point, referred to as the footpoint, of each $\xi_{des,k}$ along the known projection direction \hat{n} onto the point-sampled surface encoded in the point cloud \mathbb{C}_N .

C. Curve Projection Algorithm

In the following, we present a summary of the directed projection algorithm used for finding the footpoint of a sampled number of points of the desired curve γ_d . Consider the test point ξ_{des} with an associated projection direction \hat{n} . Each point $\xi_i \in \mathbb{C}_N$ is associated with a positive weight α_i that takes into account both the distance between ξ_i and ξ_{des} as well as the deviation of the point ξ from the projection direction \hat{n} . The corresponding footpoint ξ_{proj} of ξ_{des} is expressed as

$$\xi_{proj} = \xi_{proj}(\tilde{t}) = \xi_{des} + \tilde{t}\hat{n}, \quad (1)$$

where the parameter $\tilde{t} \in \mathbb{R}$ is chosen to minimize the weighted sum

$$E(\tilde{t}) = \sum_{i=1}^N \alpha_i \|\xi_{proj}(\tilde{t}) - \xi_i\|^2. \quad (2)$$

The optimal parameter \tilde{t}^* is given by [18]

$$\tilde{t}^* = \frac{\beta - \xi_{des}^\top \hat{n}}{\|\hat{n}\|^2}, \quad \beta = \frac{c^\top \hat{n}}{c_0}, \quad (3)$$

where

$$c_0 = \sum_{i=1}^N \alpha_i \in \mathbb{R}^+, \quad c = \sum_{i=1}^N \alpha_i \xi_i \in \mathbb{R}^3. \quad (4)$$

The choice of the weights α_i affects significantly the performance of the projection algorithm. We used the same weight function suggested by [18], i.e.

$$\alpha_i = \frac{1}{1 + \|e_i\|^2 \|e_i \wedge \hat{n}\|^2}, \quad e_i = \xi_i - \xi_{des} \in \mathbb{R}^3, \quad (5)$$

with \wedge denoting the vector (cross) product in \mathbb{R}^3 . Intuitively, the weight function (5) is maximum ($\alpha_i = 1$) if ξ_{des} is on the point cloud ($e_i = 0$) or when ξ_i lies onto the projection axis \hat{n} ($e_i \wedge \hat{n} = 0$).

The projection algorithm is achieved through an iterative procedure as described in Algorithm 1. In the original DP algorithm in [18], a working point cloud \mathbb{C}_k is constructed (which is initially equal to \mathbb{C}_N), and gradually reduced by removing from it points that have low weights. However, in our case we initialize \mathbb{C}_k in a different way, discussed in Sec. III-A. After the weights $\{\alpha_i\}$ are calculated for points in \mathbb{C}_k , the mean (α_{mean}) and maximum (α_{max}) weight of the point cloud is computed. The points that are removed in

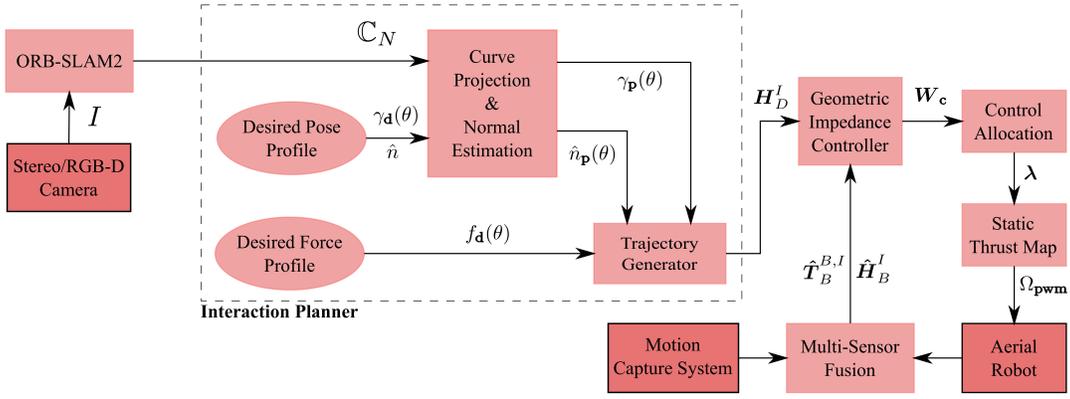


Fig. 3: Block diagram of our vision-based impedance framework. Each camera image I is used by ORB-SLAM2 to produce a point cloud \mathbb{C}_N in space. The interaction planner uses the point cloud together with the projection direction \hat{n} and the desired pose profile γ_d to compute the projected curve γ_p and the surface normals \hat{n}_p along the curve. The output of the interaction planner is the desired configuration H_D^I .

the j -th iteration are the ones with corresponding weights less than α_{lim} which is calculated as [18]

$$\alpha_{lim} = \begin{cases} \alpha_{mean} + \frac{\alpha_{max} - \alpha_{mean}}{10 - j} & j < 9, \\ \alpha_{mean} + \frac{\alpha_{max} - \alpha_{mean}}{2} & \text{otherwise.} \end{cases} \quad (6)$$

The algorithm iterates until the distance between the current projection estimation ξ_{proj} and the current test point ξ_{test} is less than a threshold ϵ or $\alpha_{max} = 1$. In the opposite case, the test point is moved to the current estimation ξ_{proj} and the new iteration starts with the smaller working point cloud with weights smaller than α_{lim} . When the algorithm terminates, it returns the footpoint ξ_{proj} of the desired point $\xi_{des,k}$. The process can be repeated for each point of the desired curve to be projected.

The surface normals are estimated using the work of [19] by estimating the normal of the tangent plane on the surface. The plane is constructed by least-square fitting of a specified number of cloud points near $\xi_{des,k}$. Note that in our framework only the normals along γ_p are relevant which eliminates the need to compute the normals of the whole point cloud.

D. Impedance Controller and Control Allocation

After the projected curve $\gamma_p(\theta) \in \mathbb{R}^3$ and the surface normal directions along the curve $\hat{n}_p(\theta) \in S^2$ are calculated, the desired pose and wrench trajectories are computed as follows: The user specifies a function $\theta(t)$ that is used to convert the projected curve and normals to time-trajectories. The end-effector's desired position $\xi_D^I(t) \in \mathbb{R}^3$ is calculated by

$$\xi_D^I(t) = \gamma_p(t) - \delta_{off}(t)\hat{n}_p(t), \quad (7)$$

where $\delta_{off} \in \mathbb{R}^+$ is a positive offset specified into the target object to allow for a compliant force exertion.

The end-effector's desired orientation is denoted by $R_D^I(t) \in SO(3)$. By choosing the \hat{x}_D axis of R_D^I to be aligned with the surface normal, a degree of freedom is

Algorithm 1: Point Projection Algorithm onto Point Cloud

```

1 Get point cloud  $\mathbb{C}_N$ , projection direction  $\hat{n}$ , selection cylinder
  radius  $\rho_{cyl}$ , and desired curve point  $\xi_{des,k}$ 
2 for  $i = 1, \dots, N$  do
3   Select  $\xi_i$  that lies within the selection cylinder
4   Build working point cloud  $\mathbb{C}_k$  with  $N_k$  points
5  $\xi_{test} = \xi_{des,k}$ 
6 for  $j = 1, \dots, j_{max}$  do
7   for  $i = 1, \dots, N_k$  do
8     Calculate  $\alpha_i$  by (5)
9   Calculate  $\xi_{proj}$  by (1,3,4)
10  Calculate  $\alpha_{max}$ ,  $\alpha_{mean}$ , and  $\alpha_{lim}$  (6)
11  if  $\|\xi_{test} - \xi_{proj}\| \leq \epsilon$  or  $\alpha_{max} = 1$  then
12    return
13  for  $i = 1, \dots, N_k$  do
14    if  $\alpha_i < \alpha_{lim}$  then
15      remove  $\xi_i$  from  $\mathbb{C}_k$ 
16   $\xi_{test} = \xi_{proj}$ 

```

remaining to be specified by the user in choosing a desired roll angle ϕ_{des} . Then we can compute $R_D^I(t)$ by

$$\begin{aligned} R_D^I &= (\hat{x}_D, \hat{z}_D \wedge \hat{x}_D, \hat{z}_D), \\ \hat{x}_D(t) &= -\hat{n}_p(t), \\ \hat{z}_D(t) &= \frac{\hat{x}_D(t) \wedge \hat{y}_D(t)}{\|\hat{x}_D(t) \wedge \hat{y}_D(t)\|}, \\ \hat{y}_D(t) &= (0, \cos(\phi_{des}(t)), \sin(\phi_{des}(t)))^\top. \end{aligned}$$

The methodology used in this work for the UAV to interact with the environment is geometric impedance control on $SE(3)$, which was derived in [20] using the port-Hamiltonian formalism. In what follows, we provide a summary of the controller.

The impedance control law W_c consists of the sum of an energy shaping wrench W_{es} and a damping injection wrench W_{di} . The control wrench component W_{di} is added to implement a virtual linear damper connected between the UAV's frame Ψ_B and the inertial frame Ψ_I such that

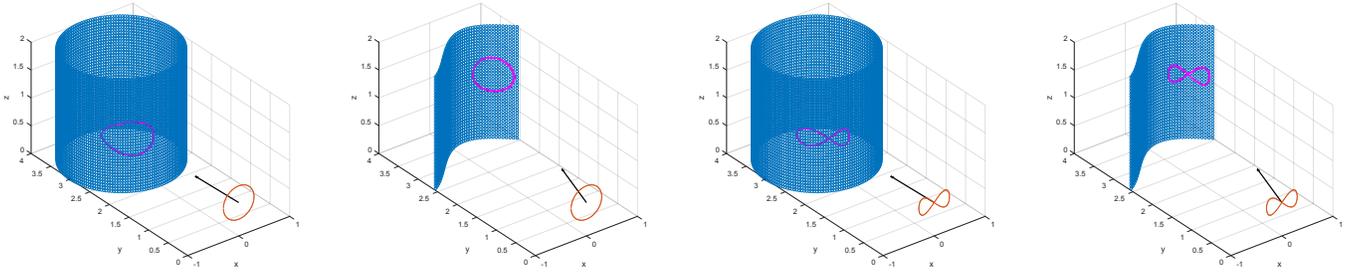


Fig. 4: Projection algorithm tested with different trajectories and heterogeneous surfaces.

energy of the closed-loop system is damped in free-flight. The role of \mathbf{W}_{es} is to add energy to the UAV to compensate the gravitational potential energy and to implement a virtual spatial nonlinear spring connected between the end-effector frame Ψ_E and the virtual desired frame Ψ_D . For small deviations between Ψ_E and Ψ_D , the spring control law behaves as a linear spring, decoupled in its axes. Therefore, if $K_{t,x}$ denotes the translational stiffness gain along the \hat{x}_D axis, choosing the offset δ_{off} by

$$\delta_{off}(t) = \frac{f_d(t)}{K_{t,x}}, \quad (8)$$

could (ideally) change the force exerted by the UAV on the environment.

The control for the robot is derived such that it outputs the desired control wrench \mathbf{W}_c^B on the vehicle's body. Then, in a model-based manner, the desired propellers' thrust λ are computed by the inverse of the mapping M (i.e. $\lambda = M^{-1}\mathbf{W}_c^B$), where the platform specific map M , known as *allocation map* (see [7]), is well defined in our case thanks to the full-actuation of the robot.

III. SIMULATION RESULTS

A. Matlab Results

In order to improve the performance and the robustness of the original DP algorithm presented in Sec. II-C before integrating it in our vision-based interaction framework, we

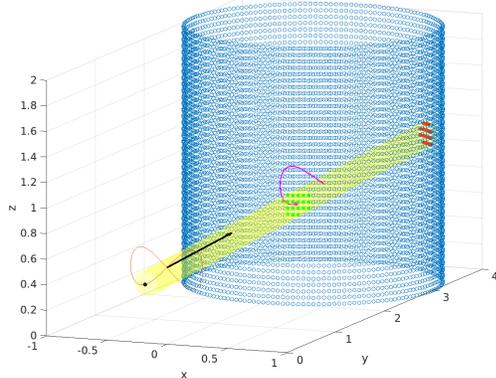


Fig. 5: Our modified version of the projection algorithm during execution. For each of the point to be projected, only a subset of entire point cloud is used (cf. the green points).

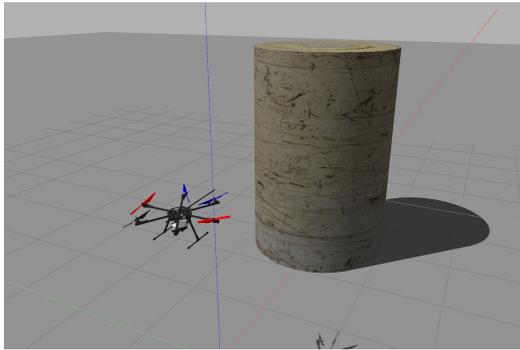
performed preliminary tests in a Matlab environment. In order to decrease the computational load of the algorithm when dealing with big point clouds, we observed that to project a sampled point $\xi_{des,k}$ of γ_d , it is not needed to take into account the entire point cloud \mathcal{C}_N but it is possible to consider just a subset \mathcal{C}_k of it, which should intuitively depend on $\xi_{des,k}$ and on the direction \hat{n} . Our choice has been to define \mathcal{C}_k as the set of points obtained intersecting \mathcal{C}_N with a virtual cylinder centered on $\xi_{des,k}$ and aligned with \hat{n} . The radius ρ_{cyl} of the cylinder is a function of the point cloud, i.e., it should guarantee that \mathcal{C}_k is not empty. If this happens, \mathcal{C}_k can be re-initialized with \mathcal{C}_N . Fig. 5 depicts the execution of the modified projection algorithm.

It is worth to remark that this shrewdness does not only decrease the computational time of the algorithm, but also improves its robustness. As a matter of fact, by limiting the query set in this way, only the point closed to the “true” projection are considered, thus guaranteeing a better outlier rejection and, as a consequence, a more precise projection. However, when dealing with a pre-scanned 3D object, it could still happen that some points on the backside surface are considered, as shown in Fig. 5 (cf. the red points). To deal with this side effect, we further modify the algorithm in order to discard the furthest points based on an adaptive threshold. The resulting algorithm was extensively tested in different scenarios providing good results, as shown in Fig. 4.

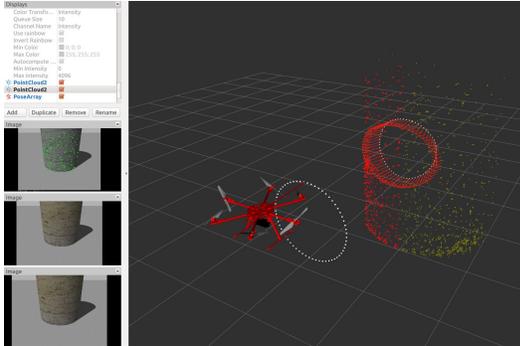
B. Gazebo Results

To assess the validity our approach upon unknown surfaces in a more realistic scenario, the overall vision-based interaction framework was implemented and extensively tested in a simulation environment based on Gazebo [21], the RotorS plugin [22], and a software-in-the-loop (SITL) version of the PX4 software [23]. The SITL receives the same commands through through ROS, as the real experimental setup. This significantly reduces the time required to validate the theory with real experiments and helps with tuning the different gains present in the control system. The simulated robot is a fully-actuated hexarotor, cf. [7], also employed in the experimental validation. In the aforementioned scenario, we modeled the interaction target as a cylindrical solid object whose surface is endowed with multiple features, cf. Fig. 6a.

At the beginning of the presented simulation, the aerial robot starts from a stationary initial configuration and is manually moved around the object of interest, whose position and



(a)



(b)

Fig. 6: The UAV and the target in the simulated environment in Gazebo (a) with a snapshot in RVIZ of the successful execution of the vision-based interaction planner (b).

bounding-box are supposed to be known, in order to scan the target object (phase 1). The implicit surface reconstruction is achieved from a stereo camera attached to the hexarotor’s frame by extracting a dense point cloud of the environment using ORB-SLAM2 [15]. To obtain the projected curve and the estimated normal directions on the interaction surface, the interaction planning algorithm module is activated and the desired path for the robot end-effector, defined by the user in its frame, is projected on the extracted point cloud along the specified direction (phase 2). As the path is parametrized in time, the resulting trajectory is autonomously tracked by the robot under the energy-based impedance controller. The correct execution of the projection algorithm can be appreciated from the snapshot of Fig. 6b, which depicts the scene in the ROS 3D visualization tool RVIZ, showing the aerial robot’s frame, the desired trajectory and its projection on the surface, and the associated normal vectors needed to define the robot orientation during the interaction.

IV. PRELIMINARY EXPERIMENTAL RESULTS

In this section, we experimentally validate the proposed vision-based interaction framework using a fully-actuated aerial robot. After a brief description of the aerial platform, the results of two different validation tests are presented and discussed. The interested reader can find the footage of such experiments in the supplementary video.

A. Setup Description

Our fully-actuated aerial robot was developed in-house based on an off-the-shelf carbon fiber frame with a diameter (rotor hub to rotor hub) of 0.68 m and a nominal total mass less than 2 kg, without batteries. For the generation of thrust, the system uses 11-inch propellers driven by six CM-2217 brushless motors with DYS SN40A electronic speed controllers. Each rotor has the capability to produce a maximum thrust of 12.5 N.

The vehicle is equipped with a Pixhawk 2.1 flight controller with integrated sensors. The Pixhawk runs the PX4 software [23], which handles interfacing to the sensors and actuators in addition to providing a modular framework which allows adding new control schemes. We augment the original PX4 software with custom-made modules that enables the control of fully-actuated multi-rotors. This involves modifying the PX4 control allocation module, and low-level orientation controllers. Using a multi-sensor fusion algorithm [24], the sensor data provided by the inertial measurement unit in the Pixhawk is fused together with the external pose information from the Optitrack motion capture system to provide reliable estimates of the UAV configuration \mathbf{H}_B^I and the body twist $\mathbf{T}_B^{B,I}$. We denote by $\mathbf{H}_B^I \in SE(3)$ the pose of the body-fixed frame Ψ_B with respect to the inertial-fixed frame Ψ_I , and by $\mathbf{T}_B^{B,I}$ the twist (generalized velocity) of Ψ_B with respect to Ψ_I , expressed in Ψ_B .

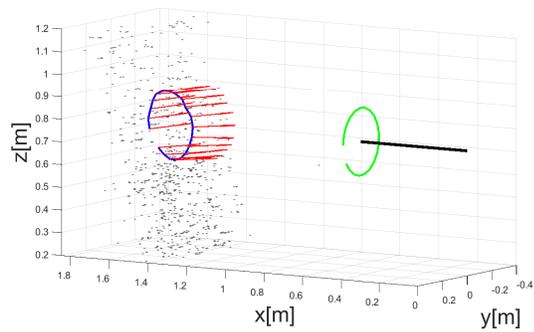
With reference to Fig. 3, the software of the multi-sensor fusion, the interaction planner, ORB-SLAM2, and the geometric impedance controller runs on a Linux PC exploiting the Robot Operating Systems (ROS) as middleware and interfacing with the PX4 software via serial USB communication at a rate of 150 Hz.

The experimental setup differs from the simulation for what concerns the sensor source used by ORB-SLAM2 for the point cloud extraction. Instead of a stereo camera, it uses an Intel RealSense D435 RGB-D camera. Apart from this hardware detail, the rest of the architecture remain unchanged. Finally, although ORB-SLAM2 provides an estimate of the camera pose (which can be related to the UAV pose), we do not use this information for the UAV control. As a matter of fact, we use the Optitrack motion capture system both in the state estimation and as a ground-truth for our vision-based framework at this preliminary stage, focusing only on indoor scenarios. We accomplish that by comparing the estimated camera pose and the measured camera pose (by Optitrack). Consequently, an error in the camera pose will also affect the estimation of the target surface.

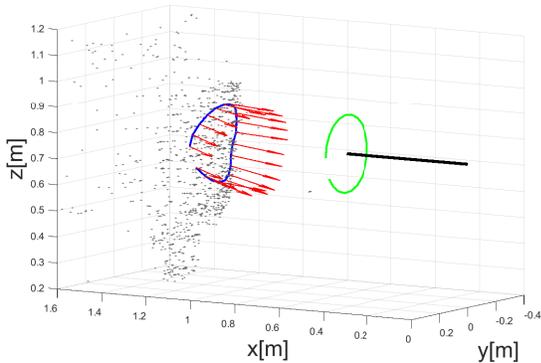
B. Experiment Scenario and results

The validation scenario for our vision-based framework is an interaction experiment with a textured surface, as shown in Fig. 1 and in Fig. 6a. In particular, as inspection targets we used both a tilted board and a panel with cylindrical shape.

The interaction planner executes two phases; the first is scanning the unknown target to get a dense point cloud. The assumption of high density features is needed for a correct



(a) Planar tilted surface



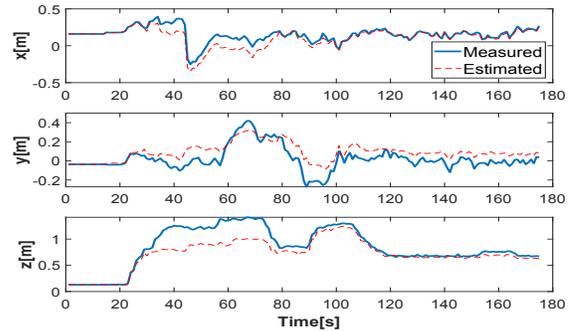
(b) Curved half-cylinder surface

Fig. 7: Exp. result: validation of the vision-based interaction planner for two target surfaces, showing the desired curve (green) in a plane fixed to the UAV's end-effector (black line). The projected curve (blue) onto the point cloud and the surface normals (red) are also shown.

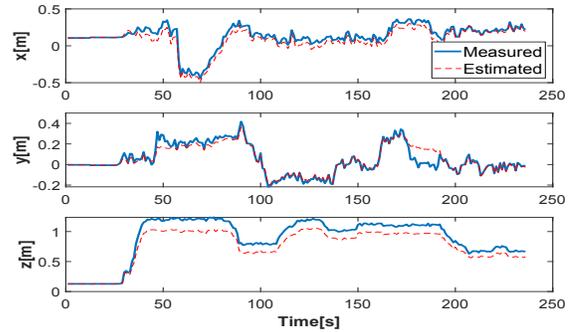
operation of ORB-SLAM2, but might be alleviated by choosing another SLAM algorithm. This phase is done by manual operation of the UAV, while extending it to autonomous scanning is a topic of future work. The second phase is projecting the desired curve, namely the (sampled) circle shown in Fig. 6b, using the aforementioned DP algorithm. Finally, the normal directions along the curve are calculated.

The results of the projection in Fig. 7 shows the robustness of our framework to the incompleteness and noise of the sampled surface. In particular, it should be appreciated that our approach eliminates the need to explicitly reconstruct the target's surface, which is a challenging task to perform relying only on-board computational capabilities.

In order to indirectly validate the accuracy of the constructed point cloud, we present in Fig. 8 the 3D position of the camera estimated by ORB-SLAM2 as well as the one measured by the motion capture system. Initially the estimation error was zero, as ORB-SLAM2 was initialized with the actual camera pose. The UAV takes off at about $t = 30$ s. After about 170 s of scanning the surface, the error decreases to an average norm of 10 cm due to some loop closure detected by ORB-SLAM2. As showcased in Fig. 8, the estimation error is mainly in the z -direction.



(a) Planar tilted surface



(b) Curved half-cylinder surface

Fig. 8: Exp. results: the Cartesian components of the estimated camera position (dashed red) by ORB-SLAM2 and the ground truth position (solid blue) captured by OptiTrack.

V. EVALUATION

This section describes some of the limitations present in our vision-based interactive planner and lists some possible adverse scenarios that might cause loss in the performance.

Following the block diagram in Fig. 3, three main building blocks can be observed: the vision-based perception module (left), the interaction planner (middle) and the geometric impedance controller (right). Each of these blocks influence the final result of the presented approach and, therefore, its own limitations are inherited in the overall solution. The perception module chosen for our experiments consists of an Intel RealSense D435 RGB-D camera feeding Orbslam 2, the go-to algorithm for localization and point-cloud building. This type of system uses the visible features present in the environment for mapping and localization purposes. As a consequence, environments that are poor in features would not suffice our intention. This limitation can be circumvented by adding permanent or temporal artificial features to the surface of interest (cf. Fig. 1). An additional pitfall of RGB-D cameras are surfaces that are highly or poorly reflective, which can cause false readings in the depth sensor.

The proposed interaction planner algorithm presented in Sec. II assumes the viability of the curve projection on the surface and a perfect disturbance rejection to satisfy both pose and force profiles. Some situations in which this algorithm might fail are, e.g., projection on surfaces which cannot contain the target curve or with projection vector parallel to the surface of concern, and scenarios with strong

external disturbances jeopardizing the interaction.

Lastly, the employed geometric impedance controller (Sec. II-D) could in principle be used to interact with any surface geometry as long as it satisfies the maximum roll and pitch angles of the UAV [20]. Moreover, the controller adopts the assumption of interaction with a passive environment. A violation of this assumption will not guarantee the contact-stability peculiar to this kind of controller.

All in all, the main contribution of this work should be considered the interaction planner module in addition to the vision-based interaction control framework as a whole, whereas some modules can be replaced by alternatives, e.g. different SLAM or interaction control algorithms.

VI. CONCLUSION

In this paper, we have presented a vision-based impedance control framework for the accomplishment of aerial interaction tasks which involve the physical contact with unknown textured surfaces. Thanks to our approach, an user-defined trajectory for the robot end-effector can be projected on an arbitrarily-shaped target surface, which can subsequently be inspected by the exertion of a contact force using impedance control. The presented results obtained in preliminary experimental validations with a fully-actuated hexarotor validate our approach and open the way to new research questions, which call for solutions to more advanced problems.

Future work includes the integration of computer vision algorithms for the state estimation in a visual servoing fashion and the use of pure on-board computation in order to target preliminary validation in outdoor scenarios. In this perspective, additional effort is still to be done to tackle the interaction with feature-less targets. Moreover, performing autonomous scanning, projection and interaction is a very promising direction. Some preliminary work in this direction has been reported in [25]. Finally, we are interested in exploring the integration of machine learning techniques to improve the robustness of our approach.

REFERENCES

- [1] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, 2018.
- [2] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi, "Modeling and control of a flying robot for contact inspection," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 3532–3537.
- [3] S. Kim, S. Choi, and H. J. Kim, "Aerial manipulation using a quadrotor with a two dof robotic arm," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 4990–4995.
- [4] F. Ruggiero, M. A. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Peréz, V. Lippiello, A. Ollero, and B. Siciliano, "A multilayer control for multirotor uavs equipped with a servo robot arm," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 4014–4020.
- [5] N. Staub, D. Bicego, Q. Sablé, V. Arellano, S. Mishra, and A. Franchi, "Towards a flying assistant paradigm: The othex," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6997–7002.
- [6] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, D. Bicego, and A. Franchi, "6d interaction control with aerial robots: The flying end-effector paradigm," *The International Journal of Robotics Research*, vol. 38, no. 9, pp. 1045–1062, 2019.
- [7] R. Rashad, J. B. Engelen, and S. Stramigioli, "Energy tank-based wrench/impedance control of a fully-actuated hexarotor: A geometric port-hamiltonian approach," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6418–6424.
- [8] K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pfändler, U. Angst, R. Siegwart, and J. Nieto, "An omnidirectional aerial manipulation platform for contact-based inspection," *arXiv preprint arXiv:1905.03502*, 2019.
- [9] Y. Zhou, X. Li, L. Yue, L. Gui, G. Sun, X. Jiang, and Y.-H. Liu, "Global vision-based impedance control for robotic wall polishing," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6022–6027.
- [10] I. Bonilla, E. J. Gonzalez-Galvan, C. Chavez-Olivares, M. Mendoza, A. Loredro-Flores, F. Reyes, and B. Zhang, "A vision-based, impedance control strategy for industrial robot manipulators," in *2010 IEEE International Conference on Automation Science and Engineering*. IEEE, 2010, pp. 216–221.
- [11] D. Song, T. Lee, and Y. J. Kim, "Artistic pen drawing on an arbitrary surface using an impedance-controlled robot," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4085–4090.
- [12] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [13] K. Alexis, G. Darivianakis, M. Burri, and R. Siegwart, "Aerial robotic contact-based inspection: planning and control," *Autonomous Robots*, vol. 40, no. 4, pp. 631–655, 2016.
- [14] M. Tognon, H. A. T. Chávez, E. Gasparin, Q. Sablé, D. Bicego, A. Mallet, M. Lany, G. Santi, B. Revaz, J. Cortés, *et al.*, "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.
- [15] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
- [16] A. R. Gaspar, A. Nunes, A. Pinto, and A. Matos, "Comparative study of visual odometry and slam techniques," in *Iberian Robotics conference*. Springer, 2017, pp. 463–474.
- [17] Y.-S. Liu, J.-C. Paul, J.-H. Yong, P.-Q. Yu, H. Zhang, J.-G. Sun, and K. Ramani, "Automatic least-squares projection of points onto point clouds with applications in reverse engineering," *Computer-Aided Design*, vol. 38, no. 12, pp. 1251–1263, 2006.
- [18] P. N. Azariadis and N. S. Sapidis, "Drawing curves onto a cloud of points for point-based modelling," *Computer-Aided Design*, vol. 37, no. 1, pp. 109–122, 2005.
- [19] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," Ph.D. dissertation, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.
- [20] R. Rashad, F. Califano, and S. Stramigioli, "Port-hamiltonian passivity-based control on se (3) of a fully actuated uav for aerial physical interaction near-hovering," *IEEE Robotics and automation letters*, vol. 4, no. 4, pp. 4378–4385, 2019.
- [21] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.
- [22] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Rotors—a modular gazebo mav simulator framework," in *Robot Operating System (ROS)*. Springer, 2016, pp. 595–625.
- [23] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 6235–6240.
- [24] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," in *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2013, pp. 3923–3929.
- [25] B. Sirmacek, R. Rashad, and P. Radl, "Autonomous uav-based 3d-reconstruction of structures for aerial physical interaction," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, pp. 601–605, 2019.