Semi-autonomous Unmanned Aerial Manipulator Teleoperation for Push–and–Slide Inspection using Parallel Force/Vision Control[⋆]

Simone D'Angelo[a,*], Mario Selvaggio[a,b], Vincenzo Lippiello[a], Fabio Ruggiero[a]

[a] *PRISMA Lab, Department of Electrical Engineering and Information Technology, University of Naples Federico II, Via Claudio 21, 80125, Naples, Italy*
[b] *Centro Servizi Metrologici e Tecnologici Avanzati (CeSMA), University of Naples Federico II, Corso Nicolangelo Protopisani, 80146, Naples, Italy*

Abstract

Performing inspection and maintenance tasks with aerial robots in complex industrial facilities require high levels of maneuverability and dexterity. As full autonomy still struggles to provide robust solutions due to limited adaptability and high development costs, this study explores the paradigm shift towards shared control teleoperation for tilting unmanned aerial manipulators (UAMs). The research initially focuses on integrating onboard camera measurements and interaction force feedback within a parallel force/vision controller for push–and–slide inspection tasks. The control loop lends itself to the development of a semi-autonomous operation architecture that enables a human operator to easily accomplish the task by means of a simple input device. The paper presents a user study evaluating task completion performance with human–in–the–loop control versus fully autonomous execution. Statistical analysis of 20 user experiences provides insights into the levels of autonomy necessary for effective task completion. Among the analyzed control modalities, statistically significant differences arise when the sliding feature is autonomous, denoting it as the most difficult to manually accomplish. The investigation is conducted within a simulated environment to ensure the safety of sensitive instruments and accommodate users with varying levels of expertise. By proposing shared control architectures, this research addresses the challenges of autonomous UAM operations in hazardous industrial environments, highlighting the benefits of human oversight and control in enhancing task efficiency and safety.

Keywords: Unmanned Aerial Manipulator, Tilting Drone, Interaction Control, Visual Servoing, Shared Control, Push–And–Slide

[⋆] This project has received funding from the AI-DROW project, in the frame of the PRIN 2022 research program, grant n. 2022BYSBYX, funded by the European Union Next-Generation EU; the COWBOT project, in the frame of the PRIN 2020 research program, grant n. 2020NH7EAZ_002.

[*] Corresponding author

1. Introduction

Industrial facility inspections often involve Non-Destructive Testing (NDT) as the standard method for assessing the condition of concrete structures. Ultrasonic probes are frequently used to measure their thickness, ensuring material integrity and the absence of internal damage. Traditionally, these tests require direct contact between the probe and the surface, necessitating human inspectors to utilize scaffolding or rope-access systems to reach inspection sites.

Unmanned Aerial Vehicle (UAV) deployment for contact-based inspections has gained significant popularity due to several key advantages: drones enhance safety by accessing hazardous or inaccessible areas without exposing humans to risks [1] lowering costs [2] and increasing efficiency and speed by covering large areas quickly, thereby reducing plant downtime [3]. Unmanned aerial manipulators (UAM) combine these vehicles to stick or articulated arms: by exploiting higher Degrees of Freedom (DoFs) and advanced sensors, they have the potential to revolutionize industrial inspections across various fields [4].

In this context, the power of semi-autonomous inspection operations lies in the synergy between human expertise and machine efficiency. Unlike fully autonomous ones, semi-autonomous systems empower operators to apply their judgment and intuition, leveraging cognitive abilities to interpret complex scenarios that may elude purely algorithmic control. Furthermore, by maintaining human oversight within the inspection loop, semi-autonomous systems adapt to unpredictable circumstances leveraging on both the benefits of autonomous control and human awareness and expertise [5]. Moreover, haptic feedback device incorporation enhances semi-autonomous inspection systems [6].

This work extends the research in [7], where real-world experiments were conducted with the same UAM platform (see Fig. 1) proving the feasibility of inspection tasks similar to those examined here.

1.1. Problem Statement

During NDT inspections, the UAM must adapt to changes in surface topology, detect and locate key inspection points [8], and maintain a safe distance from the surfaces. In the following, the inspection is defined as a mission based on a reference trajectory in the image space and a desired force profile, allowing the robot to complete the push–and–slide inspection by merging visual and force feedback. The human operator is integrated into the control loop within a leader-follower paradigm commanding the UAM inputting the set-points to a parallel force/vision control framework.

The study employs simulation based on physical simulators, providing a faithful reproduction of the actual hardware to be controlled. By using high-fidelity simulations, the framework ensures that the control strategies are robust and applicable to real-world scenarios, effectively preparing the system for practical deployment (see e.g., [9, 10, 11]). This approach allows to safely conduct

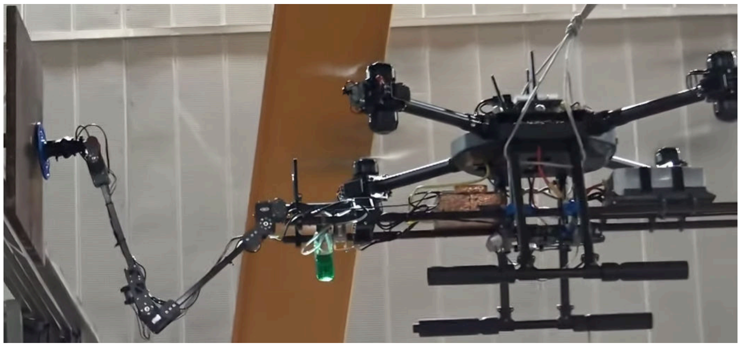


Figure 1: Indoor NDT inspection executed by an autonomous tilting aerial manipulator [7].

experiments and examine the outcome of the operations performed by different users creating a training environment.

1.2. Outline

The remaining part of the paper is organized as follows: Section 2 introduces the problem in the state–of–the–art while showing the main contributions. The proposed system is presented in Section 3 focusing on the mathematical modeling and giving miscellaneous derivations useful in the subsequent parts; Section 4 describes the control framework, the proposed technological solution, and the integration with the haptic device. The human–in–the–loop simulations in ROS/Gazebo and the experimental campaign results are reported between Section 5 and Section 6. Finally, Sections 7–8 discuss the results and conclude the work by analyzing future work and functionalities to be developed.

2. Related Works

Recent years have seen numerous advancements in hardware development, resulting in increased heterogeneity among UAMs [12] and additional complexities in modeling and control. Usually, UAMs control can be classified in centralized and decentralized [2], as the ones proposed in this paper and, more generally, interaction force control techniques are divided into indirect [13] and direct [14] algorithms, allowing the closure or not of a force-errors loop in the architecture.

On the other hand, parallel control laws combine these interaction controllers to the benefits of motion controllers in hybrid schemes [15]. Anyway, accurate measurements of robot position and orientation are not always available in real-world scenarios, leading to a bias towards visual servoing use as opposed to Cartesian motion control in parallel paradigm. Moreover, a visually servoed robot can operate without prior knowledge of the positions of objects in its workspace. In manufacturing, this approach removes the need for robot teaching, enabling tasks like assembly with imprecise fixtures or handling unoriented and moving components, such as those on overhead transfer lines [16].

Visual servoing and impedance filter combination is presented initially in [17, 18], binding the filter to a Position-Based Visual Servoing (PBVS) algorithm, with subsequent application to various areas, such as wall polishing [19, 20] and peg-in-hole insertion [21]. In recent years, this theory has increasingly been implemented to aerial platforms: for instance, in [22] and [23], the interaction force is controlled to enable writing on a given planar surface and later extended to general shapes [24].

Recent studies exploring the integration of visual and force feedback, such as [25, 26, 27], present a focus on techniques for visual features extraction. Here, the authors typically use indirect force controllers, such as admittance or impedance approaches, to adjust image-space references while interacting. These solutions lack precision in force tracking, and conventional flat aerial platforms are limited in their motion control capabilities.

Existing literature indicates that standard multi-rotor UAVs effectively control only four dimensions out of their full six DoFs, with the remaining two dimensions indirectly managed to maintain hovering stability [28, 29]. Consequently, visual servoing often relies on assumptions using the virtual camera frame. In contrast, tilting and tilted UAVs can exert thrust and torques in every direction, enhancing their flying capabilities. Their use in contact-based scenarios is widespread: in [30], a nonlinear model predictive control handles different flight phases and optimizes force and motion tracking; in [31], the drone is equipped with an articulated arm detecting welds on a pipe; and in [32], a direct force control and online task optimization are utilized. Additional fully actuated platforms equipped with a stick are presented in [33, 34, 35, 36] showing high dexterity and performances in Cartesian motion control. The literature therefore reveals an absence in their use for camera-guided inspection.

2.1. From Fully Autonomy to Shared Control

Despite all, autonomous systems remain insufficiently robust for applications within unknown environments, making teleoperation indispensable for safely deploying UAMs in such scenarios [37, 38]. Moreover, as autonomy increases, challenges arise in safety analysis, verification, and trust. Traditional methods struggle with unforeseen hazards, greater uncertainty, and the need to assess system intent, requiring new mechanisms to ensure reliability and safety [39].

Considering the need for precise control of motion and force, teleoperation places high cognitive and physical demands on operators. Implementing shared control with an abstraction layer allows an autonomous controller to assist, easing task execution [5]. This approach proves advantageous for tasks with numerous degrees of freedom or control of rotations [40, 41, 42]. shared control architectures frequently incorporate haptic feedback to provide operators with critical information about the physical constraints impacting the robotic system [6, 11, 43]. However, the application of haptic shared control teleoperation remains underexplored in the context of aerial inspection with UAM [9].

Bilateral teleoperation has demonstrated utility in industrial applications, such as object-pulling tasks [44]. In a force-based bilateral teleoperation framework for aerial robots interacting physically with their environment, as illus-

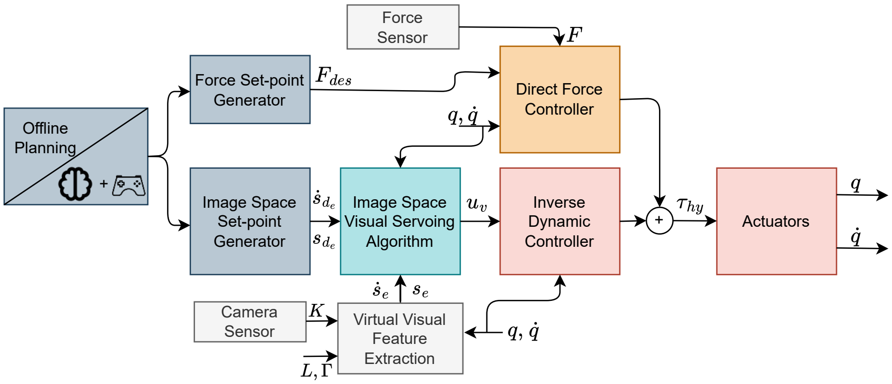


Figure 2: System architecture summarizing the proposed control law. Either an offline planner or a teleoperation system manages set-points generation, inserting the human operator (represented by the brain) into the control loop using a haptic device (schematically represented by the joystick). All the variables will be described in Section 4.

trated in [45], the operator controls free motion and applies surface forces via a passive tool. Yet, practical experiments highlight the difficulty of achieving decoupled motions in omnidirectional platforms, due to inherent physical limitations in operator input control [46]. Additionally, shared control teleoperation has been proposed for cable-suspended UAM to manage pick-and-place tasks at varying priority levels, albeit only in a sequential manner [47].

2.2. Goal and Contribution

Building upon existing literature, this paper presents a semi-autonomous teleoperation architecture for a UAM to perform push–and–slide interaction tasks. The UAM utilizes feedback from a camera and a force sensor within a novel parallel force/vision control law, briefly outlined in Fig. 2. More specifically, the architecture integrates the advantages of precise force tracking given by selective direct force control techniques to Image-Based Visual Servoing (IBVS) algorithm. Control is shared between the human operator and the autonomous controller through a haptic device, offering a necessary level of abstraction to reduce the user effort [5]. The operator sets references and receives haptic feedback to ensure task accuracy, with real-time tactile feedback enabling intuitive control adjustments. A human-subject study identifies key factors for optimizing shared control and balancing human input and autonomous responses. Simulations allow users of varying experience to test these modalities, and the system can be directly translated to hardware-compatible code [48, 49].

Based on the above description, to the best of the authors' knowledge, the following represent the paper's novel contributions:

- Parallel force/vision control paradigm for UAM interaction: implementation of a novel approach that combines vision and forces feedback solving major problems in literature and never applied to a complex tilting UAM;

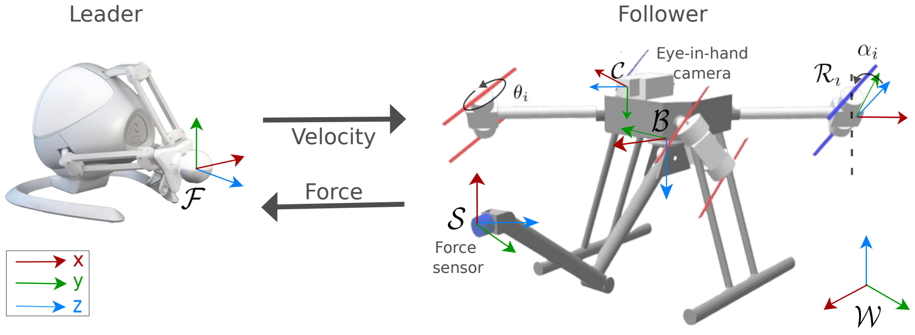


Figure 3: Aerial manipulator shared control teleoperation setup depicting the main frames.

- Haptic and shared control UAM teleoperation system: the incorporation of a haptic feedback system within the control framework that enables human operators to generate set-points and receive tactile feedback, enhancing task completion guidance;

- Human subjects study: conducting a user study with 20 subjects on a push–and–slide NDT task using a UAM aerial robot to identify preferred haptic shared control modalities in a relevant industrial application and to evaluate how the envisioned user input/control/feedback combinations lead to a measurable difference in the task execution.

3. System Description

The proposed system is based on a leader-follower paradigm, explored in the context of a telemanipulation system. Two agents define this system: a leader haptic device and a follower UAM consisting of a tilting coaxial octa-rotor equipped with a 6-DoFs robotic arm. The manipulator is mounted beneath a UAV (see Fig. 3, right side), allowing the entire task to be executed during the drone's flight. Articulated arms provide enhanced flexibility, precision, and access to confined spaces [50], improving efficiency and safety compared to traditional rigid sticks. Moreover, considering that the robot typically moves slowly (or not at all) during a force control task, the UAM's controller can be decoupled between the arm and flight controllers thanks to the hypothesis of quasi-stationary flight during the task execution [2, 51].

The operator remotely controls the robot piloting the haptic device to accomplish specific tasks. A camera mounted on top of the aerial platform provides visual feedback and helps determine the relative pose between the surface and the E–E. The paper focuses on implementing kinematic and workspace constraints on the follower side. By leveraging haptic guidance, the system empowers the operator to navigate these constraints effectively during teleoperation.

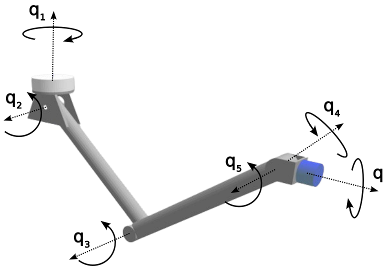


Figure 4: Kinematic scheme of the attached manipulator characterized by 6 revolute joints.

3.1. Leader Haptic Device

The haptic device in Fig. 3 represents the leader in the considered paradigm: let $\mathcal{F}$ be the frame attached to its handle. The device considered in this study has three DoFs, allowing it to generate precise set-points as inputs for task completion. It interprets the operator's physical inputs and translates them into velocities in $\mathcal{F}$. The UAM then receives these set-points, which are properly converted into desired positions in the image space and interaction force references. This integration ensures that the UAM can perform tasks with high accuracy and responsiveness, guided by the real-time inputs from the haptic device. The communication is bilateral: the robot itself can send information to the device during task execution as haptic force feedback, which the operator can use to be guided in task completion.

3.2. Follower UAM - Mathematical Model

The aerial platform consists of a coaxial tilting octa-rotor drone equipped with a 6-DoFs robotic manipulator. It features four independent servo motors, enabling the drone to maintain parallel hovering relative to the ground while maneuvering along the horizontal plane, or to achieve desired pitch/roll angles.

Concerning Fig. 3, let $\mathcal{W}$ be the world inertial reference frame; $\mathcal{B}$ be the body-fixed frame whose origin is coincident with the drone's center of mass (CoM); and $\mathcal{S}$ be the frame attached to the manipulator's E–E.

The pose of $\mathcal{B}$ in $\mathcal{W}$ is dictated by $T_b(p_b, o_b) = (p_b, R_b) \in SE(3)$, with $p_b \in \mathbb{R}^3$ the position of $\mathcal{B}$ in $\mathcal{W}$ and $R_b \in SO(3)$ the rotation matrix expressing the attitude of $\mathcal{B}$ in $\mathcal{W}$. Such orientation can also be expressed through the minimal roll-pitch-yaw (Euler angles) representation as $o_b = \begin{bmatrix}\theta & \phi & \psi\end{bmatrix}^\top \in \mathbb{R}^3$.

The attached robotic manipulator is made of a succession of links connected by $n = 6$ actuated revolute joints characterized through the joint vector $q(t) \in \mathbb{R}^n$ as in Fig. 4. A force sensor is considered on the E–E's tip and a camera sensor is mounted on the UAM. A frame $\mathcal{C}$, fixed in $\mathcal{B}$, is attached to the camera.

3.2.1. Kinematic Model

The arm's direct kinematic problem is solved through the product of exponentials formula [51], to retrieve the arm's E–E pose, $\mathcal{S}$, in $\mathcal{B}$, given by

$T_s^b(q) = (p_s^b, R_s^b) \in SE(3)$, with $p_s^b \in \mathbb{R}^3$ and $R_s^b \in SO(3)$ the position and the orientation of $\mathcal{S}$ in $\mathcal{B}$, respectively. The arm's E–E pose in $\mathcal{W}$ can be easily obtained as $T_s(q, \theta) = T_b(p_b, o_b)T_s^b(q) \in SE(3)$.

The arm differential kinematic is expressed by the relation $\upsilon_s^b = J(q)\dot{q}$, with $\upsilon_s^b \in \mathbb{R}^6$ the E–E twist related to $\mathcal{B}$ and $J(q) \in \mathbb{R}^{6\times n}$ the Jacobian expressed in $\mathcal{B}$. The Jacobian is obtained iteratively [51], and its i-th column is expressed by

$$J_i(q) = \mathrm{Ad}_{e^{[\zeta_1]q_1} e^{[\zeta_2]q_2} \dots e^{[\zeta_{i-1}]q_{i-1}}} \zeta_i \in \mathbb{R}^6, \tag{1}$$

with $\zeta_i \in \mathbb{R}^6$ the screw axis associated to the revolute joint q_i, for $i = 2, \dots, n$, and $J_1(q) = \zeta_1$. The adjoint matrix, $\mathrm{Ad}_T \in \mathbb{R}^{6\times 6}$, allows the change of the representation frame given a generic transformation matrix $T \in SE(3)$ [51].

3.2.2. Dynamic Model

The aerial manipulator dynamic model is retrieved through the recursive Newton-Euler approach exploiting once again the screw theory [51]. In quasi-stationary flight conditions, the kinematic and dynamic models of the robotic arm can be decoupled from the UAV one [2]. Let $\xi = \begin{bmatrix} p_b^\top & o_b^\top & q^\top \end{bmatrix}^\top \in \mathbb{R}^N$ be the generalized vector of joint positions of the whole system, resulting in a robot with a total of $N = 6 + n = 12$ controllable DoFs. The dynamic model's compact form can be expressed as

$$B(\xi)\ddot{\xi} + h(\xi, \dot{\xi}) = u, \tag{2}$$

where $B(\xi) \in \mathbb{R}^{n\times n}$ is the inertia matrix; $h(\xi, \dot{\xi}) \in \mathbb{R}^n$ accounts for the gravitation, centrifugal, and Coriols terms; and $u \in \mathbb{R}^N$ is the control input vector. The total input vector stacks the UAV total thrust, $u_{T_{UAV}} \in \mathbb{R}^3$, the UAV torques around the $\mathcal{B}$ axes, $\tau_{UAV} \in \mathbb{R}^3$, and the joint torques command, $\tau_{hy} \in \mathbb{R}^n$, that is $u = \begin{bmatrix} u_{T_{UAV}}^\top & \tau_{UAV}^\top & \tau_{hy}^\top \end{bmatrix}^\top$.

3.2.3. Control Allocation Problem

Considering the aerial platform, the same assumptions as in [52] are considered to retrieve the system model and solve the control allocation problem. This last consists in finding the rotors speed $\omega \in \mathbb{R}^8$ and tilt angles $\alpha \in \mathbb{R}^4$ satisfying $\begin{bmatrix} u_{T_{UAV}}^\top & \tau_{UAV}^\top \end{bmatrix}^\top = A(\alpha)\omega^2$. In the tilting UAV case, the non-linear mapping between the control inputs and the rotors' angular velocities is thus handled by the allocation matrix $A(\alpha) \in \mathbb{R}^{6\times 8}$, function of the current tilt angles vector α.

A frame $\mathcal{R}_i$ ($i = 1, ..., 4$), attached to each coaxial rotor group has to be considered, as in Fig. 3. Each $\mathcal{R}_i$ will be rotated at an angle α_i during the drone's flight: in this configuration, the drone can exert thrust along each Cartesian axis. The thrust magnitude u_{T_k} and the drag torque τ_k of the k-th rotor ($k = 1, ..., 8$) are given by

$$u_{T_k} = k_f \omega_k^2, \quad \tau_k = k_q \omega_k^2, \tag{3}$$

with $k_f, k_q > 0$ the motors coefficients.

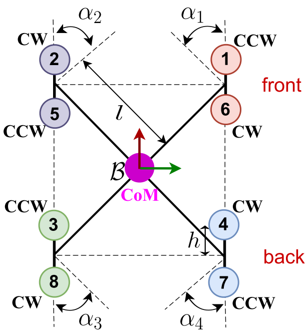


Figure 5: Omnidirectional tilting UAV scheme: clockwise (CW) and counter-clockwise (CCW) rotors configuration is depicted. Notable parameters are l, h (the arm's length and height, respectively) while α_i is the i-th tilting angle.

Such a mapping is linearized defining the vertical, $f_{v,k} \in \mathbb{R}$, and lateral, $f_{l,k} \in \mathbb{R}$, forces of the k-th motor as [52, 53]

$$f_{v,k} = u_{T_k} \cos(\alpha_i), \quad f_{l,k} = u_{T_k} \sin(\alpha_i), \tag{4}$$

where the association between i and k is done according to Fig. 5. With these considerations, the allocation problem becomes

$$\begin{bmatrix} u_{T_{UAV}} \\ \tau_{UAV} \end{bmatrix} = \bar{A} f_{dec}, \quad f_{dec} = \begin{bmatrix} f_{v,1} & f_{l,1} & \cdots & f_{v,8} & f_{l,8} & \end{bmatrix}^\top \in \mathbb{R}^{16}, \tag{5}$$

where $f_{dec} \in \mathbb{R}^{16}$ is the vector stacking the forces in (4), and $\bar{A} \in \mathbb{R}^{6\times16}$ is the static allocation matrix:

$$\bar{A} = \begin{bmatrix} 0 & -\sin(\theta_1) & \cdots & 0 & -\sin(\theta_8) \\ 0 & \cos(\theta_1) & \cdots & 0 & \cos(\theta_8) \\ -1 & 0 & \cdots & -1 & 0 \\ -l\sin(\theta_1) & -h\cos(\theta_1) - c_1\frac{k_q}{k_f}\sin(\theta_1) & \cdots & -l\sin(\theta_8) & -h\cos(\theta_8) - c_8\frac{k_q}{k_f}\sin(\theta_8) \\ l\cos(\theta_1) & c_1\frac{k_q}{k_f}\cos(\theta_1) - h\sin(\theta_1) & \cdots & l\cos(\theta_8) & c_8\frac{k_q}{k_f}\cos(\theta_8) - h\sin(\theta_8) \\ -c_1\frac{k_q}{k_f} & lc^2(\theta_1) + ls^2(\theta_1) & \cdots & -c_8\frac{k_q}{k_f} & lc^2(\theta_8) + ls^2(\theta_8) \end{bmatrix}. \tag{6}$$

The static allocation matrix depends on $l > 0$ and $h > 0$, the drone arm length and height respectively, and on θ_i, the fixed angular position of the i-th propeller with respect to $x_{\mathcal{B}}$, the body-fixed heading direction.

To minimize the system energy consumption [54], the allocation problem is solved by computing the Moore-Penrose pseudoinverse of $\bar{A}$ and inverting (5). Finally, imposing

$$\omega_k = \sqrt{\frac{1}{k_f}\sqrt{f_{v,k}^2 + f_{l,k}^2}}, \quad \alpha_i = \operatorname{atan2}(f_{l,k}, f_{v,k}), \tag{7}$$

the motor velocities ω_k and the tilt angles α_i are retrieved and directly used to actuate the aerial platform.

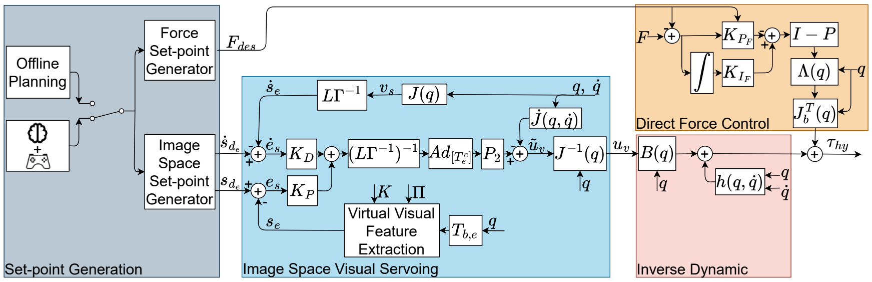


Figure 6: Aerial manipulator control scheme: the offline planner or the human operator guiding the haptic device (in grey) sends inputs to two set-point generators to convert them in the image space and force set-point. They are sent to an IBVS algorithm (in blue) and a direct force control algorithm (in orange). An inverse dynamic control scheme (in red) converts the virtual acceleration in joint torques. The matrices P and P_2 select the Cartesian-space axis to control the motion or the interaction force.

4. UAM Control

The dynamic model described in (2) consists of a set of N coupled nonlinear differential equations. To simplify the problem, we assume quasi-stationary flight, where the drone maintains a fixed position relative to the marker while the robotic arm inspects the surface. Figure 6 illustrates the entire aerial manipulator control framework. The subsequent subsections will detail the different task phases, each corresponding to a distinct control modality.

4.1. Phase 1: Visual Servoing

The first stage of the task involves guiding the UAM to establish contact with the working surface. Under the assumption of quasi-stationary flight, this stage can be divided into two sub-tasks. First, the tilting UAV is guided within the scene using visual information to maintain a fixed distance from the surface. Concurrently, the manipulator is maneuvered to establish contact with the surface. Both sub-tasks rely on visual feedback and image processing techniques to extract necessary data from the camera sensor. Exploiting the tilting capabilities, it is possible to avoid the virtual camera model assumption employed in [25, 26, 27] and effectively control the full 6-D UAV pose in the image space.

A tag is attached to a planar surface whose pose is unknown. Using image elaboration techniques, the marker is detected and visual features are extracted. Four feature points coincident with the marker corners are defined based on the classical pin-hole camera model.

4.1.1. Camera Model

By calibrating the camera sensors and retrieving the related intrinsic and extrinsic parameters, it will be possible to convert the tag position from the camera frame to the image plane and vice-versa. Let $\tilde{p}_o^c = \begin{bmatrix} p_o^{c\top} & 1 \end{bmatrix}^\top \in \mathbb{R}^4$ be the augmented position vector of the detected marker in $\mathcal{C}$. This point can be represented in pixels normalized coordinates defining the relative feature point $s \in \mathbb{R}^2$:

$$p^c = \begin{bmatrix} x^c \\ y^c \\ z^c \end{bmatrix} K\Pi\tilde{p}_o^c = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{p}_o^c \in \mathbb{R}^3, \tag{8}$$

$$s = \begin{bmatrix} X \\ Y \end{bmatrix} = \frac{1}{z^c} \begin{bmatrix} x^c \\ y^c \end{bmatrix}, \tag{9}$$

where K is the intrinsic camera parameter matrix [55].

4.1.2. Image Jacobian Matrix

During task execution, the marker's pose changes relative to the flying platform. To fully describe this pose in the 6D Cartesian space, at least four coplanar and non-collinear points are used to solve the perspective-n-points (PnP) problem [55]. These points, corresponding to the marker's corners, are extracted and fed back into the visual servoing algorithm. Moreover, the E-E's position in the image space is constantly determined by considering four virtual visual features on its tip and its twist vector in the Cartesian space. These features correspond to the corners of a square centered on $\mathcal{S}$, with its position in $\mathcal{C}$ denoted as $p_b^c \in \mathbb{R}^3$. The visual servoing feedback is then composed of two different feature vectors $s_o = \begin{bmatrix} s_{o,1} & s_{o,2} & s_{o,3} & s_{o,4} \end{bmatrix}^\top$ and $s_e = \begin{bmatrix} s_{b,1} & s_{b,2} & s_{b,3} & s_{b,4} \end{bmatrix}^\top$ describing the inspection surface pose and the E–E's pose in the image space, respectively. Considering fixed the work surface position, their time variations are only functions of the camera motion and the arm motion in $\mathcal{C}$, respectively:

$$\dot{s}_o = \begin{bmatrix} L_{o,1} \\ L_{o,2} \\ L_{o,3} \\ L_{o,4} \end{bmatrix} \Gamma^{-1}(p_o^c)v^c, \quad \dot{s}_e = \begin{bmatrix} L_{b,1} \\ L_{b,2} \\ L_{b,3} \\ L_{b,4} \end{bmatrix} \Gamma^{-1}(p_b^c)v_b^c, \tag{10}$$

where v^c, $v_b^c \in \mathbb{R}^6$ are the Cartesian velocity of the UAV and of the arm opportunely referred to $\mathcal{C}$; $\Gamma(\cdot) = \begin{bmatrix} -I_3 & \widehat{(\cdot)} \\ 0_3 & -I_3 \end{bmatrix} \in \mathbb{R}^{6\times 6}$ and $\widehat{(\cdot)}$ is the skew transformation matrix operator. This mapping is obtained through the image jacobian $J_{img_j} = L_j\Gamma^{-1}(p_{j,i}^c)$ where

$$L_{j,i} = \begin{bmatrix} -\frac{1}{z_{j,i}^c} & 0 & \frac{X_{j,i}}{z_{j,i}^c} & X_{j,i}Y_{j,i} & -1 - X_{j,i}^2 & Y_{j,i} \\ 0 & -\frac{1}{z_{j,i}^c} & \frac{Y_{j,i}}{z_{j,i}^c} & 1 + Y_{j,i}^2 & -X_{j,i}Y_{j,i} & X_{j,i} \end{bmatrix}, \tag{11}$$

is the i-th row of the interaction matrix of a set of i points: in the examined case the whole matrix will be $L_j \in \mathbb{R}^{8\times 6}$.

4.1.3. Image-Based Visual Servoing

Building on the theory in Sections 4.1.1 and 4.1.2, the UAM has to detect markers on the inspection surface and nullify the relative image-space position error exploiting a purely image-based visual servoing algorithm. The proposed control law took inspiration from [52], with several modifications to perform computations directly in the image space.

Let $s_d \in \mathbb{R}^8$ be the desired feature vector representing the drone goal position in the camera frame and $\dot{s}_d$ its time derivative. These coordinates represent the tag corners' position in normalized pixel coordinates and their variation with respect to the camera movement. The controller's goal is to nullify the following errors

$$e_s(t) = s_d - s_o(t), \quad \dot{e}_s(t) = -\dot{s}_o(t). \tag{12}$$

To ensure null error and preserve the steady state during the next control phases (e.g., the forces on the arm should push away the platform during the interaction) an image space integral error is added in the controller formulation:

$$e_i(t) = \int e_s(t)\mathrm{d}t. \tag{13}$$

A standard proportional-integral-derivative (PID) controller characterize the retrieved image-space controller:

$$u^c = K_{P_{img}} e_s + K_{D_{img}} \dot{e}_s + K_{I_{img}} e_i, \tag{14}$$

where $K_{P_{img}}$, $K_{D_{img}}$ and $K_{I_{img}} \in \mathbb{R}^{8\times 8}$ are the proportional, derivative, and integral gains matrices, respectively, and u^c represents the desired accelerations computed in the image space. To complete control law and compensate also for the dynamic effects, they need to be referred to $\mathcal{B}$:

$$u = Ad_{[T_c]}(L_c \Gamma^{-1}(p_o^c))^{-1} u^c \in \mathbb{R}^6, \tag{15}$$

with T_c indicating the transformation matrix from $\mathcal{C}$ to $\mathcal{B}$ and $J_{img} = L_c\Gamma^{-1} \in \mathbb{R}^{n\times n}$ being the image Jacobian. The result is then used in an inverse dynamic control law to compute the desired thrusts $u_{T_{UAV}}$ and torques τ_{UAV} while compensating the UAV dynamics:

$$u_{T_{UAV}} = R_b^{\top}(-mg + m \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}) \in \mathbb{R}^3, \tag{16}$$

$$\tau_{UAV} = R_b^{\top}(\omega \times J\omega + J \begin{bmatrix} u_4 \\ u_5 \\ u_6 \end{bmatrix}) \in \mathbb{R}^3. \tag{17}$$

The control input vector is then converted in rotors speed and tilt angles inverting (5).

Given that the presented IBVS algorithm addresses a regulation problem, it is crucial to prevent excessively high acceleration when the UAM is distant from the marker. Therefore, the IBVS algorithm is only activated after an initial position-based visual servoing step. In this preliminary step, the drone follows a desired Cartesian trajectory computed online by detecting the marker at the beginning of the task.

4.2. Phase 2: Parallel Force/Vision Controller

Once the errors in (12) fall within a specified threshold, and while the UAM continues to minimize these errors, the robotic manipulator can start and complete the inspection task on the detected surface. The E–E motion and interaction force with the environment are effectively controlled by integrating visual feedback with force sensor measurements in a novel parallel control. Throughout the task execution, the controller described in Section 4.1.3 remains active to maintain a constant distance between the flying platform and the interaction surface.

The controller in Fig. 6 combines an image-space inverse dynamic controller and a direct force controller. Like in [34], the factor $0 \leq \lambda \leq 1$ is introduced to represent the transition between contact-less ($\lambda = 0$) and contact-based motion ($\lambda = 1$). A cosine step function interpolated between 0 and 1 is used to smooth the transition between these two modalities [30]. When this transition is needed, direct force control can track the desired force reference along some directions while the IBVS controller tracks the E–E position along the other axes. The force controller performances are related to the tracking precision. If the image space position error is beyond the given threshold, the priority is given to the tracking task, and then the force control is triggered again. The matrix P performs the analyzed selection:

$$P = \begin{bmatrix} I_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & \bar{P} \end{bmatrix} \in \mathbb{R}^{6\times}6, \text{ with } \bar{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1-\lambda \end{bmatrix} \in \mathbb{R}^{3\times 3}. \tag{18}$$

The matrix is used in the following proportional-integral (PI) control scheme [34] to track the force reference $f_{des}(t) \in \mathbb{R}^3$ while nullifying the force error $e_f(t) = \begin{bmatrix} 0_3^\top \\ f(t)^\top - f_{des}(t)^\top \end{bmatrix} \in \mathbb{R}^6$ through

$$\tau_{fc} = J_b^\top(q)\Lambda(q)(I_6 - P)\Big(- \begin{bmatrix} 0_3 \\ f_{des}(t) \end{bmatrix} + K_{P_F} e_f(t) + K_{I_F} \textstyle\int e_F(\tilde{t})d\tilde{t}\Big), \tag{19}$$

where $\Lambda(q) = J_b^{\#^\top}(q)B(q)J_b^{\#}(q)$ is the inertia matrix in $\mathcal{S}$, $J_b = Ad_{T_s^b{}^{-1}}[J(q)]$ is the body Jacobian [51], $K_{P_F}, K_{I_F} \in \mathbb{R}^{6\times 6}$ are positive definite gains matrices. The force error e_F is proven to converge to zero for positive definite gains in [51].

The control scheme is completed with the design of an image space motion controller. Taking into account (2), the manipulator joint torques are computed choosing a properly virtual control input $\ddot{q} = u_v \in \mathbb{R}^n$. Being s_{e_D} the desired E–E position in the image space, it is then possible to impose

$$u_v = J^{-1}\Big(P_2 Ad_{T_s^c}\tilde{u}_v - \dot{J}(q,\dot{q})\dot{q}\Big), \tag{20}$$

where

$$\tilde{u}_v = (L_e \Gamma(p_e^c)^{-1})^{-1}(K_{P_e} e_{s_e} + K_{D_e}\dot{e}_{s_e}), \tag{21}$$

with $K_{P_e}, K_{D_e} \in \mathbb{R}^{8\times 8}$ positive definite symmetric matrices to ensure the controller stability (see Appendix A) and $e_{s_e} = s_{e_D} - s_e$ and $\dot{e}_{s_e} = -\dot{s}_e$ the position and velocity errors, respectively. The selection matrix P partitions the 6-D space into torques that address the motion control task and torques that address the force control task. It is opportunely rotated in the case of image space control (motion) and Cartesian space control (interaction force).

To conclude, the parallel force/vision controller is

$$\tau_{hy} = B(q)u_v + h(q, \dot{q}) + \tau_{fc}. \tag{22}$$

The two control actions are decoupled by the orthogonal projections $P_2 = Ad_{T_s^c} P$ and $I_6 - P$ and inherit the error dynamics and stability analyses of the individual controllers on their respective subspaces [51]. Notice that when $\lambda = 0$, the controller (22) is a pure IBVS controller.

4.3. Teleoperation Controller

A human operator can enter the control loop using a 3-D haptic controller device designed to provide high-fidelity kinesthetic feedback.

In the examined case, the device has to control the drone's manipulator sending set-points to the already analyzed hybrid controller (see Fig. 2). It is possible to map the movement along the device $x_{\mathcal{F}}, y_{\mathcal{F}}$-axes as the desired E–E position in the image space controlling the sliding on the surface, or map the movement along its $z_{\mathcal{F}}$-axis in force set-point while pushing on it.

The movements are mapped only if the user presses the central button, while the haptic feedback is always present. The haptic feedback makes the user an active part of the task. The user can feel the wall resistance while pushing and a repulsive effect that forces him to stay inside the camera field of view (FoW) as explained in the following.

In the control loop, the handle's position $p_f = \begin{bmatrix} p_{f_x} & p_{f_y} & p_{f_z} \end{bmatrix}^\top \in \mathbb{R}^3$ is used to retrieve the handle velocity $\dot{p}_f \in \mathbb{R}^3$ by time deriving two successive position measures. As in Section 4.1.3, these quantities are computed in $\mathcal{C}$ and translated into the image space. The haptic device image space velocity $\dot{s}_f$ is then added to the E–E IBVS setpoint through

$$\dot{s}_{e_{D_N}} = \dot{s}_{e_D} + \dot{s}_f, \quad s_{e_{D_N}} = s_{e_D} + \dot{s}_f t_s, \tag{23}$$

with $t_s > 0$ the controller sample time.

Additionally, the third haptic device linear DoF is used to reconstruct a force set-point for the interaction controller. The E–E force reference is recursively updated by employing the concept of the admittance filter [55]:

$$f_{des} = f_{des} + m_{nf}\ddot{p}_{f_z} + d_{nf}\dot{p}_{f_z} + k_{nf}p_{f_z} \in \mathbb{R}. \tag{24}$$

The Falcon position p_{f_z} and the velocity $\dot{p}_{f_z}$ are reconstructed from the user movements and used to increase or decrease the desired force reference f_{des}. $m_{nf}, d_{nf}, k_{nf} \in \mathbb{R}$ are positive definite gains to tune the filter defining a mass-damper-spring behavior. In the examined case, no acceleration $\ddot{p}_{f_z}$ is considered ($m_{nf} = 0$).

4.3.1. Haptic Feedback

While imposing a new set-point, the user can feel different forces in feedback to the haptic device. While pushing a scaled force

$$F_{rep_N} = \sigma f_{des} \tag{25}$$

is applied to feel the sense of touch. In the meantime, other repulsive forces are computed to guide the use in the task execution. A non-experienced user can fail the task execution going for example outside the field of view, or by occluding the vision and losing the only available source of information. Four forces are computed by measuring the minimum distance between the E–E and the camera borders, and the same concerning the tag. These forces have an exponential shape designed to be null at the image center and the maximum value on the image space borders

$$F_{rep_i} = (-1)^i \rho e^{-\varepsilon d_{min}(i)} d_{min}(i)^{-\beta}, \tag{26}$$

where $\rho, \varepsilon, \beta > 0$ are constants that characterize the evolution of the exponential function and $d_{min}(i) > 0$ is the distance from the i-th border $(i = 1, ..., 4)$.

Equation (26) is the vector stacking two opposite couples of vertical or horizontal repulsive forces. The two horizontal forces, as the vertical ones, are summed resulting in null at the center of the camera field and increasing while moving along the 2-D space. Combining (25) and (26) yields

$$F_{rep} = \begin{bmatrix} F_{rep_1} + F_{rep_2} & F_{rep_3} + F_{rep_4} & F_{rep_N} \end{bmatrix}^\top \in \mathbb{R}^3. \tag{27}$$

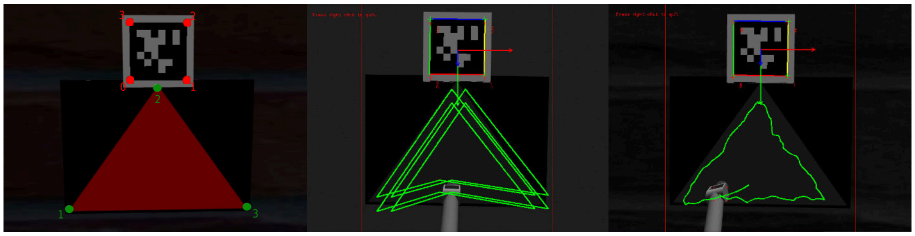
The closer will be the user to one of those bounds and the more interaction force will be commanded, the greater will be the repulsive forces. The idea is to help the user by preventing the constraints violations during the task execution.

5. Case Studies

The case studies focus on a push–and–slide task, comparing autonomous execution with a teleoperated modality. The task is performed in a virtual environment using a physical simulator described in Section 5.1. This environment (see Fig. 7) consists of a concrete surface marked with an AprilTag [56] and a triangular shape, which serves as a guide for the push–and–slide task execution. The visual sensor detects the AprilTag and adjusts the relative position between the aerial platform and the surface. After adjusting the interaction force, the three points of interest highlighted in green on the triangular shape are tracked by visual control.

In the autonomous case, interpolation evaluates the set-points in image space. The task is completed when the E–E returns to its starting position without losing contact with the surface (see Fig.7.b).

In the teleoperated mode, the user must connect the three green dots in Fig. 7.a through the Novint Falcon haptic device [57] in different control modalities and/or control the desired interaction force. An experimental campaign will evaluate different metrics to determine the most suitable control mode for completing these semi-autonomous tasks (see Fig.7.c).



(a) **(b)** **(c)**

Figure 7: Simulated scenario from the camera point-of-view (PoV). (a) The inspection surface is detected through an AprilTag [56] extracting four image features (red dots). The task involves tracking the triangular shapes interpolating the three green dots while interacting. Comparison is considered between the autonomous control mode (b) and the teleoperated control mode (c). The vertical red lines represent the E–E out-of-bound conditions along x_C.

5.1. Simulation Setup

The proposed aerial robotic manipulation system has been integrated into the Gazebo simulation environment whose choice is justified in Section 7.1. The dynamical parameters presented in Table 1 are directly retrieved from the UAM CAD model, while the camera's intrinsic parameters are directly retrieved by the proper ROS topic after the calibration procedure:

$$K = \begin{bmatrix} 476.703 & 0 & 400.5 \\ 0.0 & 476.703 & 300.5 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}. \tag{28}$$

For further details and insights on the presented platform and its mathematical modeling, please visit the GitHub pages of the project[1][2].

The tests are performed on a standard PC, with an *i7-8750H* CPU and 32GB of RAM. The OS chosen is Ubuntu, with its release *22.04 LTS Jammy Jellyfish*. The communication between the software is managed using ROS *Noetic Ninjemys* running in a docker environment. The sampling time is set to $t_s = 0.01$ s, and the controller gains displayed in Table 2 have been tuned to avoid overshoots and reach the control goal smoothly. The parameters in Table 3 are heuristically tuned in a preliminary phase of the experimental campaign to accommodate the feedback on the device usage experience. Visp [58], an open-source visual servoing platform library, is employed to detect the tag and extract the visual features.

The goal is to evaluate the performance of a push–and–slide task executed both fully autonomously (Section 5.2) and via human user semi-autonomous teleoperation (Section 5.3). Our primary aim is to isolate the effect of each

[1] https://github.com/prisma-lab/ndt2_arm_control.git

[2] https://github.com/prisma-lab/ndt2_control.git

Table 1: Dynamical parameters.

Parameter	Value
mass (m)	6.3833 kg
Inertia (I)	diag$\{[0.29163,\ 0.29163,\ 0.3527]\}$ kg m^2
UAV's arm length (l)	0.255 m
UAV's arm height (h)	0.03 m

Table 2: Control gains.

Gains	Value
$K_{P_{img}}$	$15I_{8\times8}$
$K_{D_{img}}$	$20I_{8\times8}$
$K_{I_{img}}$	$1.5I_{8\times8}$
K_{P_e}	$400I_{8\times8}$
K_{D_e}	$40I_{8\times8}$
K_{P_F}	$1.5I_{6\times6}$
K_{I_F}	$4.5I_{6\times6}$

Table 3: Haptic feedback gains.

Gains	Value
d_{nf}	0.045
k_{nf}	0.01
σ	0.85

control and feedback modality on task performance and determine which combination leads to the most effective task execution. To achieve this, we created a simulated environment in which the user performs a push–and–slide task.

The envisioned task consists of the following steps:

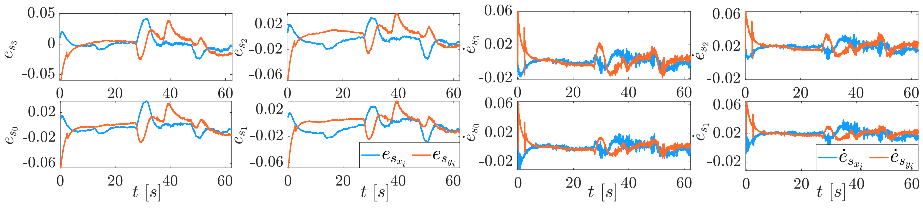
1. The UAM approaches the wall employing visual servoing (Section 4.1.3).
2. The arm pushes on the wall employing direct force control (Section 4.2).
3. The arm slides along the wall surface to ideally track a triangular profile with its E–E, while the drone keeps the distance to the detected tag. The arm must never occlude the tag to safely accomplish the task.

In the teleoperation mode, the human can command the arm to push on the wall (P), slide on the wall surface (S), or both (PS). Meanwhile, (s)he can receive mixed feedback about visual constraints acting on the system, i.e., occlusions of the tag and field of view, and about the actual force exchanged (VF). We compared the proposed control/feedback modalities with the full autonomous task execution (A), which is discussed next. The analysis results are reported in the accompanying video: https://youtu.be/0AxjhZ2wbbE.

5.2. Case 1: Autonomous Execution

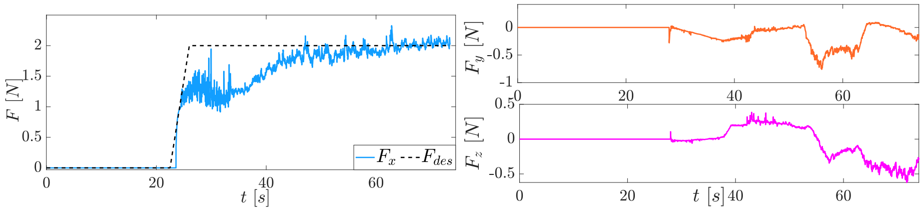
Results from fully autonomous tests will validate the system assessing the effectiveness of the proposed framework and serve as a benchmark for the upcoming tests involving human operators.

In the initial approach phase, the drone is guided by a image-space control strategy. The six controllable degrees of freedom are directed to nullify the position error in the image plane, which corresponds to a position and attitude error in Cartesian space. As illustrated in Fig. 8, despite the relatively low gains



(a) **(b)**

Figure 8: Autonomous execution. UAV's image space position errors (a) and velocity errors (b) in normalized pixel coordinates. The errors are related to each AprilTag detected corner. The tag corner order here and in the following figures is the same as shown in Fig. 7.



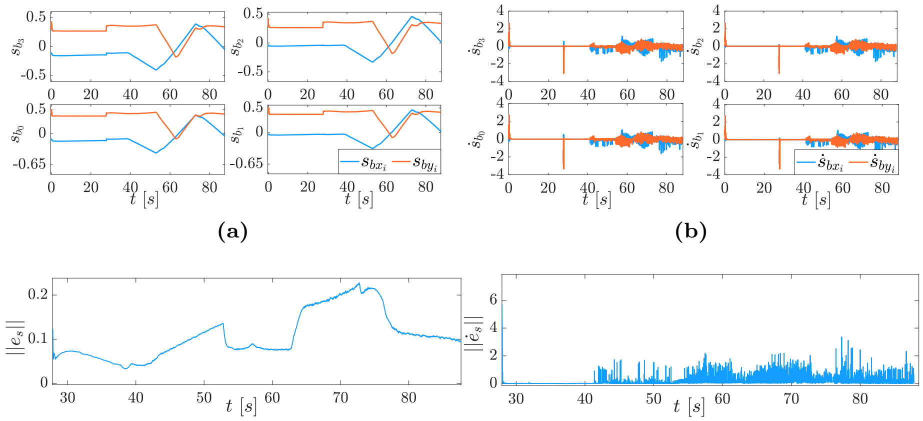
(a) **(b)**

Figure 9: Autonomous execution. (a) Interaction force error along approaching direction; (b) Sliding interaction forces.

in (8), the drone effectively cancels the position error relative to the tag and keeps it during the inspection, despite the external disturbances due to interaction. These disturbances manifest as chattering in the image plane velocity in the second half of Fig. 8.b.

While maintaining the relative pose as a steady state, the E–E is guided into contact using (22). The force sensor measures the interaction force and equation (18) selects the appropriate axis for implementing the parallel controller. An initial pushing step is required to regulate the desired force on the surfaces while the image-space reference trajectories are computed interpolating the three vertices of the triangular shape on the surface.

The interaction task is accomplished by regulating the desired force at a value of $f_{des} = 2$ N. This precise control is achieved without overshooting, thanks to the careful selection of gains in Table 2, which ensures a stable and responsive system as shown in Fig. 9. During the task execution, it is observed that the position error increases notably during sliding phases as can be appreciated in Fig. 10.c: the control system needs to prioritize one of the tasks, either force or position control, at any given time, leading to a trade-off. Furthermore, unmodeled friction between the touching parts introduces additional disturbances to the manipulator's performance. Despite these challenges, the overall error remains impressively low, which is particularly significant given that the quantities shown are measured in normalized pixel coordinates. This



(a) **(b)**

(c) **(d)**

Figure 10: Autonomous execution. At $t = 28$ s, the UAV establishes stable contact with the work surface while keeping the distance to the tag. The E–E virtual corners position (a) and velocities (b) are depicted while (c)-(d) show the position and velocity norm errors in the image space. The quantities are expressed in normalized pixel coordinates.

level of accuracy underscores the robustness of the control strategy in handling unmodeled dynamics and prioritizing task objectives effectively.

5.3. Case 2: Teleoperated Execution

In the teleoperated case study, the control law and gains remain consistent with the previous case study, but a key difference is the inclusion of the human operator in the control loop. The operator manages the generation of set-points in the image plane and the desired force applied to the surface while receiving feedback on the repulsive forces calculated by the Falcon, as previously explained. These results will be compared with the previous study to discuss the overall experimental campaign.

The benefits of teleoperation become apparent in Fig. 11, which presents the UAV image space regulation results. Here, the position error is lower than in the previous case, even during interaction. The operator's skill in balancing motion and force commands reduces chattering in velocity errors, minimizing external disturbances on the flying platform during interaction control.

Figure 12 shows the interaction force error, where the human operator generates the desired interaction force command. The force measurements exhibit an oscillatory behavior, which is canceled at the steady-state during the sliding phase.

In the teleoperated scenario, the triangular shape tracking demonstrates higher precision in positioning and reduced oscillation on the velocities, as illustrated in Fig. 13. The Novint Falcon's user-friendly design allows users of all skill levels to participate in the control loop and successfully complete complex inspection tasks.

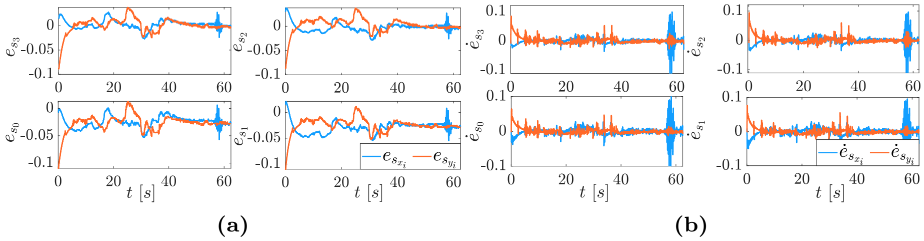


Figure 11: Teleoperated execution. UAV's image space position errors (a) and velocity errors (b) in normalized pixel coordinates. The errors are related to each tag-detected corner.

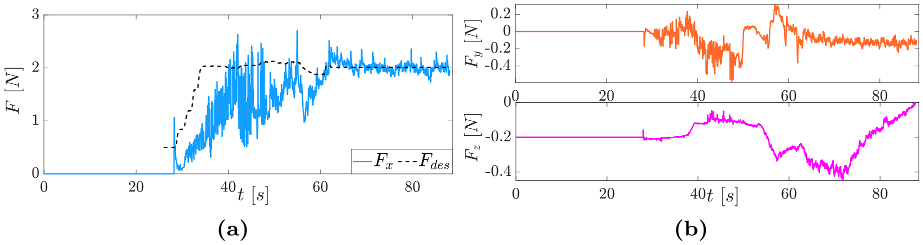


Figure 12: Teleoperated execution. (a) Interaction force error along the approaching direction; (b) Sliding interaction forces.

Finally, Fig. 14 displays the E–E distances from the camera borders during task execution and the corresponding repulsive forces computed to provide a tactile sense while inspecting. These results are relative to one control modality (PSVF) completed by one of the users involved in the experimental campaign.

6. Human Subjects Study

As mentioned above, semi-autonomous teleoperation can be accomplished in several ways by mixing user inputs, autonomous control, and feedback modalities. In our case, we are interested in evaluating if some of the envisioned user input/control/feedback combinations lead to a measurable difference in the task execution. The scope of this study is to establish how the considered factors are affecting the envisioned task to concentrate future design efforts on them. We thus envisioned three user inputs/control modalities that are described in the following.

- P: the user is in charge of only regulating the pushing force via the haptic device, the rest is autonomous.
- S: the user is in charge of only regulating the sliding motion via the haptic device, the rest is autonomous.

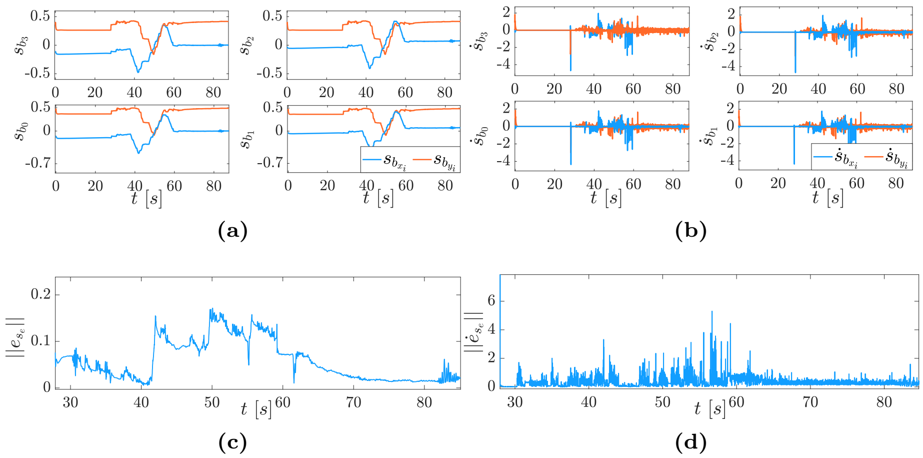


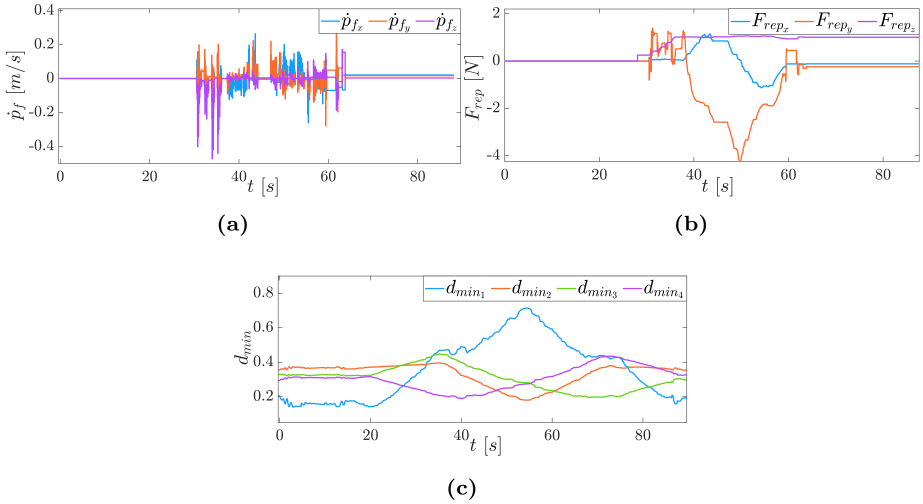
Figure 13: Teleoperated execution. At $t = 28$ s, the UAV establishes stable contact with the work surface while keeping the distance to the tag. The E–E virtual corners position (a) and velocities (b) are depicted while (c)-(d) show the position and velocity norm errors in the image space. The quantities are expressed in normalized pixel coordinates.

- PS: the user is in charge of controlling both the push force and the sliding motion via the haptic device.

We mix the three control modes P, S, and PS, with the presence/absence of the feedback modality VF, computed as in described in Section 4.3. Combining them, we obtain a total of six trials to be performed by each subject. We recruited a population of 20 human subjects (18 male, 2 female) familiar with robotics; the group included both expert and non-expert users who had never used our specific system before. The sequence of trials has been randomized across subjects to avoid bias in the results caused by the learning curve. We evaluated both quantitative (data recorded during the experiments) and qualitative (user preferences expressed after each trial) metrics. The qualitative evaluation consisted of a user answering the following two questions for each tested modality:

1. How satisfied are you with the overall user interface and ease of use of the application?
2. How satisfied are you with the speed and responsiveness of the application?

The answers were collected via a 5-point Likert scale with the following options: very satisfied, satisfied, neutral, unsatisfied, and very unsatisfied. In the following, we report only the statistically significant results obtained by means of ANOVA and running multiple comparisons when statistically significant changes were detected. Means and standard deviations of the data reporting interesting results are shown in the bar plots contained in Fig. 15 and summarized in Table 4 where the control modalities comparison is highlighted by the relative means and standard deviations $(\bar{(\cdot)} \pm \sigma(\cdot))$.



(a)

(b)

(c)

Figure 14: Teleoperated execution. (a) Novint falcon velocity in $\mathcal{F}$. (b-c) Novint falcon repulsive forces and E–E distance to the camera borders according to (26)-(27).

For the completion time ($\mathcal{T}$), there is a statistically relevant difference ($p < 0.01$, $F = 8.09$) between tests in which the sliding is autonomous (P, PVF) and tests in which it is not (PS, PSVF, SVF). This is expected as manual sliding is more challenging and requires more time to be accomplished. This is also confirmed by the notable trend showing less completion time when only pushing is manual.

Next, we look into subjective evaluations, more specifically the ease of use ($\mathcal{E}$) evaluated as answer to the first of the two questions given above, and discovered a statistically significant difference ($p < 0.01$, $F = 8.63$) between the same modalities, i.e., tests in which the sliding is autonomous (P, PVF) and tests in which it is not (PS, PSVF, S, SVF). On the same line, this is a clear evidence that subjects evaluated as easier tests in which sliding is autonomous.

We also evaluated the metric $\mathcal{V}$ which contains information about potential occlusions via the mean of the minimum distance d (in the image space) between the E–E position and the $i-$th image border, in formula

$$\mathcal{V} = \frac{1}{\mathcal{T}} \int_0^{\mathcal{T}} \min_{i=1,\ldots,4} (d_i) \, \mathrm{d}t. \tag{29}$$

We again discovered a statistically significant ($p < 0.05$, $F = 2.49$) difference between the same modalities, i.e., tests in which the sliding is autonomous (P, PVF) and tests in which it is not (PS, PSVF, S, SVF). It is worth mentioning however that in this case although differences are statistically significant, multiple comparisons return negative results.

Surprisingly, the autonomous sliding control feature also significantly affects

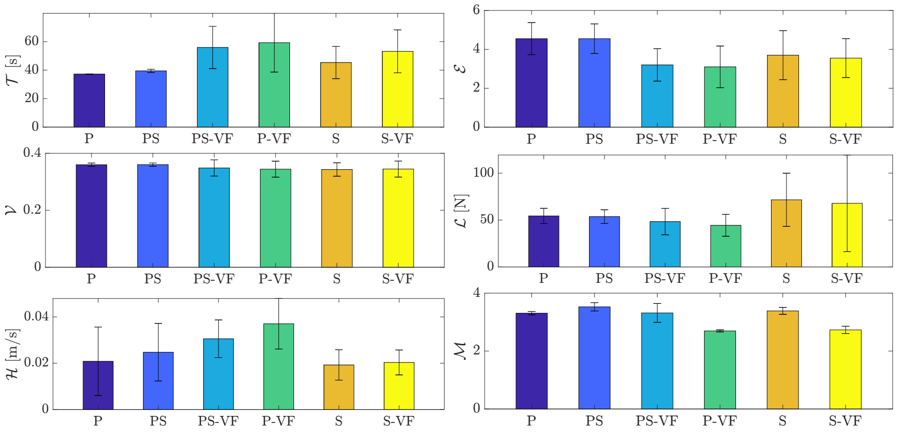


Figure 15: Statistically significant metrics plot. $\mathcal{T}$ is the completion time, $\mathcal{E}$ is the ease of use, $\mathcal{V}$ is the minimum E–E image-space distance to the image border, $\mathcal{L}$ is the force error norm, $\mathcal{H}$ is the maximum haptic device commanded velocity and $\mathcal{M}$ is the operational smoothness.

the mean of the pushing force error norm ($\mathcal{L}$) evaluated along the trajectory as

$$\mathcal{L} = \frac{1}{\mathcal{T}} \int_0^{\mathcal{T}} ||F_d - F||^2 \, \mathrm{d}t. \tag{30}$$

A statistically significant difference ($p < 0.01$, $F = 3.58$) in this metric means that the manual regulation of the sliding makes more/less difficult the regulation of the pushing force. This time the difference is between the pairs PSVF-S and PSVF-SVF only. This means that adding force feedback during the manual regulation of pushing and sliding leads to improvements in terms of force error compared to the autonomous regulation of pushing force without feedback.

Finally, evaluating the max two norm of the haptic device velocity v_h along the trajectory, i.e.,

$$\mathcal{H} = \max_t ||v_h(t)||^2, \tag{31}$$

a statistically significant difference ($p < 0.01$, $F = 9.39$) can be found between the pairs PSVF-PVF, PSVF-S, PSVF-SVF, PSVF-P, and the pairs PS-S. These differences testify that the simultaneous manual regulation of pushing and sliding leads in general to higher commanded velocities compared to the autonomous regulation of pushing or sliding alone.

An additional statistically significant difference ($p < 0.001$, $F = 93.85$) has been found in the operational smoothness metric defined as:

$$\mathcal{M} = \frac{1}{\mathcal{T}} \int_0^{\mathcal{T}} ||a(t)||^2 \, \mathrm{d}t, \tag{32}$$

where a is the arm E-E acceleration during the task execution. The difference between (P, PS) and (PVF, SVF, PSVF) tells us that the addition of haptic

Table 4: Statistical analysis results.

Subjects	20 (2 females, 18 males)
Task	Control the teleoperation system to accomplish push–and–slide with the UAM E-E.
Conditions	**Teleoperation** P (teleoperated pushing), S (teleoperated sliding), PS (teleoperated pushing and sliding) **Feedback** VF (haptic feedback), $\cancel{\text{VF}}$ (no haptic feedback)[3]

Statistical analysis (significant p values, means and standard deviations)	
Completion time, $\mathcal{T}$ **(**$p < 0.01$)	
(P, PVF) vs. (SVF, PS, PSVF)	38.36 ± 1.58 vs. 56.15 ± 3.05
Ease of use, $\mathcal{E}$ ($p < 0.01$)	
(P, PVF) vs. (S, SVF, PS, PSVF)	4.55 ± 0.0 vs. 3.38 ± 0.28
Minimum distance from image border, $\mathcal{V}$ ($p < 0.05$)	
(P, PVF) vs. (S, SVF, PS, PSVF)	$0.36 \pm 1.75e-4$ vs. 0.34 ± 0.0023
Force error norm, $\mathcal{L}$ ($p < 0.01$)	
PSVF vs. (SVF, S)	58.01 ± 19.24 vs. 56.10 ± 8.26
Maximum haptic device velocity, $\mathcal{H}$ ($p < 0.01$)	
PSVF vs. (P, PVF, SVF, S)	0.037 ± 0.0 vs. 0.021 ± 0.0024
Operational smoothness, $\mathcal{M}$ ($p < 0.01$)	
(P, PS) vs. (PVF, SVF, PSVF)	3.31 ± 0.0065 vs. 2.98 ± 0.47

feedback is beneficial in terms of operational smoothness as it leads to lower acceleration values.

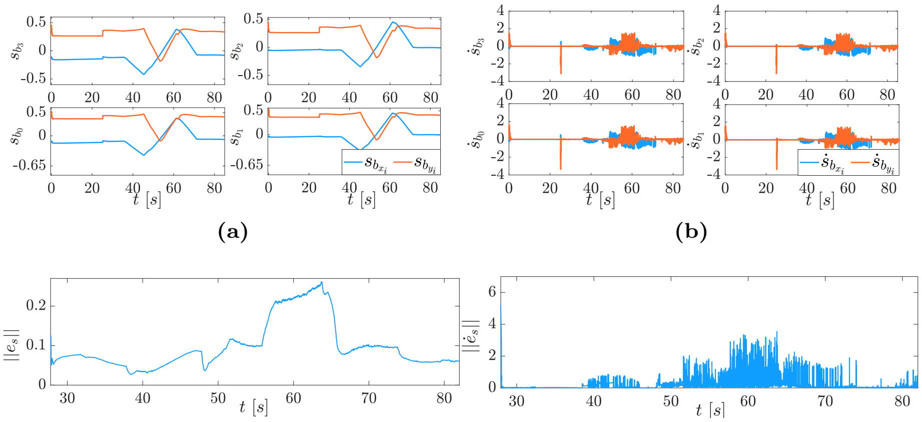
In addition, it is important to mention that the detachment between the E–E and the inspection point only appears in one of the considered cases when pushing was set to manual.

6.1. Additional Comparison

An additional comparison is proposed between the autonomous and the teleoperated case study. Subsequent to the experimental campaign, we calculated the average task execution time in the PSVF control modality and consequently adapted the autonomous push–and–slide task timing to re-assess the performance of the autonomous execution. It was verified that the average execution time was $t_M = 59.4$ s including both the pushing and sliding teleoperated task.

We repeated the fully autonomous case study adding different sources of noise and disturbances into the system to prove the controller's robustness and to get closer to the real-world scenario. In particular, Gaussian noises are applied to both camera and force sensor feedback. Additional noises are introduced in

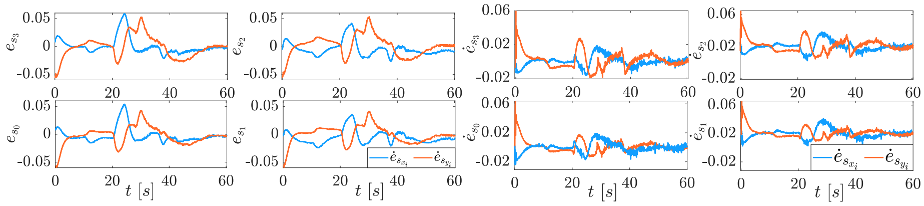
[3] For simplicity, if not stated explicitly, the control mode includes no haptic feedback (e.g., PS=PS$\cancel{\text{VF}}$).



(a) **(b)**

(c) **(d)**

Figure 16: Autonomous task execution matching the teleoperated task timing with added noises and external disturbances: E-E image space normalized position and velocity (a)-(b) and the relative norm errors (c)-(d).



(a) **(b)**

Figure 17: Autonomous task execution matching the teleoperated task timing with added noises and external disturbances: UAM image space position (a) and velocity errors (b).

the Gazebo model IMU measurements and a plugin simulating the wind effect in the physics engine is introduced. To conclude we added uncertainty on the arm dynamical parameters up to 50%.

Compared to the autonomous execution described in Section 5.2, the drone exhibits bigger oscillations in the velocity error within the image space due to the faster image space references, as shown in Fig.16.b. These errors are not fully canceled at a steady state, primarily due to the continuous presence of external disturbances (e.g., simulated wind) and measurement noise (e.g., Gaussian noise on the IMU and sensors). Nevertheless, these low persistent errors can be canceled easily providing an additional controller tuning step.

In comparison to the fully teleoperated case (PSVF) previously analyzed (see Section 5.3), this method achieves lower commanded velocities for the manipulator arm, indicating a more efficient operation. Regarding the remaining performance indexes, the results of the two approaches are similarly comparable.

7. Discussion

7.1. Simulated Environment

To prove the system's performances, we implemented our control framework in Gazebo, a realistic simulator with embedded physics engines. This is a powerful tool that allows direct communication with ROS/ROS2 nodes. It is fully customizable permitting the import of custom robots as models.

Unlike numerical simulators (e.g., MATLAB), Gazebo eliminates the need to implement the system's mathematical model, focusing solely on the controller design. This approach ensures the simulator's dynamics differ from the controller's assumptions, testing its stability under realistic conditions. Additionally, it provides a safe environment to validate the framework and enables training for user studies without risking hardware damage. As cons, the shadows' absence makes the guided task more challenging removing depth references for the user. In a real application, the shadows should help the users to identify the distances directly from the camera PoV making the teleoperated task easier.

Moreover, the Gazebo simulator allows the definition of external disturbances on the simulated models and noises on the sensors' plugins to get closer to the real scenario. The controller is proven to be robust to both Gaussian noises on IMU and sensors and wind disturbance as already discussed in Section 6.1.

7.2. Statistical Analysis

As previously explained, the simulation enables us to quickly and safely evaluate different control and feedback modalities with multiple users without the risk of damaging a physical system. Here we delve into the discussion of the results showcased in the previous section analysing their meaning and relevance. We addressed the gap evidenced by the literature review about the scarcity of haptic shared control teleoperation methods for interaction-based tasks with UAMs. We have shown the possibility of developing an effective framework making use of visual servoing and hybrid force–motion control to satisfy the stringent task requirements from industrial tasks. These are the positioning of the drone, exertion of pushing force, and implementation of sliding behavior, which are all satisfactorily accomplished with minimal errors (as shown in Section 5.2). To the best of authors' knowledge this was the first time a parallel force/vision controller with precise force tracking capabilities and positioning in the image space has been applied to a complex omnidirectional tilting UAM with an articulated arm.

Besides autonomous executions, the proposed control framework has been used in teleoperation mode to test the possibility of including a human in the loop. As can be seen from the experiments, performance in teleoperation mode depends on the user performing the task, but results have shown satisfactory behavior demonstrating the effective usability of the framework (as shown in Sec. 5.3). Lastly, as the system and the task allow multiple aspects to be autonomous or manually operated, we isolated teleoperation features into factors and ran a human-subject study involving 20 participants. From these we can assert that most of the statistically significant differences can be evidenced when the sliding feature is autonomous, denoting it as the most difficult to be accomplished by the user. Surprisingly, the presence of haptic feedback/guidance has led to no significant differences in performance. The most plausible reason to explain this is that 1) contact force information is not useful as force control always keeps the E–E pressing against the wall, and 2) force guidance keeping the E–E within the image influences a feature that the human can easily control. In summary, it does not add information about the status of the system other than the ones that humans can already monitor/control. Future works may concentrate on tests performed with a real platform on a more challenging scenario, for instance, the inspection of curved surfaces (e.g. pipes).

7.3. Limitations and Errors Sources

The presented application required the resolution of several limitations highlighted in the literature review proposing innovations on both the software and hardware sides. Some limitations and sources of errors, we tried to improve, will be analyzed below.

7.3.1. Camera Shake and Platform Dynamics

Standard flat UAVs often rely on the virtual camera frame assumption [25, 26, 27] as already mentioned. It is impossible for them to directly track the attitude in the image space and avoid disturbances given to the camera shakes and oscillation. An initial solution could be the use of ad hoc gimbals including more complexity and controllable DoFs. The proposed system benefits from the omnidirectional capabilities of the aerial platform, which allow direct attitude adjustments to maintain the target feature within the camera's field of view (FoV). This feature reduces the reliance on mechanical stabilization systems such as actuated gimbals. However, residual platform oscillations can still affect the camera feed. To mitigate this, the control architecture incorporates an attitude stabilization loop, which minimizes vibrations and ensures smoother visual feedback.

7.3.2. Field of View Limitations

The limited FoV of the onboard camera is a known constraint. To address this, an image-based visual servoing (IBVS) approach was employed, dynamically adjusting the UAV's position and orientation to ensure continuous tracking of the visual target. Predictive feature tracking algorithms further enhance system robustness by compensating for scenarios where the target briefly exits the FoV. This combination of hardware and software strategies minimizes disruptions in visual feedback.

Moreover, an additional safety layer is given by the haptic device itself. It continuously computes repulsive forces guiding the user to both remain in the camera FoV and avoid marker occlusion. These repulsive forces physically constrain the operator to guide the E-E in safe camera space positions.

7.3.3. Lighting Conditions and the Role of Shadows

Lighting variations in industrial environments pose another significant challenge. The visual feedback system employs adaptive image preprocessing techniques, such as histogram equalization and filtering, to improve the visibility of features under varying illumination levels. A noteworthy limitation of the current system, however, is the absence of shadows in the simulated environment. While shadows can introduce noise in some applications, they often provide valuable depth and orientation cues for operators during teleoperation tasks. Their absence in the simulation likely increased task difficulty for human participants, potentially affecting their performance during sliding tasks. In real scenarios, shadows can aid in depth perception, potentially reducing such issues. Nevertheless, the simulation may be slightly conservative in this respect, as our model includes assumptions of uniform friction and wind conditions, which may not fully account for every environmental factor in practical deployments.

7.3.4. Future Directions

Despite the measures implemented, the stability of visual feedback remains an area for further development. Future improvements could include integrating complementary sensor modalities, such as LiDAR or depth cameras, to provide additional robustness against environmental variations. Furthermore, leveraging advanced computer vision techniques, such as deep learning-based feature recognition, could enhance system adaptability to rapidly changing environments.

By addressing these limitations, the system can be further optimized for real-world deployment in dynamic industrial scenarios, improving both autonomous and teleoperated task performance.

8. Conclusions

In the paper, we developed a semi-autonomous teleoperation control architecture to perform push–and–slide inspection tasks with a UAM, proposing a parallel force/vision controller working alongside a human operator to ease task accomplishment. We showcased the performance of our control architecture during teleoperated and fully autonomous task execution and evaluated notable metrics from a human-subject study performed with 20 participants, showing statistically significant differences enabled by the activation of the autonomous sliding feature. These results provide us with guidelines for the future design of next-generation shared control teleoperated UAM systems for interaction tasks.

8.1. Future works

Moving forward, the methodology could be extended to enable effective inspection and interaction with non-vertical surfaces that are either convex or non-convex, which are prevalent in real-world industrial environments. Such tasks require advanced control strategies to maintain stability and precision during interaction, particularly when dealing with surfaces that impose varying normal directions or present complex geometries. This task will require ad rolling probes and need to re-locate the camera sensor in a new configuration mimicking the setup in [59].

One promising direction is the exploitation of the redundancy of the UAM system to improve maneuverability and adaptability in these challenging scenarios. This can be achieved by planning in the null space of the UAM's configuration, allowing the system to prioritize interaction-related objectives while simultaneously optimizing secondary tasks, such as collision avoidance or minimization of energy consumption.

Additionally, the integration of real-time environment perception through advanced sensing modalities, such as lidar or stereo cameras, combined with adaptive motion planning, could enhance the system's capability to navigate and interact autonomously with complex geometries. This would be particularly valuable for extending the proposed framework to autonomous inspection tasks on irregular structures like turbine blades, curved pipelines, or other non-planar surfaces.

Finally, incorporating machine learning techniques for adaptive control tuning based on surface characteristics or operator preferences could further enhance the system's usability and efficiency. This would enable a tailored response for different industrial scenarios, ensuring that the UAM performs optimally regardless of the task complexity or environmental conditions.

Appendix A. IBVS Stability Analysis

To analyze the stability of the controller in Section 4, we define two Lyapunov functions $V_1(e_{s_e}, \dot{e}_{s_e})$ and $V_2(e_F, e_{I_F})$ considering that the parallel controller is decoupled by the selection matrix P.

We want to recall that: $e_{s_e} = s_{e_D} - s_e$ and $\dot{e}_{s_e} = \dot{s}_{e_D} - \dot{s}_e$ represent the position and velocity error in the image space; $e_F = f(t) - f_{des}(t)$ and $e_{I_F} = \int e_F \, \mathrm{d}t$ are the interaction force error and its integral; finally, $K_{(\cdot)}$ is a symmetric positive definite weighting matrix. It is possible to prove the stability of the two controllers individually, neglecting the presence of the selection matrices [51].

Appendix A.1. Lyapunov Function analysis - IBVS stability

$$
\begin{aligned}
V_1 &= \frac{1}{2} e_{s_e}^\top K_{P_e} e_{s_e} + \frac{1}{2} \dot{e}_{s_e}^\top \dot{e}_{s_e} \\
\dot{V}_1 &= e_s^\top K_{P_e} \dot{e}_s + \dot{e}_s^\top \ddot{e}_s, \\
\ddot{e}_s &= \ddot{s}_D - \ddot{s}, \text{ and } \ddot{s}_D = \mathbf{0}_8, \; \ddot{s} = K_{P_e} e_s + K_{D_e} \dot{e}_s, \\
\dot{V}_1 &= e_s^\top K_{P_e} \dot{e}_s + \dot{e}_s^\top (K_{P_e} e_s + K_{D_e} \dot{e}_s), \\
\dot{V}_1 &= \underbrace{e_s^\top K_{P_e} \dot{e}_s - \dot{e}_s^\top K_{P_e} e_s}_{=0} \underbrace{- \dot{e}_s^\top K_{D_e} \dot{e}_s}_{\leq 0 \text{ always}} .
\end{aligned}
\tag{A.1}
$$

Appendix A.1.1. Discussion

We compute the time derivative of $V_1 = \frac{1}{2} e_{s_e}^\top K_{P_e} e_{s_e} + \frac{1}{2} \dot{e}_{s_e}^\top \dot{e}_{s_e}$:

$$\dot{V}_1 = e_s^\top K_{P_e} \dot{e}_s + \dot{e}_s^\top \ddot{e}_s. \tag{A.2}$$

Here, $\ddot{e}_s$ is the second derivative of the error, which is defined as:

$$\ddot{e}_s = \ddot{s}_D - \ddot{s}, \tag{A.3}$$

where $\ddot{s}_D = \mathbf{0}_8$ represents the desired acceleration, and $\ddot{s}$ is the actual acceleration of the system, retrieved substituting (20) and (21) in the manipulator dynamics. From the inverse dynamic control algorithm and neglecting the selection matrices P and $P2$ as already explained:

$$\ddot{q} = u_v \;\; \rightarrow \;\; \ddot{s} = K_{P_e} e_s + K_{D_e} \dot{e}_s.$$

Substituting this into (A.2) and (A.3), we get:

$$\dot{V}_1 = e_s^\top K_{P_e} \dot{e}_s - \dot{e}_s^\top (K_{P_e} e_s + K_{D_e} \dot{e}_s). \tag{A.4}$$

Expanding this expression yields:

$$\dot{V}_1 = e_s^\top K_{P_e} \dot{e}_s - \dot{e}_s^\top K_{P_e} e_s - \dot{e}_s^\top K_{D_e} \dot{e}_s = -\dot{e}_s^\top K_{D_e} \dot{e}_s. \tag{A.5}$$

The term in the derivative is always non-positive, since K_{D_e} is a positive-definite matrix, and the term $\dot{e}_s^\top K_{D_e} \dot{e}_s$ represents a quadratic form, always positive.

Thus, the overall derivative of the Lyapunov function $\dot{V}_1$ is guaranteed to be less than or equal to zero. These conditions ensure that the system's dynamics are stable, with the Lyapunov function decreasing over time.

In the same hyperplane, we have $\ddot{e}_s = \mathbf{0}_8 \;\; \rightarrow \;\; \ddot{s} = \mathbf{0}_8$ considering $\ddot{s}_D = \mathbf{0}_8$. For the previous proof $\dot{e}_s \rightarrow \mathbf{0}_8$ and the closed loop system $\mathbf{0}_8 = K_P e_s$ converges to zero if and only if $e_s \rightarrow \mathbf{0}_8$. Thus the asymptotical stability is then proved.

References

[1] G. Aiello, F. Hopps, D. Santisi, M. Venticinque, The employment of unmanned aerial vehicles for analyzing and mitigating disaster risks in industrial sites, IEEE Transactions on Engineering Management 67 (3) (2020) 519–530.

[2] F. Ruggiero, V. Lippiello, A. Ollero, Aerial manipulation: A literature review, IEEE Robotics and Automation Letters 3 (3) (2018) 1957–1964.

[3] M. Javaid, I. H. Khan, R. P. Singh, S. Rab, R. Suman, Exploring contributions of drones towards industry 4.0, Industrial Robot: The international journal of robotics research and application 49 (3) (2022) 476–490.

[4] J. Cacace, A. Finzi, V. Lippiello, G. Loianno, D. Sanzone, Aerial service vehicles for industrial inspection: Task decomposition and plan execution, in: M. Ali, T. Bosse, K. V. Hindriks, M. Hoogendoorn, C. M. Jonker, J. Treur (Eds.), Recent Trends in Applied Artificial Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 302–311.

[5] M. Selvaggio, M. Cognetti, S. Nikolaidis, S. Ivaldi, B. Siciliano, Autonomy in physical human-robot interaction: A brief survey, IEEE Robotics and Automation Letters 6 (4) (2021) 7989–7996.

[6] M. Selvaggio, F. Abi-Farraj, C. Pacchierotti, P. R. Giordano, B. Siciliano, Haptic-based shared-control methods for a dual-arm system, IEEE Robotics and Automation Letters 3 (4) (2018) 4249–4256.

[7] S. Marcellini, S. D'Angelo, A. De Crescenzo, M. Marolla, V. Lippiello, B. Siciliano, Development of a semi-autonomous framework for NDT inspection with a tilting aerial platform, in: M. H. Ang Jr, O. Khatib (Eds.), Experimental Robotics, Springer Nature Switzerland, Cham, 2024, pp. 353–363.

[8] T. Mao, K. Huang, X. Zeng, L. Ren, C. Wang, S. Li, M. Zhang, Y. Chen, Development of power transmission line defects diagnosis system for UAV inspection based on binocular depth imaging technology, in: 2019 2nd International Conference on Electrical Materials and Power Equipment, 2019, pp. 478–481.

[9] M. Selvaggio, F. Esposito, V. Lippiello, F. Ruggiero, Shared-control teleoperation methods for a cable-suspended dual-arm unmanned aerial manipulator, in: 2024 International Conference on Unmanned Aircraft Systems, 2024, pp. 1132–1139.

[10] E. Cuniato, J. Cacace, M. Selvaggio, F. Ruggiero, V. Lippiello, A hardware-in-the-loop simulator for physical human-aerial manipulator cooperation, in: 2021 20th International Conference on Advanced Robotics, 2021, pp. 830–835.

[11] M. Selvaggio, P. Robuffo Giordano, F. Ficuciello, B. Siciliano, Passive task-prioritized shared-control teleoperation with haptic guidance, in: 2019 International Conference on Robotics and Automation, 2019, pp. 430–436.

[12] A. Ollero, M. Tognon, A. Suarez, D. Lee, A. Franchi, Past, present, and future of aerial robotic manipulators, IEEE Transactions on Robotics 38 (1) (2022) 626–645.

[13] B. Siciliano, L. Villani, Indirect Force Control, Springer US, Boston, MA, 1999, pp. 31–64.

[14] B. Siciliano, L. Villani, Direct Force Control, Springer US, Boston, MA, 1999, pp. 65–87.

[15] B. Siciliano, Parallel force/position control of robot manipulators, in: G. Giralt, G. Hirzinger (Eds.), Robotics Research, Springer London, London, 1996, pp. 78–89.

[16] P. Corke, Visual control of robots: High-performance visual servoing, 1996.

[17] V. Lippiello, B. Siciliano, L. Villani, A position-based visual impedance control for robot manipulators, in: 2007 IEEE International Conference on Robotics and Automation, 2007, pp. 2068–2073.

[18] V. Lippiello, B. Siciliano, L. Villani, Interaction control of robot manipulators using force and vision, International Journal of Optomechatronics 2 (3) (2008) 257–274.

[19] Y. Zhou, X. Li, L. Yue, L. Gui, G. Sun, X. Jiang, Y.-H. Liu, Vision-based adaptive impedance control for robotic polishing, in: 2019 Chinese Control Conference, 2019, pp. 4560–4564.

[20] Y. Zhou, X. Li, L. Yue, L. Gui, G. Sun, X. Jiang, Y.-H. Liu, Global vision-based impedance control for robotic wall polishing, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2019, pp. 6022–6027.

[21] A. A. Oliva, P. R. Giordano, F. Chaumette, A general visual-impedance framework for effectively combining vision and force sensing in feature space, IEEE Robotics and Automation Letters 6 (3) (2021) 4441–4448.

[22] S. Hamaza, I. Georgilas, T. Richardson, 2d contour following with an unmanned aerial manipulator: Towards tactile-based aerial navigation, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2019, pp. 3664–3669.

[23] D. Tzoumanikas, F. Graule, Q. Yan, D. Shah, M. Popovic, S. Leutenegger, Aerial Manipulation Using Hybrid Force and Position NMPC Applied to Aerial Writing, in: Proceedings of Robotics: Science and Systems, Corvalis, Oregon, USA, 2020.

[24] R. Rashad, D. Bicego, R. Jiao, S. Sanchez-Escalonilla, S. Stramigioli, Towards vision-based impedance control for the contact inspection of unknown generically-shaped surfaces with a fully-actuated UAV, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2020, pp. 1605–1612.

[25] A. Hu, M. Xu, H. Wang, X. Liang, H. Castañeda, Vision-based hierarchical impedance control of an aerial manipulator, IEEE Transactions on Industrial Electronics (2022) 1–9.

[26] A. Hu, M. Xu, H. Wang, H. Castaneda, Vision-based impedance control of an aerial manipulator using a nonlinear observer, IEEE Transactions on Automation Science and Engineering (2022) 1–11.

[27] M. Xu, A. Hu, H. Wang, Image-based visual impedance force control for contact aerial manipulation, IEEE Transactions on Automation Science and Engineering (2022) 1–10.

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

[29] T. Lee, M. Leok, N. H. McClamroch, Nonlinear robust tracking control of a quadrotor uav on se(3), Asian Journal of Control 15 (2) (2013) 391–408.

[30] L. Peric, M. Brunner, K. Bodie, M. Tognon, R. Siegwart, Direct force and pose NMPC with multiple interaction modes for aerial push-and-slide operations, in: 2021 IEEE International Conference on Robotics and Automation, 2021, pp. 131–137.

[31] M. Tognon, H. A. T. Chávez, E. Gasparin, Q. Sablé, D. Bicego, A. Mallet, M. Lany, G. Santi, B. Revaz, J. Cortés, A. Franchi, A truly-redundant aerial manipulator system with application to push-and-slide inspection in industrial plants, IEEE Robotics and Automation Letters 4 (2) (2019) 1846–1851.

[32] G. Nava, Q. Sablé, M. Tognon, D. Pucci, A. Franchi, Direct force feedback control and online multi-task optimization for aerial manipulators, IEEE Robotics and Automation Letters 5 (2) (2020) 331–338.

[33] K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pfändler, U. Angst, R. Siegwart, J. Nieto, An omnidirectional aerial manipulation platform for contact-based inspection, in: Proceedings of Robotics: Science and Systems, FreiburgimBreisgau, Germany, 2019.

[34] K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pfändler, U. Angst, R. Siegwart, J. Nieto, Active interaction force control for contact-based inspection with a fully actuated aerial vehicle, IEEE Transactions on Robotics 37 (3) (2021) 709–722.

[35] A. Praveen, X. Ma, H. Manoj, V. L. Venkatesh, M. Rastgaar, R. M. Voyles, Inspection-on-the-fly using hybrid physical interaction control for aerial manipulators, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 1583–1588.

[36] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, D. Bicego, A. Franchi, 6d interaction control with aerial robots: The flying end-effector paradigm, The International Journal of Robotics Research 38 (9) (2019) 1045–1062.

[37] A. Shukla, H. Karki, Application of robotics in onshore oil and gas industry—a review part i, Robotics and Autonomous Systems 75 (2016) 490–507.

[38] M. Fisher, R. C. Cardoso, E. C. Collins, C. Dadswell, L. A. Dennis, C. Dixon, M. Farrell, A. Ferrando, X. Huang, M. Jump, G. Kourtis, A. Lisitsa, M. Luckcuck, S. Luo, V. Page, F. Papacchini, M. Webster, An overview of verification and validation challenges for inspection robots, Robotics 10 (2) (2021).

[39] M. Farrell, M. Luckcuck, L. Pullum, M. Fisher, A. Hessami, D. Gal, Z. Murahwi, K. Wallace, Evolution of the ieee p7009 standard: Towards fail-safe design of autonomous systems, in: 2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), 2021, pp. 401–406.

[40] A. Franchi, C. Secchi, H. I. Son, H. H. Bulthoff, P. R. Giordano, Bilateral teleoperation of groups of mobile robots with time-varying topology, IEEE Transactions on Robotics 28 (5) (2012) 1019–1033.

[41] D. Lee, A. Franchi, H. I. Son, C. Ha, H. H. Bülthoff, P. R. Giordano, Semiautonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles, IEEE/ASME Transactions on Mechatronics 18 (4) (2013) 1334–1345.

[42] M. Young, C. Miller, Y. Bi, W. Chen, B. D. Argall, Formalized task characterization for human-robot autonomy allocation, in: 2019 International Conference on Robotics and Automation, 2019, pp. 6044–6050.

[43] M. Selvaggio, J. Cacace, C. Pacchierotti, F. Ruggiero, P. R. Giordano, A shared-control teleoperation architecture for nonprehensile object transportation, IEEE Transactions on Robotics 38 (1) (2022) 569–583.

[44] J. Byun, D. Eom, H. J. Kim, Haptic-based bilateral teleoperation of aerial manipulator for extracting wedged object with compensation of human reaction time, in: 2024 International Conference on Unmanned Aircraft Systems, 2024, pp. 624–630.

[45] G. Gioioso, M. Mohammadi, A. Franchi, D. Prattichizzo, A force-based bilateral teleoperation framework for aerial robots in contact with the environment, in: 2015 IEEE International Conference on Robotics and Automation, 2015, pp. 318–324.

[46] M. Allenspach, N. Lawrance, M. Tognon, R. Siegwart, Towards 6dof bilateral teleoperation of an omnidirectional aerial vehicle for aerial physical interaction, in: 2022 International Conference on Robotics and Automation, 2022, pp. 9302–9308.

[47] A. Coelho, Y. Sarkisov, X. Wu, H. Mishra, H. Singh, A. Dietrich, A. Franchi, K. Kondak, C. Ott, Whole-body teleoperation and shared control of redundant robots with applications to aerial manipulation, Journal of Intelligent & Robotic Systems 102 (1) (Apr 2021).

[48] S. D'Angelo, F. Pagano, F. Ruggiero, V. Lippiello, Development of a control framework to autonomously install clip bird diverters on high-voltage lines, in: 2023 International Conference on Unmanned Aircraft Systems, 2023, pp. 377–382.

[49] S. D'Angelo, F. Pagano, F. Longobardi, F. Ruggiero, V. Lippiello, Efficient development of model-based controllers in px4 firmware: A template-based customization approach, in: 2024 International Conference on Unmanned Aircraft Systems, 2024, pp. 1155–1162.

[50] S. D'Angelo, A. Corrado, F. Ruggiero, J. Cacace, V. Lippiello, Stabilization and control on a pipe-rack of a wheeled mobile manipulator with a snake-like arm, Robotics and Autonomous Systems 171 (2024) 104554.

[51] K. Lynch, F. Park, Modern Robotics: Mechanics, Planning, and Control, Cambridge Univeristy Press, 2017.

[52] M. Kamel, S. Verling, O. Elkhatib, C. Sprecher, P. Wulkop, Z. Taylor, R. Siegwart, I. Gilitschenski, The voliro omniorientational hexacopter: An agile and maneuverable tiltable-rotor aerial vehicle, IEEE Robotics & Automation Magazine 25 (4) (2018) 34–44.

[53] M. Allenspach, K. Bodie, M. Brunner, L. Rinsoz, Z. Taylor, M. Kamel, R. Siegwart, J. Nieto, Design and optimal control of a tiltrotor micro-aerial vehicle for efficient omnidirectional flight, The International Journal of Robotics Research 39 (10-11) (2020) 1305–1325.

[54] E. Sadien, M. Carton, C. Grimault, L. E. Romana, C. Roos, A. Birouche, M. Basset, A detailed comparison of control allocation techniques on a realistic on-ground aircraft benchmark, in: 2019 American Control Conference (ACC), 2019, pp. 2891–2896.

[55] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, "Robotics: Modelling, Planning and Control", Springer, 2009.

[56] E. Olson, AprilTag: A robust and flexible visual fiducial system, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2011, pp. 3400–3407.

[57] S. Martin, N. Hillier, Characterisation of the novint falcon haptic device for application as a robot manipulator, in: Australasian Conference on Robotics and Automation (ACRA), Citeseer, 2009, pp. 291–292.

[58] E. Marchand, F. Spindler, F. Chaumette, Visp for visual servoing: a generic software platform with a wide class of robot control skills, IEEE Robotics and Automation Magazine 12 (4) (2005) 40–52.

[59] V. Lippiello, G. A. Fontanelli, F. Ruggiero, Image-based visual-impedance control of a dual-arm aerial manipulator, IEEE Robotics and Automation Letters 3 (3) (2018) 1856–1863.