A hardware-in-the-loop simulator for physical human-aerial manipulator cooperation

Eugenio Cuniato[1], Jonathan Cacace[2], Mario Selvaggio[2], Fabio Ruggiero[2], Vincenzo Lippiello[2]

***Abstract*— A hardware-in-the-loop simulator for human cooperation with an aerial manipulator is presented in this paper. The simulator provides the user with realistic haptic feedback proper of a human-aerial manipulator interaction activity. The forces exchanged between the hardware interface and the human/environment are measured and supplied to a dynamically simulated aerial manipulator. In turn, the simulated aerial platform feeds back its position to the hardware allowing the human to feel and evaluate the interaction effects. Besides human-aerial manipulator cooperation, the simulator lends itself to developing and testing autonomous control strategies in aerial manipulation. Therefore, the effectiveness of the proposed system is evaluated along with two case studies: a collaborative task where the human operator attaches a tool to the robot end-effector and an autonomous bird diverter installation task.**

SUPPLEMENTARY MATERIAL

Video is available at: https://youtu.be/7qHKb4OppBI. Code can be found at: https://github.com/prisma-lab/HIL_airmanip.

I. INTRODUCTION

Vertical Take-off and Landing (VToL) Unmanned Aerial Vehicles (UAVs) migrate daily from purely inspection tasks, like surveillance, monitoring, and remote sensing, into maintenance and construction tasks requiring manipulation capabilities. Therefore, VToL UAVs are nowadays often endowed with suitable grippers or manipulation tools. More often, they are instead equipped with one or more multiple Degrees of Freedom (DoFs) robotic arms, becoming Unmanned Aerial Manipulator (UAM) devices. An overview about aerial manipulation can be found in [1].

Maintenance of large linear infrastructures, such as refineries, railways, electrical lines, highways, and others, requires teams of specialised crew climbing using ropes, scaffolding, or elevated platforms. In this context, UAMs can be employed to perform manipulation while flying. To perform accurate and powerful manipulation, they can even dock on extremely constrained places, such as poles and cables. Referring to drones as UAVs or UAMs indifferently, they can interact with humans and help them in daily activities, becoming efficient Aerial Co-Workers (ACWs), particularly for working at height in inspection and maintenance activities that still require human intervention. Therefore, as long as the application range of drones increases, the possibility of sharing the human workspace increases as well. Hence, it becomes of paramount importance to understand how the interaction between humans and drones is established.

As for the classic human-robot co-working, human-drone interaction can exist in three ways: *coexisting*, *collaboration*, and *cooperation* [2]. Coexistence means that the drone and the humans share the same workspace, but they do different jobs. Collaboration is intended as both the drone and the humans work on the same task, but they are not in physical contact. Cooperation between a drone and a human worker means direct or indirect physical contact to pursue the final goal. Safety is indeed the word joining all these aspects.

So far, most works in the literature deal with human-drone coexistence and collaboration. In these contexts, research is devoted to finding ways to control and communicate with the drone. This is usually pursued through gestures [3], speech [4], brain-computer interfaces [5], and multimodal interaction [6], [7]. Some works are trying to feedback information from the drone to the human operator to reduce its workload and improve its state awareness [8]. An exhaustive survey can be found in [9].

A few articles deal with the problem tackled by this work: the proximal cooperation between a human and a drone involving physical contacts [10]. A safe-to-touch UAV was considered in [11] to understand if humans feel comfortable touching UAVs and if they naturally choose touch as a means of interaction. Touch was also exploited in [12] where the interaction is measured by looking at the peaks in the built-in inertial and measurement unit of the UAV. By dispensing with additional sensors, the employed UAV size can be kept to a minimum. A hardware-in-the-loop (HIL) simulator for human-UAV interaction was devised in [13]. The human can intuitively command a UAV by exchanging forces measured by pushing some buttons in certain locations. Sensors can distinguish between human interaction forces and other disturbances, thanks to an estimator of unmodelled wrenches. The estimated values are, in turn, used in an admittance control scheme that modifies the planned trajectory based on the human interaction. The admittance control approach to physically interact with UAVs has also been used in [14]. External forces acting on the UAV are estimated from position and attitude information and then sent to the admittance controller, which modifies the vehicle reference trajectory accordingly. The UAVs mentioned in these past works were

The research leading to these results has been supported by both the AERIAL-CORE project, European Union's Horizon 2020 research and innovation programme, under Grant Agreement No 871479, and the AERO-TRAIN project, European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 953454. The authors are solely responsible for its content.

[1] Autonomous Systems Lab, ETH Zurich, Switzerland. Corresponding author's e-mail: ecuniato@ethz.ch

[2] PRISMA Lab, Department of Electrical Engineering and Information Technology, University of Naples Federico II, 80125, Naples, Italy.

quadcopters. An admittance control-based scheme to perform physical interaction with a fully actuated UAV (i.e., a tilted hexacopter) was proposed in [15]. In the above-cited works, the human interaction with the UAV is exerted through pushing actions. Instead, a tethered interaction between the human and the UAV was investigated in [16]. The human is physically connected to an aerial vehicle by a cable: the exchanged forces are used as an indirect communication channel. An admittance-based control approach is employed, and the system is shown to be passive.

The above literature review shows that the human-drone cooperation was carried out so far through UAVs only. Unlike the previous works, this paper deals with the physical human-drone collaboration using UAMs, devising a HIL simulator. The advantages of a HIL simulator for this application are listed in [13]. Also, it may play the role of a training interface for workers since such a simulator aims to provide the user with a realistic haptic sensation proper of a human-aerial manipulator interaction activity. The simulator lends itself to developing and testing both autonomous and human-aerial manipulator interaction control strategies. In this approach, the simulated UAM interacts with the human/environment through a hardware interface and an underlying software communication framework connecting the real and simulated worlds. The hardware interface allows a force exchange between the human/environment and the simulation while mimicking the manipulator's behaviour attached to the UAV. The software communication interface introduces a safety layer based on passivity arguments to guarantee a safe human operator interaction. Moreover, it avoids moving the hardware interface outside of a predetermined bounding box.

The provided contributions that make this work different from the existing literature (e.g., [14], [13], and [15]) namely are: *(i)* the construction of a HIL simulator for a UAM instead of a UAV; *(ii)* the physical presence of a moving robot interacting with the human operator; *(iii)* the additional possibility to employ the HIL simulator in autonomous tasks; *(iv)* the presence of an energy-tank to guarantee passivity and increase safety during human interaction; *(v)* the re-usability of the architecture thanks to the free distribution of the simulator's source code.

As for the paper's outline, the next section describes the architecture of the implemented HIL simulator: the simulation and hardware sides and the communication interface will be detailed. Section III presents the controllers implementation, while in Section IV the case study and the related results are discussed. Section V concludes the paper.

II. System architecture

The proposed system is schematically represented in Fig. 1, and it is composed of three main parts: (i) a model-based simulation of a UAM, composed of a quadrotor (the floating base) with a 6-DoFs robotic arm; (ii) a hardware interface to enable a force-based interaction with the simulated robotic model while rendering the motion effects of the floating base; (iii) a bilateral communication interface connecting the hardware with the simulated model.

All these components are integrated through the Robot Operating System (ROS) framework, as shown in Fig. 1, with specifically designed modules to obtain sensor data, and control both the simulated and real systems.

As the hardware interface (see Sec. II-B) interacts with the human operator or the environment, all the exchanged forces, measured by a force sensor, are applied to the simulated UAM in the simulated environment (see Sec. II-A). The simulation is carried out in Gazebo, which is a widely employed robot simulator with physics engines. However, our architecture is general and can integrate different dynamic simulators (e.g., CoppeliaSim, DART, etc.). The position of the floating base, affected by the interaction forces, is then fed back to the hardware interface, which adjusts its position consequently (see Sec. II-C). This information is exchanged simultaneously using standard ROS messages. The hardware and the simulation controllers, developed for the envisioned tasks/experiments, are detailed in Sec. IV. However, notice that our architecture can accommodate different controllers, which the users might develop as ROS modules according to their needs. The code is indeed freely released[1].

A. Simulation side

The simulated quadrotor was implemented using the plugins and functionalities offered by the RotorS library [17]. It is modelled as a rigid body in the space actuated by four propellers with parallel axes. Its position and orientation are defined through the frame Σ_b attached to the body center of mass (CoM), expressed in the inertial frame Σ_w. The position of Σ_b in Σ_w is denoted by $\mathbf{p}_b = [\, x \; y \; z \,]^T \in \mathbb{R}^3$, whereas its attitude is described by the rotation matrix $\mathbf{R}_b \in SO(3)$. The quadrotor dynamic model is

$$m\ddot{\mathbf{p}}_b = mg\mathbf{e}_3 - u_T\mathbf{R}_b\mathbf{e}_3 + \mathbf{f}_{ext}, \tag{1a}$$

$$\dot{\mathbf{R}}_b = \mathbf{R}_b S(\boldsymbol{\omega}_b^b), \tag{1b}$$

$$\mathbf{I}_b\dot{\boldsymbol{\omega}}_b^b = -S(\boldsymbol{\omega}_b^b)\mathbf{I}_b\boldsymbol{\omega}_b^b + \boldsymbol{\tau}^b + \boldsymbol{\tau}_{ext}^b, \tag{1c}$$

where $\mathbf{e}_3 = [\, 0 \;\; 0 \;\; 1 \,]^T \in \mathbb{R}^3$, $m > 0$ is the quadrotor mass, $\mathbf{I}_b \in \mathbb{R}^{3\times3}$ is its symmetric and positive definite inertia matrix in Σ_b, $\boldsymbol{\omega}_b^b \in \mathbb{R}^3$ is the rotation velocity vector of Σ_b with respect to itself, $S(\cdot) \in \mathbb{R}^{3\times3}$ is the skew-symmetric operator, $u_T \in \mathbb{R}$ and $\boldsymbol{\tau}^b = [\, \tau_x \; \tau_y \; \tau_z \,]^T \in \mathbb{R}^3$ are the total thrust force and the control torques given by the propellers, respectively. Finally, $\mathbf{f}_{ext} \in \mathbb{R}^3$ and $\boldsymbol{\tau}_{ext}^b \in \mathbb{R}^3$ represent the external forces and torques disturbances acting on the quadrotor, respectively. These last take into account unmodelled terms (e.g., aerodynamic disturbances), arm movements, and human-UAM interaction forces.

Defining as $\Omega_i \in \mathbb{R}$, with $i = 1, ..., 4$, the rotational velocities of the four propellers, the quadrotor's control input $\mathbf{u} = [\, u_T \; \boldsymbol{\tau}^{b^T} \,]^T$ can be computed as [18]

$$\mathbf{u} = \begin{bmatrix} c_t & c_t & c_t & c_t \\ 0 & l\,c_t & 0 & -l\,c_t \\ -l\,c_t & 0 & l\,c_t & 0 \\ c_a & -c_a & c_a & -c_a \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}, \tag{2}$$

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

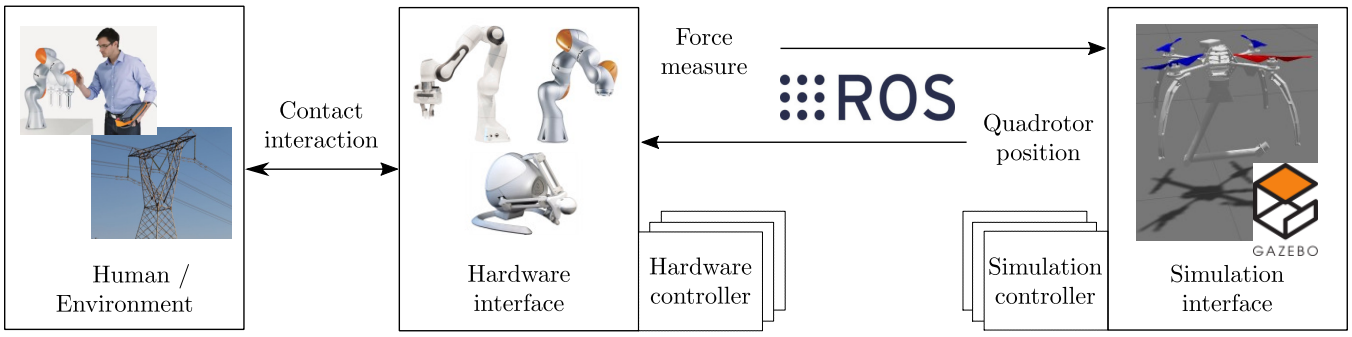

Fig. 1. Conceptual scheme of the system architecture.

where $c_t > 0$, $c_a > 0$ represent the thrust and torque coefficients of the propellers, respectively, and $l > 0$ is the distance between each propeller and the origin of Σ_b.

The actuated joints of the 6-DoF arm attached to the quadrotor are configured as an anthropomorphic arm with a spherical wrist. Denote with $Sigma_e$ the manipulator's end-effector frame while its base frame coincides with Σ_b. The direct kinematics from Σ_b to Σ_e is described by

$$\mathbf{A}_e^b = \begin{bmatrix} \mathbf{R}_e^b(\mathbf{q}) & \mathbf{p}_e^b(\mathbf{q}) \\ \mathbf{0}_3^T & 1 \end{bmatrix}, \tag{3}$$

where $\mathbf{q} = [q_1 \ \dots \ q_6]^T \in \mathbb{R}^6$ is the vector of joint coordinates, $\mathbf{R}_e^b \in SO(3)$ and $\mathbf{p}_e^b \in \mathbb{R}^3$ are the rotation matrix and the position of Σ_e with respect to Σ_b, respectively, and $\mathbf{0}_n \in \mathbb{R}^n$ is the zero vector of proper dimensions.

B. Hardware

The hardware interface connected to the simulation allows the user to interact with the simulated environment through the measured contact forces. At the same time, the user receives the haptic feedback related to the movements of the simulated UAM, also taking into account the position of the floating base. The architecture is independent of the specific used hardware, which can be any device as long as its position in the space can be commanded (e.g., manipulators and/or haptic interfaces). In this work, the hardware interface consists of a 7-DoFs KUKA IIWA manipulator equipped with an ATI Mini45 force/torque sensor at the end-effector. Notice that, since the hardware interface mimics the simulated UAM in operational space coordinates, the two kinematic chains can be structurally different as long as a suitable inverse kinematics algorithm is adopted.

In particular, to perform a safe human-robot interaction, the dynamics of the hardware interface is given by an admittance controller. Let $\Sigma_{b'}$ be the base frame of the hardware side. Let $\mathbf{x}_{e'}^{b'} \in \mathbb{R}^3$ be the position of the hardware end-effector frame, $\Sigma_{e'}$, with respect to $\Sigma_{b'}$, and let $\mathbf{x}_{e',c}^{b'} \in \mathbb{R}^3$ be the position of the compliant frame, $\Sigma_{e',c}$, with respect to $\Sigma_{b'}$. The hardware-side admittance equations are

$$\mathbf{M}_d \ddot{\tilde{\mathbf{x}}}' + \mathbf{K}_D \dot{\tilde{\mathbf{x}}}' + \mathbf{K}_P \tilde{\mathbf{x}}' = \mathbf{h}_h + \mathbf{h}_q, \tag{4}$$

where $\mathbf{M}_d \in \mathbb{R}^{6\times6}$, $\mathbf{K}_D \in \mathbb{R}^{6\times6}$, and $\mathbf{K}_P \in \mathbb{R}^{6\times6}$ are the apparent mass, damping, and stiffness matrices, respectively, while $\tilde{\mathbf{x}}' = \begin{bmatrix} \tilde{\mathbf{x}}_p'^T & \tilde{\mathbf{x}}_o'^T \end{bmatrix}^T \in \mathbb{R}^6$ is the operational space error between $\Sigma_{e'}$ and $\Sigma_{e',c}$. In particular, $\tilde{\mathbf{x}}_p' = \mathbf{x}_{e'}^{b'} - \mathbf{x}_{e',c}^{b'} \in \mathbb{R}^3$ is the position error, whereas the orientation error $\tilde{\mathbf{x}}_o' \in \mathbb{R}^3$ is given by the vector part of the quaternion representing the rotation between the hardware compliant frame and the hardware end-effector frame, expressed by $\mathbf{R}_{e',c}^{e'} \in SO(3)$. The right-hand side of (4) represents the wrench applied to the system given by: (i) $\mathbf{h}_h \in \mathbb{R}^6$, the interaction wrench exerted by the human operator at the hardware end-effector $\Sigma_{e'}$ expressed in $\Sigma_{b'}$; (ii) $\mathbf{h}_q \in \mathbb{R}^6$, the haptic feedback from the simulated UAM and detailed in Sec. II-C.

Notice that, besides human-aerial manipulator interaction, the same hardware (capable of rendering the UAM dynamics given by the term $\mathbf{h}_q$ in (4)) can be used to test advanced control methods for autonomous aerial manipulation tasks.

C. Communication interface

The connection between the real hardware and the simulation is implemented as follows. The wrench, $\mathbf{h}_e \in \mathbb{R}^6$, applied in simulation at the UAM's end-effector can be then computed as $\mathbf{h}_e = \mathbf{Ad}_{T_{ew}} \mathbf{Ad}_{T_{wb'}} \mathbf{Ad}_{T_{b'e'}} \mathbf{h}_h$,where $\mathbf{Ad}_{T_{12}} \in \mathbb{R}^{6\times6}$ is the adjoint transformation matrix between two generic frames, Σ_1 and Σ_2 [19]. Notice that the transformation matrix $\boldsymbol{T}_{b'}^w$ should be chosen by the user to connect the real hardware base frame to the simulated world.

Regarding the simulation feedback to the hardware interface, the displacement $\mathbf{e}_p \in \mathbb{R}^3$ between the commanded and the actual position of the quadrotor's CoM can be fed back onto the hardware interface to emulate the effects of the floating base displacements. In particular, this contribution can be seen as an additional wrench (added to $\mathbf{h}_h$), playing the role of haptic feedback for the human operator, given by

$$\mathbf{M}_d \ddot{\mathbf{e}}_p' + \mathbf{K}_D \dot{\mathbf{e}}_p' + \mathbf{K}_P \mathbf{e}_p' = \mathbf{h}_q \,, \tag{5}$$

where $\mathbf{e}_p' = \mathbf{Ad}_{T_{b'w}} \mathbf{e}_p$ is the quadrotor's displacement transformed from Σ_w into $\Sigma_{b'}$.

III. System Controllers

This section shows how our proposed architecture lends itself to simulating both human-UAM and autonomous interaction tasks. We present the hardware and the simulation controllers later used in the two case studies (see Sec. IV). Here, a decentralised controller [1] has been implemented

on the simulated UAM, employing a geometric position tracking controller for the quadrotor and a Cartesian variable admittance controller for the arm. Also, the quadrotor's position tracking is enhanced with a momentum-based external wrench estimator [20] to compensate for the arm dynamics, the interaction forces, and other unmodelled disturbances.

A. Quadrotor controller

Because of the under-actuation of the system, a hierarchical approach is followed to control both the position, $\mathbf{p}_b$, and the attitude, $\mathbf{R}_b$, of the quadrotor. From this point of view, the geometric tracking controller in $SE(3)$ [21] is implemented on the quadrotor.

The outer position loop tracking errors are $\mathbf{e}_p = \mathbf{p}_b - \mathbf{p}_{b,d}$ and $\mathbf{e}_v = \dot{\mathbf{p}}_b - \dot{\mathbf{p}}_{b,d}$, where $\mathbf{p}_{b,d} \in \mathbb{R}^3$ is the desired position of the Σ_b's origin in Σ_w, obtained from an external trajectory planner. Let $\mathbf{R}_{b,d} = [\,\mathbf{x}_{b,d}\ \mathbf{y}_{b,d}\ \mathbf{z}_{b,d}\,] \in SO(3)$ be the desired rotation matrix, where $x_{b,d} \in \mathbb{R}^3$ is given from the trajectory planner. The necessary thrust u_T and the desired body axis $\mathbf{z}_{b,d} \in R^3$ can be computed as

$$u_T = (\mathbf{K}_p\,\mathbf{e}_p + \mathbf{K}_v\mathbf{e}_v + mg\mathbf{e}_3 - m\ddot{\mathbf{p}}_{b,d})^T\mathbf{R}_b\mathbf{e}_3, \quad (6a)$$

$$\mathbf{z}_{b,d} = -\frac{-\mathbf{K}_p\mathbf{e}_p - \mathbf{K}_v\mathbf{e}_v - mg\mathbf{e}_3 + m\ddot{\mathbf{p}}_{b,d}}{\| -\mathbf{K}_p\mathbf{e}_p - \mathbf{K}_v\mathbf{e}_v - mg\mathbf{e}_3 + m\ddot{\mathbf{p}}_{b,d}\|}, \quad (6b)$$

where $\mathbf{K}_p \in \mathbb{R}^{3\times3}$ and $\mathbf{K}_v \in \mathbb{R}^{3\times3}$ are positive definite gain matrices, while $\|\cdot\| > 0$ is the Cartesian norm. The desired rotation matrix, $\mathbf{R}_{b,d}$, is then constructed accordingly to the needs of the position control loop as $\mathbf{y}_{b,d} = \dfrac{S(\mathbf{z}_{b,d})\mathbf{x}_{b,d}}{\|S(\mathbf{z}_{b,d})\mathbf{x}_{b,d}\|}$ and $\mathbf{x}_{b,d} = S(\mathbf{y}_{b,d})\mathbf{z}_{b,d}$.

The tracking errors of the inner attitude loop are given by $\mathbf{e}_R = 0.5\,(\mathbf{R}_{b,d}^T\mathbf{R}_b - \mathbf{R}_b^T\mathbf{R}_{b,d})^\vee$ and $\mathbf{e}_\omega = \boldsymbol{\omega}_b^b - \mathbf{R}_b^T\mathbf{R}_{b,d}\,\boldsymbol{\omega}_{b,d}^{b,d}$, where $\boldsymbol{\omega}_{b,d}^{b,d} \in \mathbb{R}^3$ the desired body rotation velocity in Σ_b, and $^\vee : \mathbb{R}^{3\times3} \to \mathbb{R}^3$ is a map performing the inverse of the skew-symmetric operator. The control law

$$\boldsymbol{\tau}^b = -\mathbf{K}_R\,\mathbf{e}_R - \mathbf{K}_\omega\,\mathbf{e}_\omega + S(\boldsymbol{\omega}_b^b)\,\mathbf{I}_b\,\boldsymbol{\omega}_b^b + \\ - \mathbf{I}_b\,[S(\boldsymbol{\omega}_b^b)\,\mathbf{R}_b^T\,\mathbf{R}_{b,d}\,\boldsymbol{\omega}_{b,d}^{b,d} - \mathbf{R}_b^T\,\mathbf{R}_{b,d}\,\dot{\boldsymbol{\omega}}_{b,d}^{b,d}] \quad (7)$$

asymptotically stabilises the attitude dynamics if $\mathbf{K}_R \in \mathbb{R}^{3\times3}$ and $\mathbf{K}_\omega \in \mathbb{R}^{3\times3}$ are positive definite gain matrices. Because there might be significant external disturbances in our case, this control scheme needs an external wrench estimator to keep good performance in trajectory tracking. In this work, the estimator presented in [20] has been used.

B. Arm variable admittance controller

Due to the limited payload capabilities of aerial platforms, the arm of a UAM is typically actuated by position-controlled joints (e.g., servo motors) [1]. Admittance control is implemented to interact with the humans safely and operate in contact with the environment, exhibiting a desired compliant behaviour. However, as human and environment interaction tasks may require different compliant choices, a variable-gain admittance control is used. In the operational space, the admittance controlled manipulator dynamics is given by

$$\mathbf{M}_d\ddot{\tilde{\mathbf{x}}} + \mathbf{K}_D\dot{\tilde{\mathbf{x}}} + \mathbf{K}_P\tilde{\mathbf{x}} = \mathbf{h}_e^b, \quad (8)$$

where $\tilde{\mathbf{x}} = \begin{bmatrix}\tilde{\mathbf{x}}_p^T & \tilde{\mathbf{x}}_o^T\end{bmatrix}^T \in \mathbb{R}^6$ is the operational space error between the desired end-effector frame and the compliant frame, in the simulation side, while $\mathbf{h}_e^b \in \mathbb{R}^6$ is the wrench measured at the simulated end-effector, expressed in Σ_b. In particular, $\tilde{\mathbf{x}}_p = \mathbf{x}_e^b - \mathbf{x}_{e,c}^b \in \mathbb{R}^3$ is the position error, whereas the orientation error $\tilde{\mathbf{x}}_o \in \mathbb{R}^3$ is given by the vector part of the quaternion representing the rotation between the simulated compliant frame and the simulated end-effector frame, expressed by $\mathbf{R}_{e,c}^e \in SO(3)$.

C. Passivity analysis

The dynamics in (8) are passive with respect to the power port $(\mathbf{h}_e^b, \dot{\tilde{\mathbf{x}}})$. Indeed, by choosing as a storage function

$$V(\tilde{\mathbf{x}}, \dot{\tilde{\mathbf{x}}}) = \frac{1}{2}\dot{\tilde{\mathbf{x}}}^T\mathbf{M}_d\dot{\tilde{\mathbf{x}}} + \frac{1}{2}\tilde{\mathbf{x}}^T\mathbf{K}_P\tilde{\mathbf{x}}\,, \quad (9)$$

its time derivative is

$$\dot{V} = \dot{\tilde{\mathbf{x}}}^T\mathbf{M}_d\ddot{\tilde{\mathbf{x}}} + \dot{\tilde{\mathbf{x}}}^T\mathbf{K}_P\tilde{\mathbf{x}} = \dot{\tilde{\mathbf{x}}}^T\mathbf{h}_e^b - \dot{\tilde{\mathbf{x}}}^T\mathbf{K}_D\dot{\tilde{\mathbf{x}}} \le \dot{\tilde{\mathbf{x}}}^T\mathbf{h}_e^b. \quad (10)$$

The same passivity argument can be extended to the hardware side (see (4)) with respect to the interaction force $\mathbf{h}_h$. By defining $\bar{\mathbf{x}} = \tilde{\mathbf{x}} - \boldsymbol{e}_p'$, equation (4) can be rewritten as $\mathbf{M}_d\ddot{\bar{\mathbf{x}}} + \mathbf{K}_D\dot{\bar{\mathbf{x}}} + \mathbf{K}_P\bar{\mathbf{x}} = \mathbf{h}_h$, which is passive with respect to the power port $(\boldsymbol{h}_h, \dot{\bar{\mathbf{x}}})$ with storage function

$$V(\bar{\mathbf{x}}, \dot{\bar{\mathbf{x}}}) = \frac{1}{2}\dot{\bar{\mathbf{x}}}^T\mathbf{M}_d\dot{\bar{\mathbf{x}}} + \frac{1}{2}\bar{\mathbf{x}}^T\mathbf{K}_P\bar{\mathbf{x}}\,. \quad (11)$$

However, if the admittance gains are time-variant, functions in (9) and (11) are no longer valid. Indeed, equation (10) becomes

$$\begin{aligned}\dot{V} &= \dot{\tilde{\mathbf{x}}}^T\mathbf{M}_d\ddot{\tilde{\mathbf{x}}} + \dot{\tilde{\mathbf{x}}}^T\mathbf{K}_P\tilde{\mathbf{x}} + \frac{1}{2}\left[\dot{\tilde{\mathbf{x}}}^T\dot{\mathbf{M}}_d\dot{\tilde{\mathbf{x}}} + \tilde{\mathbf{x}}^T\dot{\mathbf{K}}_P\tilde{\mathbf{x}}\right] \\ &= \dot{\tilde{\mathbf{x}}}^T\mathbf{h}_e - \dot{\tilde{\mathbf{x}}}^T\mathbf{K}_D\dot{\tilde{\mathbf{x}}} + \frac{1}{2}\left[\dot{\tilde{\mathbf{x}}}^T\dot{\mathbf{M}}_d\dot{\tilde{\mathbf{x}}} + \tilde{\mathbf{x}}^T\dot{\mathbf{K}}_P\tilde{\mathbf{x}}\right]\,,\end{aligned} \quad (12)$$

and passivity is guaranteed only if the following holds

$$\dot{\tilde{\mathbf{x}}}^T\mathbf{K}_D\dot{\tilde{\mathbf{x}}} \ge \frac{1}{2}\left[\dot{\tilde{\mathbf{x}}}^T\dot{\mathbf{M}}_d\dot{\tilde{\mathbf{x}}} + \tilde{\mathbf{x}}^T\dot{\mathbf{K}}_P\tilde{\mathbf{x}}\right]\,. \quad (13)$$

Notice that the same can be applied to (11). To guarantee passivity despite (13), an *energy-tank* can be employed [22], [23]. The tank dynamics is $\dot{z} = \dfrac{\varphi}{z}P_d - \dfrac{1}{z}\gamma w$, where $\varphi \in \mathbb{R}$ and $\gamma \in R$ are two parameters, $P_d = \dot{\tilde{\mathbf{x}}}^T\mathbf{K}_D\dot{\tilde{\mathbf{x}}} \ge 0$ is the power dissipated by the admittance system (8), $w = \dfrac{1}{2}\dot{\tilde{\mathbf{x}}}^T\dot{\mathbf{M}}_d\dot{\tilde{\mathbf{x}}} + \tilde{\mathbf{x}}^T\dot{\mathbf{K}}_P\tilde{\mathbf{x}}$ is the tank input. Let define the tank's storage function as $\mathcal{T}(z) = 0.5z^2$. To guarantee the passivity, γ and φ are chosen as

$$\gamma = \begin{cases}0, & \text{if } \mathcal{T} \ge \overline{\mathcal{T}}\ \&\ w \le 0\\ \alpha, & \text{otherwise}\end{cases} \quad (14a)$$

$$\alpha = \begin{cases}1, & \text{if } w > 0\\ \frac{1}{2}\left(1 - cos(\pi\frac{\mathcal{T}-\underline{\mathcal{T}}}{\overline{\mathcal{T}}-\underline{\mathcal{T}}})\right), & \text{otherwise}\end{cases} \quad (14b)$$

$$\varphi = \begin{cases}\varphi_d, & \text{if } \mathcal{T} < \overline{\mathcal{T}}\\ 0, & \text{otherwise}\end{cases}\,, \quad (14c)$$

with $\varphi_d \le 1$ represents the amount of the dissipated energy redirected to the tank, while $\overline{\mathcal{T}} \in \mathbb{R}$ and $\underline{\mathcal{T}} \in \mathbb{R}$ are the upper and lower energy limits of the tank, respectively.

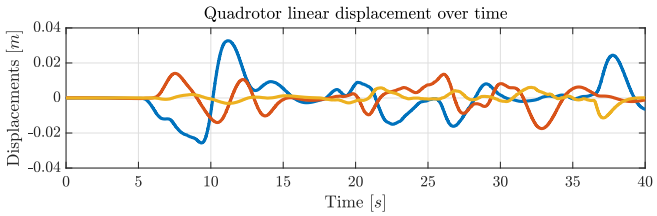

Fig. 2. Aerial base position displacement $\boldsymbol{e}_p$ along axes x (blue), y (red) and z (orange) during experiment A.

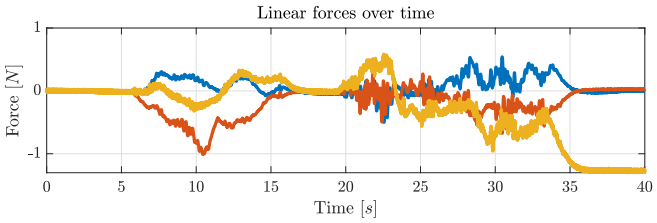

Fig. 3. Human-hardware contact forces $\boldsymbol{h}_{e'}^{b'}$ along axes x (blue), y (red) and z (orange) during Experiment A.

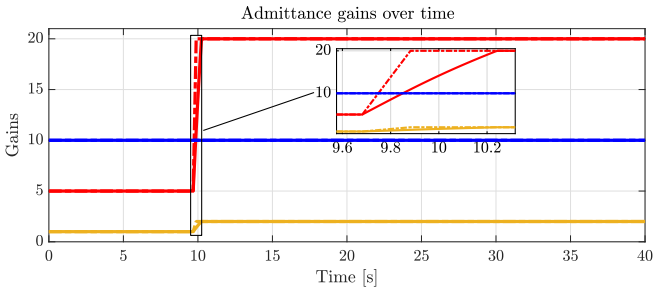

Fig. 4. Quadrotor arm admittance gains $\mathbf{K}_P$ (red), $\mathbf{K}_D$ (blue) and $\mathbf{M}_d$ (orange) during experiment A: desired (dashed) and actual profiles (solid).

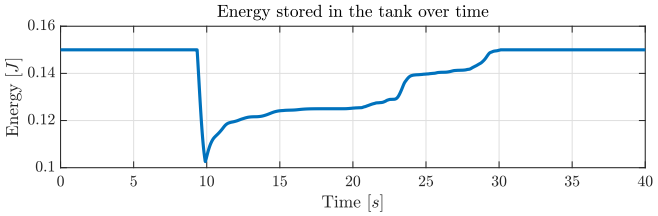

Fig. 5. Energy $\mathcal{T}$ inside the tank during experiment A.

If the desired values of $\dot{\mathbf{M}}_d$ and $\dot{\mathbf{K}}_P$ are substituted by $\alpha\dot{\mathbf{M}}_d$ and $\alpha\dot{\mathbf{K}}_P$, respectively, the overall storage function is given by

$$\begin{aligned}\dot{V}+\dot{\mathcal{T}} &= \dot{\tilde{\mathbf{x}}}^T\mathbf{M}_d\ddot{\tilde{\mathbf{x}}}+\dot{\tilde{\mathbf{x}}}^T\mathbf{K}_P\tilde{\mathbf{x}}+\alpha w+\varphi P_d-\gamma w\\ &= \dot{\tilde{\mathbf{x}}}^T\mathbf{h}_e-\dot{\tilde{\mathbf{x}}}^T\mathbf{K}_D\dot{\tilde{\mathbf{x}}}+\alpha w+\varphi P_d-\gamma w \qquad (15)\\ &= \dot{\tilde{\mathbf{x}}}^T\mathbf{h}_e-(1-\varphi)P_d+(\alpha-\gamma)w\leq\dot{\tilde{\mathbf{x}}}^T\mathbf{h}_e\,,\end{aligned}$$

where the passivity of arm plus tank system is always verified despite the sign of w.

The energy tank partially stores the energy dissipated by the admittance dynamics, releasing it later if necessary. By injecting controlled amounts of energy in the system, the tank temporarily violates the system passivity, like changing the admittance virtual stiffness or mass.

IV. Case Study

The effectiveness of the proposed system is evaluated along with two case studies: (i) a collaborative experiment where the human operator attaches a tool to the robot end-effector; (ii) an autonomous bird diverter installation task. The simulations have been performed on a standard Ubuntu 18.04 distribution with ROS Melodic, running at 200 Hz. This frequency was chosen accordingly to the computing performance of the platform running the simulation side.

A. Human-drone interaction

The first part of the experiment considers the collaboration task between a human operator and the UAM through the hardware interface. In this context, the aerial platform awaits in mid-air for a contact from a human operator. Correspondingly, the hardware interface keeps its initial position until the interaction starts. The operator approaches and starts the interaction phase with the admittance-controlled hardware interface, which provides a compliant behaviour. The goal in this phase is to mount a tool on the manipulator gripper. When the aerial platform recognises that both the tool is mounted on the hardware interface and the operator has finished the interaction with it, the experiment ends.

The interaction between the human operator and the simulation framework is demonstrated in Fig. 2 by the motion of the aerial manipulator subject to the forces generated on the hardware side. As stated above, these positions are fed back to the hardware to provide the operator with a realistic interaction feeling. The following variables have been recorded during each experiment: the human-hardware contact forces $\boldsymbol{h}_h$; the aerial manipulator displacement $\boldsymbol{e}_p$; the admittance control gains variations.

The human-drone interaction forces at the arm's tip, expressed in the world frame, are shown in Fig. 3. After a few seconds from the beginning of the experiment, the operator grabs the hardware manipulator. In this case, the interaction lasts for about 30 s, until the operator successfully mounts the tool. At this point, the tool weight force of about 1.4 N is the only one applied at the manipulator's end-effector. The linear displacement of Σ_b relative to the commanded fixed position, due to the interaction forces, is represented in Fig. 2. The admittance gains are diagonal matrices whose variation is shown in Fig. 4. The arm starts with low gains to improve the comfort of the human operator and increase system's safety. When a human contact is found (around 10 s), the virtual stiffness and the mass gains are increased to aid the tool's placement process. The increased gains violate the arm passivity which can be guaranteed by means of an energy tank. The tank partially discharges as in Fig. 5, consequently delaying the increase in the admittance gains. However, later in the experiment, the tank recharges thanks to the energy dissipated from the human interaction.

B. Bird diverter installation

To install the diverter, it must impact the power line with sufficient force. The arm admittance controller gains

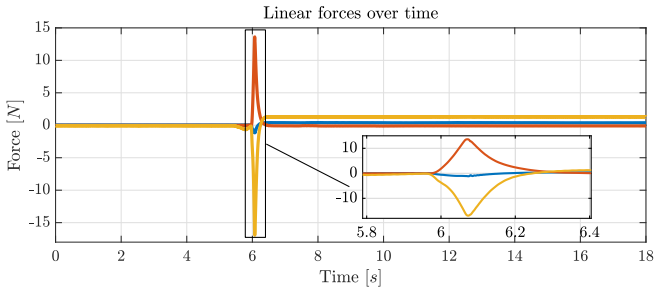

Fig. 6. Diverter installation impact forces $\boldsymbol{h}_{e'}^{b'}$ along axes x (blue), y (red) and z (orange) during experiment B.

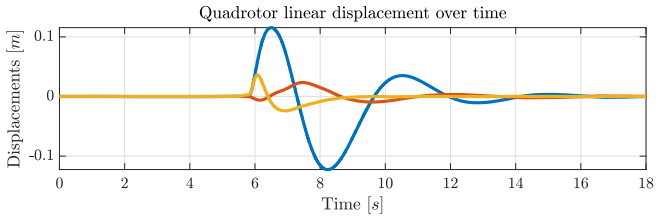

Fig. 7. Aerial base position displacement $\boldsymbol{e}_p$ in Σ_w along axes x (blue), y (red) and z (orange) during experiment B.

are increased with respect to the human-UAM interaction phase to provide the necessary rigidity. Only the gains along the front direction are kept low to ease the diverter installation and attenuate the impact effects on the floating base. In the beginning, the aerial manipulator is in free flight approaching the installation point. When this point is reached, the quadrotor is commanded to be still in position. Then, the arm is positioned under the aerial cable and rapidly rises to hook the diverter.

Because of the impact forces, shown in Fig. 6, right after the diverter was hooked, the quadrotor undergoes a displacement of about 0.1 m along the x-axis in Σ_w, which is recovered by the flight controller as shown in Fig. 7.

V. Conclusion

A general framework for HIL simulation of human-aerial manipulator collaboration was presented. We first described the simulated aerial manipulator and the hardware interface. These two were then connected through a communication interface to implement the force and position feedback between the simulation and the environment. The overall architecture's effectiveness was evaluated in two case studies: a collaborative task with a human operator and an autonomous bird diverter installation task. During these experiments, we demonstrated the possibility of performing human-aerial manipulator interaction without endangering the operator.

References

[1] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Rob. Autom. Lett.*, vol. 3, no. 3, pp. 1957–1964, 2018.

[2] S. Haddadin and E. Croft, "Physical human-robot interaction," in *Springer Handbook of Robotics*, 2nd ed., B. Siciliano and O. Khatib, Eds. Springer, 2016, pp. 1835–1874.

[3] A. C. S. Medeiros, P. Ratsamee, Y. Uranishi, T. Mashita, and H. Takemura, "Human-drone interaction: Using pointing gesture to define a target object," in *Human-Computer Interaction. Multimodal and Natural Interaction*, M. Kurosu, Ed. Springer International Publishing, Cham, 2020, pp. 688–705.

[4] J. Cacace, A. Finzi, V. Lippiello, M. Furci, N. Mimmo, and L. Marconi, "A control architecture for multiple drones operated via multimodal interaction in search rescue mission," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2016, pp. 233–239.

[5] D. Tezza, S. Garcia, T. Hossain, and M. Andujar, "Brain eRacing: An exploratory study on virtual brain-controlled drones," in *Virtual, Augmented and Mixed Reality. Applications and Case Studies (Lecture Notes in Computer Science)*, J. Chen and G. Fragomeni, Eds. Springer International Publishing, Cham, 2019, vol. 11575, pp. 150–162.

[6] J. L. E., I. L. E., J. A. Landay, and J. R. Cauchard, "Drone & Wo: Cultural influences on human-drone interaction techniques," in *2017 CHI Conference on Human Factors in Computing Systems*, 2017, pp. 6794—-6799.

[7] J. Cacace, A. Finzi, and V. Lippiello, "A robust multimodal fusion framework for command interpretation in human-robot cooperation," in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication*, 2017, pp. 372–377.

[8] C. Rognon, M. Koehler, C. Duriez, D. Floreano, and A. M. Okamura, "Soft haptic device to render the sensation of flying like a drone," *IEEE Rob. Autom. Lett.*, vol. 4, no. 3, pp. 2524–2531, 2019.

[9] D. Tezza and M. Andujar, "The state-of-the-art of human-drone interaction: A survey," *IEEE Access*, vol. 7, pp. 167 438–167 454, 2019.

[10] M. Selvaggio, M. Cognetti, S. Nikolaidis, S. Ivaldi, and B. Siciliano, "Autonomy in physical human-robot interaction: A brief survey," *IEEE Rob. Autom. Lett.*, vol. 6, no. 4, pp. 7989–7996, 2021.

[11] P. Abtahi, D. Y. Zhao, J. L. E., and J. A. Landay, "Drone near me: Exploring touch-based human-drone interaction," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 3, 2017.

[12] M. Lieser, U. Schwanecke, and J. Berdux, "Tactile human-quadrotor interaction: Metrodrone," in *15th International Conference on Tangible, Embedded, and Embodied Interaction*, 2021, pp. 1–6.

[13] S. Rajappa, H. Bülthoff, and P. Stegagno, "Design and implementation of a novel architecture for physical human-UAV interaction," *Int. J. Rob. Res.*, vol. 36, no. 5–7, pp. 800–819, 2017.

[14] F. Augugliaro and R. D'Andrea, "Admittance control for physical human-quadcopter interaction," in *2013 European Control Conference*, 2013, pp. 1805–1810.

[15] S. Rajappa, H. Bülthoff, M. Odelga, and P. Stegagno, "A control architecture for physical human-UAV interaction with a fully actuated hexarotor," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 4618–4625.

[16] M. Tognon, R. Alami, and B. Siciliano, "Physical human-robot interaction with a tethered aerial vehicle: Application to a force-based human guiding problem," *IEEE Trans. Rob.*, pp. 1–12, 2021.

[17] F. Furrer, M. Burri, and R. Achtelik, M. aand Sigwart, "RotorS—A modular Gazebo MAV simulator framework," in *Robot Operating System (ROS). Studies in Computational Intelligence*, K. A. (eds), Ed. Springer, Cham, 2016, vol. 625.

[18] T. Madani and A. Benallegue, "Backstepping control for a quadrotor helicopter," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 3255—-3260.

[19] R. Murray, Z. Li, and S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[20] F. Ruggiero, J. Cacace, H. Sadeghian, and V. Lippiello, "Impedance control of VToL UAVs with a momentum-based external generalized forces estimator," in *2014 IEEE International Conference on Robotics and Automation*, 2014, pp. 2093–2099.

[21] T. Lee, M. Leok, and N. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *9th IEEE Conference on Decision and Control*, 2010, pp. 5420–5425.

[22] F. Ferraguti, C. Secchi, and C. Fantuzzi, "A tank-based approach to impedance control with variable stiffness," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4948–4953.

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