Teleoperation of a Suspended Aerial Manipulator Using a Handheld Camera with an IMU

Miguel Arpa Perozo, Ethan Niddam, Loïc Cuvillon, Sylvain Durand, Jacques Gangloff

Teleoperation of a Suspended Aerial Manipulator Using a Handheld Camera with an IMU

Miguel Arpa Perozo, Ethan Niddam, Loïc Cuvillon, Sylvain Durand and Jacques Gangloff

***Abstract*— This paper presents a simple, low-cost teleoperation system. The leader device is a handheld camera integrated with an Inertial Measurement Unit (IMU), making it feasible to use a modern smartphone for this purpose. Existing leader devices require expensive and/or complex hardware, and sensors to measure both the user interactions and to control the follower device. By contrast, the proposed method uses the handheld camera both as a leader device and as a sensor to control the position of the follower device through visual servoing. To the best of the author's knowledge, this visual servoing scenario where the camera is held by a user has not been thoroughly studied. The measurements from the handheld device and the follower are fused together in an Extended Kalman Filter (EKF) to improve further the pose estimation. A Virtual Camera and IMU (VCI) concept is introduced to filter hand tremors for teleoperation efficiency without hindering the bandwidth of the relative pose control loop. The EKF and the VCI performance are assessed experimentally by teleoperating a Suspended Aerial Manipulator (AMES) prototype.**

I. Introduction

Teleoperation makes it possible to combine human cognitive capabilities with the precision, repeatability and strength of robots. The large workspace of aerial vehicles makes them ideal for teleoperation tasks in hard-to-access locations, or tasks inherently requiring large workspaces like the inspection of industrial or commercial buildings [1]. Omnidirectional aerial vehicles, capable of exerting thrust in any direction, offer superior maneuverability compared to conventional underactuated systems, making them highly suitable for tasks requiring physical interaction [2], [3]. Suspended aerial manipulators compensate for the aerial vehicle's weight, thus increasing their autonomy, and payload, though at the cost of reducing the system workspace [4], [5]. Currently, suspended aerial manipulators prototypes use omnidirectional aerial vehicles to achieve dexterous tasks. The aerial vehicle can be suspended to a crane with a cable robot like the SAM robot [4], or to a cable-driven parallel robot with an elastic link, like the Aerial Manipulator with Elastic Suspension (AMES) [5], [6].

Teleoperation systems are usually composed of a leader and a follower devices [7]. The operator interacts with the leader device. This interaction is measured and used to generate a reference which is sent to the internal controller of the follower device. Several leader devices can be found in the literature for the teleoperation of omnidirectional aerial vehicles. A 7 Degrees of Freedom (DoF) Franka Emika Panda arm robot is used as a haptic device for remote bilateral teleoperation of a tiltrotor omnidirectional aerial vehicle in [8], where a push-and-slide experiment is carried. Pose of the vehicle is provided by a motion capture system. A 2-DoF force feedback joystick is used as a leader device for remote bilateral teleoperations of the SAM robot in [9] and [10]. The operator in front of a 2D screen display manually switches between different tasks to compensate for the reduced number of DoFs of the leader device [10]. Finally, a novel human-robot interface using mixed reality head-mounted display for teleoperating omnidirectional aerial vehicles is developed in [11]. Mixed reality presents the advantage of not being limited by the hardware constraints of traditional teleoperation interfaces like joysticks, gives a better spatial awareness to the user compared to a 2D camera feed, and enables the operator to independently control all the DoFs of the follower manipulator. This approach is limited to line-of-sight operations where the user can distinguish relevant details in the environment, and has only been evaluated on simulation.

In this paper, a handheld device consisting of a camera and a nine[1] DoF Inertial Measurement Unit (IMU), is proposed as a leader device to teleoperate a 6-DoF system, with an application to an AMES here in particular. The working principle is illustrated in Fig. 1. The pose of the AMES is controlled using Position Based Visual Servoing (PBVS) [12] to maintain a constant pose between the robot and the handheld device. Hence, an operator is able to control the pose of the robot by moving the camera safely at a distance.

The main advantage of the proposed teleoperation framework is its low-cost and simplicity. Since only a camera equipped with an IMU is needed, a modern smartphone could be used as a leader device. Compared to the aforementioned teleoperation setups, the same device is used simultaneously as a leader interface and as a pose sensor of the follower device. The proposed teleoperation interface is ergonomic and intuitive to use. There is no need to generate a reference for the follower device, since this reference is constant and corresponds to the initial relative pose measured by the camera. The operator is not restrained by mechanical linkages, and the natural movements of the hand are replicated by the robot. However, our solution presents some limitations similar to [11]: (i) lack of force feedback from the handheld camera, (ii) teleoperation limited to line-of-sight tasks. Targeted applications are primarily visual inspection (as in [11]), tracking shots for the cinema, and spray-painting of structures [13]. Handling applications in the construction industry are also possible, allowing an operator to precisely and effortlessly move heavy loads with an aerial platform

[1]With a three axis gyroscope, three axis magnetometer, and three axis accelerometer

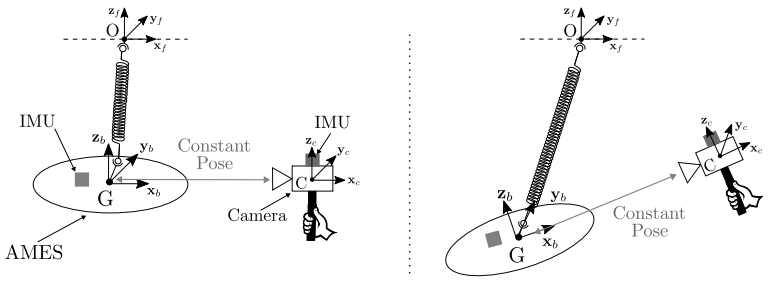

Fig. 1: Illustration of the visual servoing of an AMES using a handheld camera with an IMU.

suspended from a construction crane. All do not require a force feedback.

Many works have proposed to use a camera as a primary interface [14], [15], [16]. However to the extent of the author's knowledge, the visual servoing scenario where the camera is held by the user to directly control the robot in its field of view has not been studied. The handheld camera acts simultaneously as a user interface and as the sensor of the robot pose controller.

The presented method does not make any assumption on the type of robot, it can be used to control any 6-DoF robot effector. Nevertheless, this is particularly useful for robots without accurate proprioceptive position sensors, like aerial vehicles. This approach has the benefit to add at a low cost an external exteroceptive localization device which can also be used as a teleoperation user interface. Moreover, it can easily be extended to teleoperate underactuated systems such as quadrotors or mobile robots by controlling only their limited DoF with the camera.

Our paper is organized as follows. In Section II, the relative dynamic model of the system is derived. The model is used in Section III where an EKF is developed for state estimation and sensor fusion, and the PBVS controller is presented. The concept of Virtual Camera and IMU (VCI) is introduced in Section IV to filter out hand tremors without hindering the bandwidth of the relative pose control loop. Experimental results are presented in Section V. Conclusion and future work are discussed in Section VI.

II. Model

A. Notations and Preliminaries

1) Notations: Vectors and matrices are represented by bold lowercase and bold uppercase letters respectively. An upper script is used to indicate the coordinate frame the vector is projected in, i.e., ${}^a\boldsymbol{v}$ corresponds to the projection of the vector $\boldsymbol{v}$ in the coordinate frame $\mathscr{F}_a$. An arrow is used to describe a vector built from geometrical points: the vector from point A to point B is noted $\overrightarrow{AB}$. The cross product matrix is noted $[\cdot]_\times$ such that $\boldsymbol{c} = \boldsymbol{a} \wedge \boldsymbol{b} = [\boldsymbol{a}]_\times \cdot \boldsymbol{b}$.

Let us consider two coordinate frames $\mathscr{F}_a$ and $\mathscr{F}_b$. The rotation matrix from $\mathscr{F}_a$ to $\mathscr{F}_b$ is noted ${}^a\boldsymbol{R}_b$ such that ${}^a\boldsymbol{v} = {}^a\boldsymbol{R}_b \cdot {}^b\boldsymbol{v}$. Homogeneous coordinate vectors are noted with a tilde: ${}^a\tilde{\mathbf{p}} = \begin{bmatrix} {}^a\mathbf{p}^T & 1 \end{bmatrix}^T$. The homogeneous transformation matrix from $\mathscr{F}_a$ to $\mathscr{F}_b$ is noted ${}^a\boldsymbol{T}_b$ such that ${}^a\tilde{\boldsymbol{v}} = {}^a\boldsymbol{T}_b \cdot {}^b\tilde{\boldsymbol{v}}$. The angular velocity vector of $\mathscr{F}_b$ w.r.t. $\mathscr{F}_a$ is noted $\boldsymbol{\omega}_{b/a}$. Let $\boldsymbol{v}$ be a vector, its time derivative w.r.t. a frame $\mathscr{F}_a$ is noted $\frac{{}^a d\boldsymbol{v}}{dt}$.

2) Preliminaries: Let $\mathscr{F}_a$ and $\mathscr{F}_b$ be two coordinate frames, $\boldsymbol{v} \in \mathbb{R}^3$ a vector, and $\boldsymbol{\omega}_{b/a}$ the rotational speed of $\mathscr{F}_b$ w.r.t. $\mathscr{F}_a$. The time derivative of $\boldsymbol{v}$ in $\mathscr{F}_a$ is related to the time derivative of $\boldsymbol{v}$ in $\mathscr{F}_b$ by the following equation [17, Section 7.1.1]:

$$\frac{{}^a d\boldsymbol{v}}{dt} = \frac{{}^b d\boldsymbol{v}}{dt} + \boldsymbol{\omega}_{b/a} \wedge \boldsymbol{v} \tag{1}$$

The orientation of $\mathscr{F}_b$ w.r.t. a frame $\mathscr{F}_a$ is parameterized by three Cardanian angles "roll, pitch and yaw" using the ZYX convention [18]: $\boldsymbol{\eta}_{b/a} = [\theta_r,\ \theta_p,\ \theta_y]$. The relationship between $\boldsymbol{\omega}_{b/a}$ and the time derivative of ${}^a\boldsymbol{R}_b$ is the following [18]:

$$\left[{}^a\boldsymbol{\omega}_{b/a}\right]_\times = {}^a\dot{\boldsymbol{R}}_b \cdot {}^a\boldsymbol{R}_b^T \tag{2}$$

From (2) we derive the expression for the analytical Jacobian ${}^a\boldsymbol{S}(\boldsymbol{\eta}_{b/a})$ which relates the time derivative of $\boldsymbol{\eta}_{b/a}$ to the angular velocity $\boldsymbol{\omega}_{b/a}$ (see [19, Section 3.6]):

$${}^a\boldsymbol{\omega}_{b/a} = {}^a\boldsymbol{S}(\boldsymbol{\eta}_{b/a}) \cdot \dot{\boldsymbol{\eta}}_{b/a} \tag{3}$$

B. System Parametrization

Model parameters of the AMES using a handheld camera with an IMU can be found in Fig. 1. In total, three coordinate frames are considered: first, the fixed inertial frame $\mathscr{F}_f \{\boldsymbol{x}_f, \boldsymbol{y}_f, \boldsymbol{z}_f\}$ centered at O, second, the body frame attached to the CoM G of the aerial vehicle $\mathscr{F}_b \{\boldsymbol{x}_b, \boldsymbol{y}_b, \boldsymbol{z}_b\}$, and third, the camera frame $\mathscr{F}_c \{\boldsymbol{x}_c, \boldsymbol{y}_c, \boldsymbol{z}_c\}$ centered at C. The camera optical axis is aligned with $\boldsymbol{x}_c$. Due to the small size of the handheld device, and without loss of generality, it is assumed here that camera and IMU coordinate frames coincide.

In addition to the IMU attached to the camera, another one is embedded in the aerial vehicle. The IMU attached to the camera is assumed to be a 9-DoF IMU providing an accurate estimate of its orientation w.r.t. the inertial frame $\mathscr{F}_f$ [20], [21]. Let us define:

$$\boldsymbol{p}_{b/f} = \overrightarrow{OG}, \quad \boldsymbol{p}_{c/f} = \overrightarrow{OC}, \quad \boldsymbol{p}_{b/c} = \overrightarrow{CG} \tag{4}$$

Throughout this document, a parameter is referred to as *absolute* when defined w.r.t. $\mathscr{F}_f$ and *relative* when it is defined w.r.t. $\mathscr{F}_c$. Hence, $\boldsymbol{p}_{b/f}$ is called the absolute position of the robot, whereas $\boldsymbol{p}_{b/c}$ designates its relative position. The goal

of this section is to determine the relative acceleration of the AMES w.r.t. an accelerated (thus non-inertial) camera frame $\mathscr{F}_c$. These quantities are used for the state estimator presented in Section III. Since we are interested in relative velocities and accelerations w.r.t. $\mathscr{F}_c$, the conventional dot notation used to designate the time derivative of a vector will be avoided, whenever possible, as it does not indicate w.r.t. which frame the vector is being differentiated. To avoid potential confusions, a naming convention is used to describe absolute and relative velocities and accelerations. For the linear velocities, the velocity of the body i at its CoM w.r.t. the frame $\mathscr{F}_j$ is noted:

$$\boldsymbol{v}_{i/j} = \frac{^j d\, \boldsymbol{p}_{i/j}}{dt} \tag{5}$$

Similarly, the translational and rotational accelerations of the body i at its CoM w.r.t. to the coordinate frame $\mathscr{F}_j$ are defined as:

$$\boldsymbol{a}_{i/j} = \frac{^j d\, \boldsymbol{v}_{i/j}}{dt}, \quad \boldsymbol{\delta}_{i/j} = \frac{^j d\, \boldsymbol{\omega}_{i/j}}{dt} \tag{6}$$

C. Relative Dynamics

From the rotational speed composition rule, we have the following equality:

$$\boldsymbol{\omega}_{b/c} = \boldsymbol{\omega}_{b/f} - \boldsymbol{\omega}_{c/f} \tag{7}$$

By using (1), (5), and (6), the relationship between absolute and relative acceleration can be established. Due to space limitations only the results are given below, for more information we refer the reader to [17, Section 4.3.2].

$$\begin{aligned} \boldsymbol{a}_{b/c} = \boldsymbol{a}_{b/f} - \boldsymbol{a}_{c/f} - 2\left(\boldsymbol{\omega}_{c/f} \wedge \boldsymbol{v}_{b/c}\right) - \\ \boldsymbol{\delta}_{c/f} \wedge \boldsymbol{p}_{b/c} \; - \boldsymbol{\omega}_{c/f} \wedge \left(\boldsymbol{\omega}_{c/f} \wedge \boldsymbol{p}_{b/c}\right) \end{aligned} \tag{8}$$

III. State Estimation and Control

A. Sensor Models

From the observability studies in [22], [23], it is known that for an IMU attached to a camera, all the IMU biases are observable as long as there is a non-biased position or velocity measurement available from the camera image processing. However, our case is different because there are two IMUs and one camera. The camera IMU is assumed to be of high quality and finely calibrated such that the biases can be neglected. IMUs embedded in aerial vehicles are usually more low-cost; thus they are more prone to biases. Therefore, the body IMU measurements are assumed to be biased and disturbed by additive white Gaussian noises. Let us define:

$$\boldsymbol{\omega}_{b/f} = \boldsymbol{\omega}_{m,b/f} - \boldsymbol{b}_\omega - \boldsymbol{n}_\omega \qquad \boldsymbol{a}_{b/f} = \boldsymbol{a}_{m,b/f} - \boldsymbol{b}_a - \boldsymbol{n}_a \tag{9}$$

where $\boldsymbol{\omega}_{m,b/f}$ and $\boldsymbol{a}_{m,b/f}$ denote the biased vehicle gyroscope and accelerometer measurements, $\boldsymbol{b}_\omega$ and $\boldsymbol{b}_a$ are the gyroscope and accelerometer biases respectively, $\boldsymbol{n}_\omega$ and $\boldsymbol{n}_a$ represent white Gaussian noises. The biases $\boldsymbol{b}_\omega$ and $\boldsymbol{b}_a$ are considered constant along time ($\dot{\boldsymbol{b}}_\omega = \dot{\boldsymbol{b}}_a = \boldsymbol{0}$)

B. State Representation

Following the reasoning presented in [22], the system model for the Kalman filter considers the aerial vehicle and the camera as free-floating 6-DoF bodies. In this approach, the vehicle dynamic model and the propeller speed measurement are not exploited to augment further the EKF state-space model. The filtering and estimation rely solely on the pose time derivatives provided by the sensors (IMUs and camera). As pointed in [22, Chapter 3], this modeling approach has the following advantages: (i) the system state vector is smaller, (ii) thereby the EKF implementation is less computationally expensive when compared to an implementation that includes the full model, (iii) there is no need to identify the body model parameters: mass, inertia, and the speed-to-thrust coefficient. The system state, of dimension 30 , is:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p}_{b/c}^T & \boldsymbol{\eta}_{b/c}^T & \boldsymbol{v}_{b/c}^T & \boldsymbol{\omega}_{b/f}^T & \boldsymbol{\omega}_{c/f}^T & \boldsymbol{a}_{b/f}^T & \boldsymbol{a}_{c/f}^T & \boldsymbol{\delta}_{c/f}^T & \boldsymbol{b}_\omega^T & \boldsymbol{b}_a^T \end{bmatrix}^T \tag{10}$$

C. State Dynamics and Measurements

As it is common in pose estimation and filtering [24], the inertial measurements $\boldsymbol{\omega}_{b/f}$, $\boldsymbol{a}_{b/f}$, and $\boldsymbol{a}_{c/f}$, are assumed to be constant between two sampling points, thus their time derivatives are considered null, except for the camera gyroscope measurement $\boldsymbol{\omega}_{c/f}$. Indeed, the camera angular acceleration $\boldsymbol{\delta}_{c/f}$ is required in (8) to estimate the relative acceleration $\boldsymbol{a}_{b/c}$ through the fusion of the camera and IMU measurements. Thus, the angular acceleration is included in the state vector and estimated by the EKF based on velocity measurements.

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{v}_{b/c}^T & \dot{\boldsymbol{\eta}}_{b/c}^T & \boldsymbol{a}_{b/c}^T & \boldsymbol{0}^T & \boldsymbol{\delta}_{c/f}^T & \boldsymbol{0}^T & \boldsymbol{0}^T & \boldsymbol{0}^T & \boldsymbol{0}^T & \boldsymbol{0}^T \end{bmatrix}^T \tag{11}$$

where $\boldsymbol{a}_{b/c}$ is given by (8), $\dot{\boldsymbol{\eta}}_{b/c}$ from (3) and (7), and $\boldsymbol{\delta}_{c/f}$ is part of the state vector $\boldsymbol{x}$.

From the handheld IMU absolute orientation $\boldsymbol{\eta}_{c/f}$ and the body relative orientation from the camera $\boldsymbol{\eta}_{b/c}$, the body absolute orientation $\boldsymbol{\eta}_{b/f}$ is obtained:

$$^f\boldsymbol{R}_b(\boldsymbol{\eta}_{b/f}) = {}^f\boldsymbol{R}_c(\boldsymbol{\eta}_{c/f}) \cdot {}^b\boldsymbol{R}_c(\boldsymbol{\eta}_{b/c})^{-1} \tag{12}$$

Knowing $\boldsymbol{\eta}_{c/f}$ and $\boldsymbol{\eta}_{b/f}$, the gravity component is subtracted from the accelerometer measurements such that the EKF measurement vector is:

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{p}_{b/c}^T & \boldsymbol{\eta}_{b/c}^T & \boldsymbol{\omega}_{m,b/f}^T & \boldsymbol{\omega}_{c/f}^T & \boldsymbol{a}_{m,b/f}^T & \boldsymbol{a}_{c/f}^T \end{bmatrix}^T \tag{13}$$

The relative position $\boldsymbol{p}_{b/c}$ and orientation $\boldsymbol{\eta}_{b/c}$ are estimated from the camera by detecting the image of Light Emitting Diodes (LEDs) attached to the AMES and applying the Perspective-n-Point (PnP) algorithm from [25]. The current EKF implementation does not handle occlusion of the visual markers by relying only on inertial data. A prolonged occlusion will destabilize the system with a drift of the AMES position. However, this limitation is not inherent to our setup. The same problem occurs if the measurements from a motion capture system or a GPS are lost.

D. Control

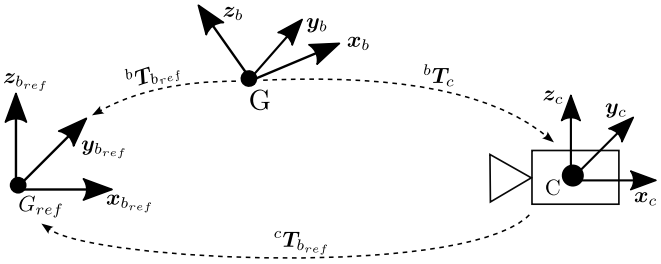

Fig. 2: Pose error illustration.

A controller is designed to control the body frame $\mathscr{F}_b$ such that its pose w.r.t. a moving camera frame $\mathscr{F}_c$ is constant. Because $\mathscr{F}_b$ is regulated w.r.t. to $\mathscr{F}_c$, all the control errors must be defined w.r.t. $\mathscr{F}_c$.

The relative nominal reference pose ${}^c\boldsymbol{T}_{b_{ref}}$ between the camera and the robot is assumed to be constant and equal to the first ${}^c\boldsymbol{T}_b$ measurement at the beginning of the teleoperation. The relative pose ${}^c\boldsymbol{T}_b$ is measured by detecting visual markers placed in the AMES and using the Perspective-n-Point (PnP) algorithm from [25]. Given the available measurements, the pose error ${}^b\boldsymbol{T}_{b_{ref}}$, illustrated in Fig. 2, is defined by:

$$ {}^b\boldsymbol{T}_{b_{ref}} = {}^c\boldsymbol{T}_b^{-1} \cdot {}^c\boldsymbol{T}_{b_{ref}} \tag{14} $$

From ${}^b\boldsymbol{T}_{b_{ref}}$ a position and orientation error can be extracted:

$$ \boldsymbol{e}_p = \overrightarrow{GG_{ref}}, \qquad \boldsymbol{e}_\eta = \boldsymbol{\eta}_{b_{ref}/b} \tag{15} $$

where $\boldsymbol{\eta}_{b_{ref}/b}$ are the Cardanian angles "roll, pitch yaw" between $\mathscr{F}_b$ and $\mathscr{F}_{b_{ref}}$. The total pose error is noted $\boldsymbol{e}$ and equal to: $\boldsymbol{e} = \begin{bmatrix} \boldsymbol{e}_p^T & \boldsymbol{e}_\eta^T \end{bmatrix}^T$.

From (3), we deduce:

$$ \dot{\boldsymbol{e}}_\eta = {}^b\boldsymbol{S}(\boldsymbol{e}_\eta)^{-1} \cdot {}^b\boldsymbol{\omega}_{b_{ref}/b} \tag{16} $$

From the composition rule of rotational velocities, we have:

$$ \boldsymbol{\omega}_{b_{ref}/b} = \boldsymbol{\omega}_{b_{ref}/c} + \boldsymbol{\omega}_{c/f} + \boldsymbol{\omega}_{f/b} \tag{17} $$

The reference relative pose between the camera and the aerial vehicle is supposed to be constant, thus (16) becomes:

$$ \dot{\boldsymbol{e}}_\eta = {}^b\mathbf{S}(\boldsymbol{e}_r)^{-1} \cdot \left({}^b\boldsymbol{\omega}_{c/f} - {}^b\boldsymbol{\omega}_{b/f} \right) \tag{18} $$

Similarly, the position error derivative is equal to:

$$ \dot{\boldsymbol{e}}_p = \frac{{}^c d\, \boldsymbol{e}_p}{dt} = \frac{{}^c d \left(\overrightarrow{CG_{ref}} - \overrightarrow{CG} \right)}{dt} = -\, \boldsymbol{v}_{b/c} \tag{19} $$

A PID controller is considered to validate the EKF estimation and the virtual camera and IMU presented in Section IV. The considered PID control law is:

$$ \boldsymbol{\tau} = \boldsymbol{K}_p \cdot \boldsymbol{e} \quad + \quad \boldsymbol{K}_i \cdot \int \boldsymbol{e}\, dt \quad + \quad \boldsymbol{K}_d \cdot \dot{\boldsymbol{e}} \tag{20} $$

where $\boldsymbol{K}_p$, $\boldsymbol{K}_i$, $\boldsymbol{K}_d \in \mathbb{R}^{6\times6}$ are symmetric positive definite gain matrices. The wrench $\boldsymbol{\tau}$ is then allocated among the propeller thrust of the AMES using the thrust allocation techniques presented in [26], [6].

IV. Virtual Camera and IMU

A. Motivations

The distance between the camera and the aerial vehicle creates a *lever-arm effect* where a small camera rotation induces an important robot translation as illustrated in Fig. 3. This problem occurs for rotations of the camera around the two axes perpendicular to its optical axis. This teleoperation behavior is not desirable. A small, possibly unwanted, rotation of the operator hand holding the camera will result in a high-speed translation of the AMES. The user hand tremors are also amplified by the distance between the camera and the robot. These vibrations must be filtered out without compromising the controller performance w.r.t. disturbance rejection, like wind bursts. The system must have a fast relative pose controller, to efficiently reject disturbances, while filtering out the camera absolute rotations so that hand tremors are not directly fed in the relative pose controller.

The proposed solution consists in expressing all the measurements w.r.t. a Virtual Camera and IMU (VCI) coordinate frame $\mathscr{F}_v$ which has the same origin as $\mathscr{F}_c$ but whose attitude is low-pass filtered. The robot relative pose is then regulated w.r.t. $\mathscr{F}_v$ instead of $\mathscr{F}_c$ as illustrated in Fig. 3. Because $\mathscr{F}_v$ is obtained by low-pass filtering the absolute attitude of $\mathscr{F}_c$, hand tremors from the user are filtered out, and the lever-arm effect is attenuated. Lastly, because only the absolute attitude of $\mathscr{F}_c$ is filtered acting as a reference pre-filter, the bandwidth of the relative pose control loop is unmodified. Any disturbance on the robot is directly measured by the relative pose measurements from the camera and robot IMU with no low-pass filtering involved.

B. Implementation

Let $\mathscr{F}_v$ be a coordinate frame which has the same origin as $\mathscr{F}_c$ (see Fig. 3) but whose attitude is obtained by low-pass filtering the attitude of $\mathscr{F}_c$ such that:

$$ \boldsymbol{\eta}_{v/f} = \mathrm{LP}(\boldsymbol{\eta}_{c/f}) \tag{21} $$

where the $\mathrm{LP}(\cdot)$ operator returns the low-pass filtered roll, pitch and yaw angles describing the absolute orientation of $\mathscr{F}_c$. Below, all the measurements from the primary device (camera and its IMU) are expressed w.r.t. $\mathscr{F}_v$. We recall that the primary device measurements are: ${}^c\boldsymbol{p}_{b/c}$, $\boldsymbol{\eta}_{b/c}$, ${}^c\boldsymbol{\omega}_{c/f}$, ${}^c\boldsymbol{a}_{c/f}$.

By combining the camera measurements $\boldsymbol{\eta}_{b/c}$, the hand-held IMU attitude $\boldsymbol{\eta}_{c/f}$, and $\boldsymbol{\eta}_{v/f}$ given by (21), it is possible to obtain the attitude between the body and the virtual camera frame $\boldsymbol{\eta}_{b/v}$:

$$ {}^v\boldsymbol{R}_b(\boldsymbol{\eta}_{b/v}) = {}^v\boldsymbol{R}_f(\boldsymbol{\eta}_{v/f}) \cdot {}^f\boldsymbol{R}_c(\boldsymbol{\eta}_{c/f}) \cdot {}^c\boldsymbol{R}_b(\boldsymbol{\eta}_{b/c}) \tag{22} $$

The rotational speed of $\mathscr{F}_v$ is obtained by tacking the time derivative of $\boldsymbol{\eta}_{v/f}$ and using (3). In practice, taking the time derivative of $\boldsymbol{\eta}_{v/f}$ is not a problem because $\boldsymbol{\eta}_{v/f}$ is a smooth low-pass filtered signal. Since $\mathscr{F}_v$ and $\mathscr{F}_c$ have the same

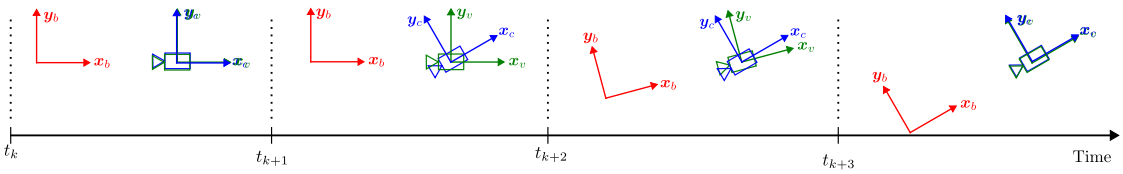

Fig. 3: Teleoperation using a virtual camera and IMU. A small rotation of the handheld camera (blue frame $\mathscr{F}_c$) induces a large translation of the robot body frame (in red $\mathscr{F}_b$) to control a constant pose between both frames. To slow down the induced translation and filter the hand tremors, the robot is controlled w.r.t. the virtual camera frame $\mathscr{F}_v$ depicted in green.

origin, $\boldsymbol{a}_{v/f}$ and $\boldsymbol{a}_{c/f}$ are equal. All in all, the measurements from the VCI are:

$$ {}^{v}\boldsymbol{p}_{b/v} = {}^{v}\boldsymbol{R}_{c} \cdot {}^{c}\boldsymbol{p}_{b/c}, \qquad {}^{v}\boldsymbol{a}_{v/f} = {}^{v}\boldsymbol{R}_{c} \cdot {}^{c}\boldsymbol{a}_{c/f} \tag{23} $$

$$ {}^{v}\boldsymbol{\omega}_{v/f} = {}^{v}\boldsymbol{R}_{f} \cdot {}^{f}\boldsymbol{S}(\boldsymbol{\eta}_{v/f}) \cdot \dot{\boldsymbol{\eta}}_{v/f} \tag{24} $$

The goal is for the robot to keep a constant pose w.r.t. $\mathscr{F}_v$ as illustrated in Fig. 3. As a consequence, the control error and state estimation presented in previous sections are now defined w.r.t. $\mathscr{F}_v$ instead of $\mathscr{F}_c$. Because only the camera's absolute orientation $\boldsymbol{\eta}_{c/f}$ is low-pass filtered, the VCI does not add any delay or phase lag into the relative pose control loop. The control block diagram with all the elements: controller, state estimation, and VCI is given in Fig. 4.

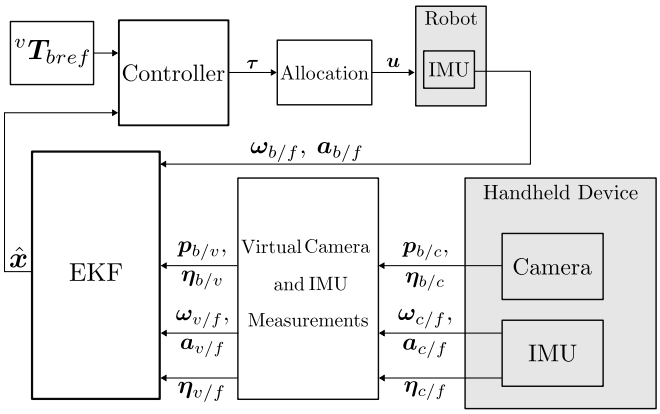

Fig. 4: Teleoperation control block diagram.

V. Experimental Application

A. Experimental Setup

The main components of the experimental setup can be seen in Fig. 5: an AMES developed in [6], a handheld device combining a camera and an IMU, and a UR5e robot arm used to simulate a human operator in a repeatable way. For most of the experiments, the distance between the robot and the camera is approximately 2.5 m. The sampling frequency of the controller, and EKF is 200 Hz.

A total of five LEDs, used as visual markers, are attached to the arms of the AMES. The LEDs are positioned to maximize their visibility from different viewpoints, and to ensure that at least four LEDs are visible. With four detected points, the PnP problem has a unique solution [25].

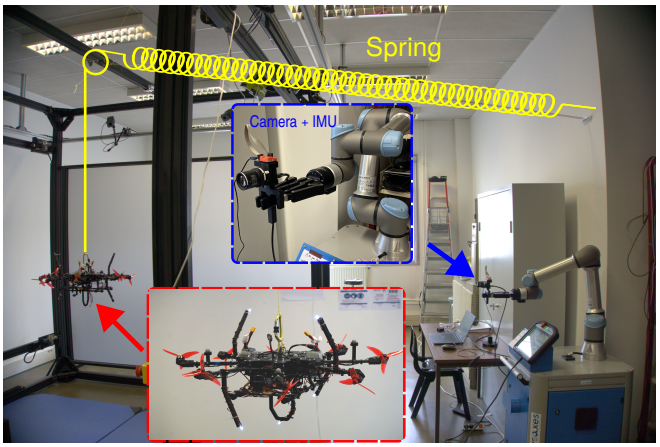

Fig. 5: Overview of the experimental setup.

The handheld device is connected through USB to a ground station: a standard Dell laptop running Ubuntu 20 (Intel i5-2.5 GHz with 16 GiB of RAM). On this ground station, a program performs the image processing that detects the LEDs, runs the PnP algorithm, reads the IMU data, and packages all the information that is sent over WiFi to the AMES embedded computer. The image processing is done using the C++ implementation of OpenCV version 4.5. The AMES on-board computer (NVIDIA Jetson Xavier NX) handles the control algorithms, sensor fusion, and communicates with the ground station through Wi-fi TCP/IP sockets thanks to the open-source Simulink toolbox RPIt developed in our laboratory [27].

The handheld device is made of a CMOS, USB, high-speed 500 fps camera *Ximea-MQ003MG-CM*. A Tamron C-Mount lens *12VM1040ASIR* with adjustable focal length from 10 to 40 mm is mounted on the camera. The handheld device is also equipped with a 9-DoF IMU, a Xsens *MTI-630* running at 400 Hz.

B. Extended Kalman Filter Influence

The system performance is assessed first by comparing its precision, then by comparing the execution time of a telemanipulation task performed by an operator, with and without the EKF. The EKF nonlinear state model presented in (10), and (11) is discretized using a classical Runge-Kutta method, and implemented using the Matlab Control System Toolbox. In these first experiments, the filtering of the camera attitude is disabled.

1) Performance Assessment: The camera is moved in a repeatable way by a robot arm. The robot arm executes a pure translation of the camera along the three axes of $\mathscr{F}_c$ with an amplitude of 3 cm in 0.3 s. The absolute displacement of the camera w.r.t. its initial position is noted:

$$\boldsymbol{\Delta p}_{c/f} = \boldsymbol{p}_{c/f} - \boldsymbol{p}_{c/f,init} \tag{25}$$

where $\boldsymbol{p}_{c/f,init}$ is the initial absolute position of the camera at the beginning of the experiment.

The controller gains are available in Table II. They were tuned experimentally to minimize the settling time of the system, and its sensitivity to measurement noise. Since the measurement without EKF is much noiser, the derivative and proportional gains could not be increased too much without yielding unacceptable vibrations of the system. This yields a less damped response which is the best tradeoff that we could find with minimal degradation of telemanipulation user experience.

The RMS of the controller errors for the translational DoFs are available in Table I. It can be seen that the EKF increases the system precision, especially for camera movements along its optical axis $\boldsymbol{x}_c$. Indeed, the PnP reconstruction is known to be more sensitive to image noise in this direction, and the fusion with accelerometer data in the EKF helps decreasing this sensitivity. The error signal on the AMES position and the controller output can be found in Fig. 6 where the camera moves along its optical axis $\boldsymbol{x}_c$. The EKF fuses and filters the camera and inertial measurements, increasing the signal-to-noise ratio on the relative pose estimation, and particularly the relative position along the camera optical axis as shown in Fig. 6, as a consequence, it is possible to have higher gains for the controller (see Table II), thus increasing the system precision.

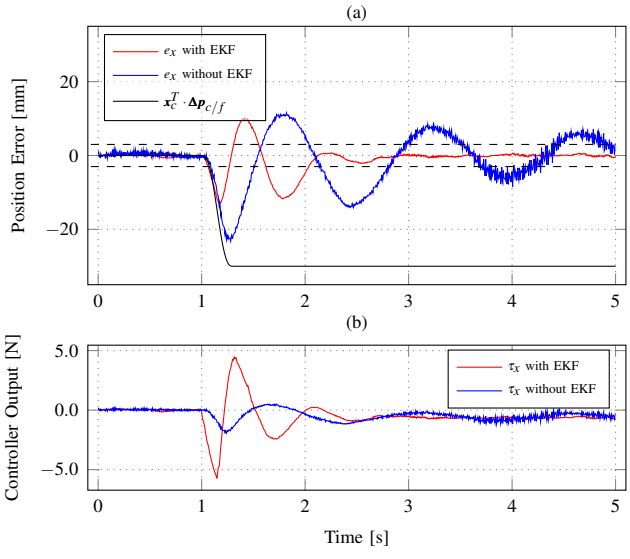

Fig. 6: Relative pose error along $\boldsymbol{x}_b$ axis (a), for a camera movement along its optical axis $\boldsymbol{x}_c$ represented by the black solid line. The black dashed lines represent the 10 % settling time limit. Corresponding controller output for the AMES thrust along $\boldsymbol{x}_b$ axis are shown in (b).

	Without EKF			With EKF		
	x	y	z	x	y	z
RMS Error [mm]	7.11	5.76	4.54	3.65	4.31	4.09

TABLE I: Controller RMS error comparison with and without EKF.

	x	y	z	roll	pitch	yaw
K_p	100	100	200	5	5	5
K_i	150	150	300	20	20	20
K_d	30	30	30	1	1	2

(a) With EKF.

	x	y	z	roll	pitch	yaw
K_p	30	50	100	5	5	2
K_i	90	150	200	15	15	5
K_d	10	15	15	0.5	0.5	1

(b) Without EKF.

TABLE II: Diagonal elements of the controller gains with and without EKF.

2) Telemanipulation Task: In this experiment, the system is tested with and without the EKF for a telemanipulation task. The task involves positioning and aligning a laser w.r.t. a tubular target so that the laser beam enters from one end and exits from the other tube end. The task is considered finished when the operator is able to maintain the laser beam through the target for 3 s. The operator must finish the task under 60 s, otherwise, the task is considered unsuccessful. The target and the task can be seen in the video[2] attached to this paper. The inner diameter of the tubular target is 30 mm and its axis has a 15 deg angle w.r.t. the vertical, to force the operator to alter the robot orientation to successfully complete the task.

The task is realized by two different operators. Each operator starts the test with a different flavor of the controller in order to detect and eliminate potential biases related to an effect of training with a first controller, yielding better results with the second controller. The tests are repeated five times with each controller flavor. The mean time to realize the task for both operators are available in Table III. Results indicate that the EKF reduces task time for both operators, regardless of the experiment order. The EKF improves the relative pose estimation which increases the accuracy of the controller as shown in the beginning of this section. As a consequence, it is easier for the operators to control the pose of the AMES and achieve the task.

C. Virtual Camera and IMU Experiments

For these experiments the robot arm is used to rotate the camera along an axis perpendicular to its optical axis, thus creating a "problematic" lever-arm situation. The experiments are conducted with and without considering the VCI, to quantify its influence on the system precision. The

[2] https://youtu.be/0LnaS2n7N4I?feature=shared

Operator	Without EKF [s]	With EKF [s]
1	34.93	21.89
2	18.51	13.89

TABLE III: Mean time to realize the task with and without EKF for operators 1 and 2.

disturbance rejection performance of the system when using the VCI is also tested.

A second order low-pass filter with time constants $\tau_1 = 3.18$ and $\tau_2 = 0.16$ is used to generate the VCI smooth motion. This corresponds to a -3 dB cutoff frequency of 0.05 Hz which is intentionally very low to help visualize the effects of the VCI during the experiments. Increasing the cut-off frequency will increase the reactivity of the system along its DoF susceptible to the lever-arm effect, but decreasing the frequency will slow down the robot, at the risk of losing the visual markers from the camera field of view. We have found experimentally that a cutoff frequency of 0.28 Hz allows for a comfortable teleoperation experience. It corresponds to the cutoff frequency used in the fifth experiment of the attached video where the operator performs a free flight. Because only small camera rotations are considered, $\boldsymbol{\omega}_{v/f}$ is obtained directly by low-pass filtering the camera gyroscope measurements. The experiments are carried out without the EKF. The gains of the controller are unmodified and available in Table IIb.

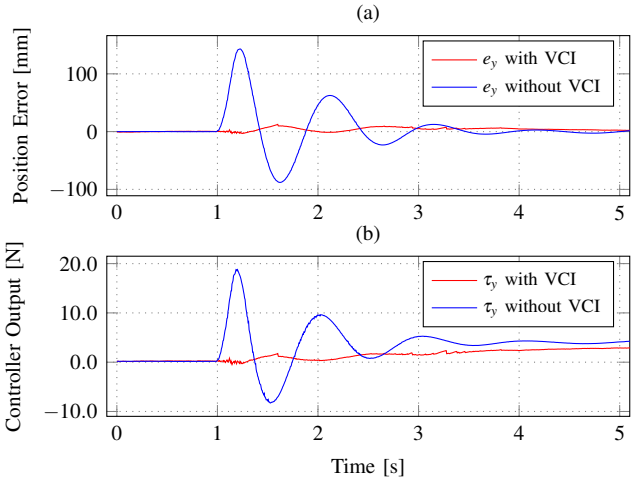

Fig. 7: Relative pose error (a) and controller output (b) with and without the VCI along $\boldsymbol{y}_b$ axis.

1) VCI Performance Evaluation: The robot arm is programmed to rotate the camera around its axis $\boldsymbol{z}_c$ by 3 deg in 0.3 s. This creates an important error on the y and yaw DoFs. The controller error and controller output signals are compared in Fig. 7. The RMS of the signal errors can be found in Table IV. From Fig. 7-b, it can be seen that the controller output does not tend towards zero. This is because the controller needs to compensate for wrench due to the spring restoring force.

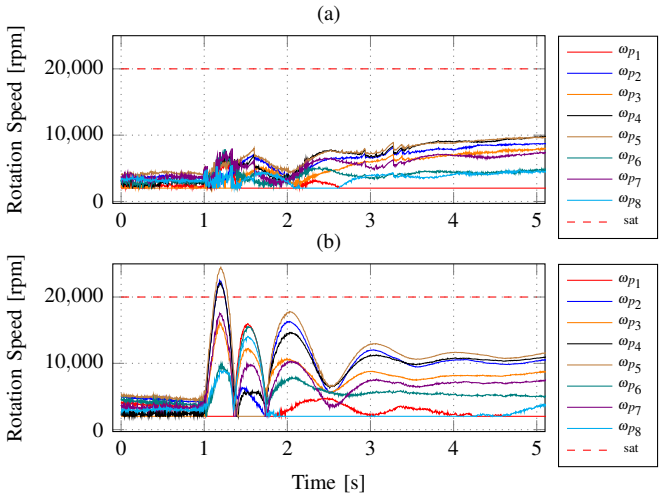

Fig. 8: Allocation output with (a) and without the VCI (b). The actuators saturation is depicted by a red dashed line.

Without the VCI, the pose error is defined w.r.t. to the reference frame $\mathcal{F}_{b_{ref}}$ which is rigidly linked to the camera frame $\mathcal{F}_c$ (see Fig. 2). The steep rotation of the camera induces a steep change of $\mathcal{F}_{b_{ref}}$ location. The corresponding transient tracking error is high and oscillating due partly to the saturation of the actuators (see Fig. 8). With the VCI implementation, the reference frame $\mathcal{F}_{b_{ref}}$ is rigidly linked to the virtual camera frame $\mathcal{F}_v$. Since the virtual camera attitude is the low-pass filtered attitude of the real camera, the error evolves slowly and the error can be efficiently regulated to zero without actuator saturation by the visual servoing controller. The position RMS error is only 4.34 mm with the VCI, compared to 32.9 mm without (see Table IV)

	Without VCI		With VCI	
	y	yaw	y	yaw
RMS Error [mm] and [deg]	32.92	0.84	4.34	0.21

TABLE IV: Controller error comparison with and without VCI.

2) Disturbance Rejection: This experiment shows that the low-pass filtering of the VCI does not affect the bandwidth of the relative pose control. Thereby, the same experiment is conducted as before, but a constant force disturbance offset of -4 N is applied at 5.6 s to the aerial vehicle using the thrusters. The disturbance is applied after the robot arm movement, but before the aerial vehicle reaches its final position. The relative position measured by the camera along $\boldsymbol{y}_c$ is available in Fig. 9; from this figure, two dynamics can be distinguished. A slower dynamic corresponding to the tracking of the slowly varying VCI, and a faster dynamic, between 5.5 s and 9 s, corresponding to the disturbance rejection dynamics of the relative pose controller.

VI. Conclusion

The teleoperation of an AMES using a handheld camera equipped with an IMU is considered in this paper. The

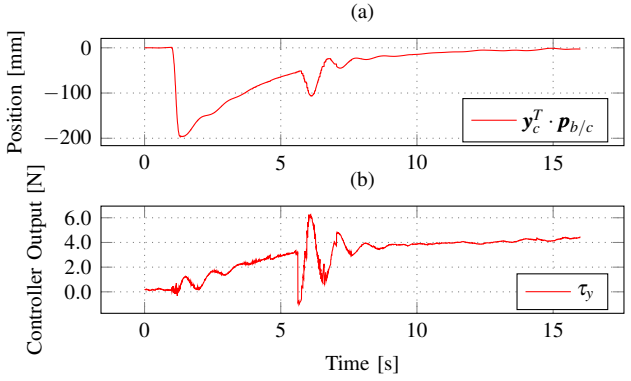

Fig. 9: Relative position of the aerial vehicle along $\boldsymbol{y}_c$ (a), and controller output (b) along $\boldsymbol{y}_b$. At $t = 1\,\text{s}$, the camera is rotated around $\boldsymbol{z}_c$. At $t = 5.6\,\text{s}$, a constant disturbance of $-4\,\text{N}$ is exerted along $\boldsymbol{y}_b$.

teleoperation framework allows for controlling intuitively, and at a safe distance, any 6-DoF of the robot effector. The performance of the teleoperation framework is improved by fusing the visual and inertial measurements through an EKF, and by introducing a virtual camera + IMU concept (VCI). The proposed VCI implementation filters hand motion and tremor to improve the teleoperation efficiency without hindering performance of the pose regulation and disturbance rejection. The performance improvement is quantified experimentally by teleoperating an AMES prototype with different scenarios.

Future work will focus on teleoperation tasks requiring physical contact. These will certainly require modifications by adding some compliance on the hardware and/or the control law to ensure stability. A formal stability analysis of the teleoperation framework will also be conducted. The VCI concept will be improved by selectively filtering the camera attitude only along the axes that cause a lever-arm effect. Finally, a smartphone will be used as a leader device.

References

[1] D. Lattanzi and G. Miller, "Review of Robotic Infrastructure Inspection Systems," *Journal of Infrastructure Systems*, vol. 23, no. 3, p. 04017004, 2017.

[2] M. Hamandi, F. Usai, Q. Sablé, N. Staub, M. Tognon, and A. Franchi, "Survey on Aerial Multirotor Design: A Taxonomy Based on Input Allocation," *The Int. Journal of Robotics Research*, p. 26, 2021.

[3] A. Ollero, M. Tognon, A. Suarez, D. Lee, and A. Franchi, "Past, Present, and Future of Aerial Robotic Manipulators," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 626–645, 2022.

[4] Y. S. Sarkisov, M. J. Kim, D. Bicego, D. Tsetserukou, C. Ott, A. Franchi, and K. Kondak, "Development of SAM: Cable-Suspended Aerial Manipulator," in *2019 Int. Conference on Robotics and Automation (ICRA)*, 2019, pp. 5323–5329.

[5] A. Yiğit, L. Cuvillon, M. A. Perozo, S. Durand, and J. Gangloff, "Dynamic Control of a Macro–Mini Aerial Manipulator With Elastic Suspension," *IEEE Transactions on Robotics*, pp. 1–17, 2023.

[6] M. A. Perozo, J. Dussine, A. Yiğit, L. Cuvillon, S. Durand, and J. Gangloff, "Optimal Design and Control of an Aerial Manipulator with Elastic Suspension Using Unidirectional Thrusters," in *2022 Int. Conference on Robotics and Automation (ICRA)*, 2022, pp. 1976–1982.

[7] S. N. Young and J. M. Peschel, "Review of Human–Machine Interfaces for Small Unmanned Systems With Robotic Manipulators," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 2, pp. 131–143, 2020.

[8] M. Allenspach, N. Lawrance, M. Tognon, and R. Siegwart, "Towards 6DoF Bilateral Teleoperation of an Omnidirectional Aerial Vehicle for Aerial Physical Interaction," in *2022 Int. Conference on Robotics and Automation (ICRA)*, 2022, pp. 9302–9308.

[9] J. Lee, R. Balachandran, Y. S. Sarkisov, M. De Stefano, A. Coelho, K. Shinde, M. J. Kim, R. Triebel, and K. Kondak, "Visual-Inertial Telepresence for Aerial Manipulation," in *2020 IEEE Int. Conference on Robotics and Automation (ICRA)*, 2020, pp. 1222–1229.

[10] A. Coelho, Y. Sarkisov, X. Wu, H. Mishra, H. Singh, A. Dietrich, A. Franchi, K. Kondak, and C. Ott, "Whole-Body Teleoperation and Shared Control of Redundant Robots with Applications to Aerial Manipulation," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 1, p. 14, 2021.

[11] M. Allenspach, T. Kötter, R. Bähnemann, M. Tognon, and R. Siegwart, "Design and Evaluation of a Mixed Reality-based Human-Robot Interface for Teleoperation of Omnidirectional Aerial Vehicles," in *2023 Int. Conference on Unmanned Aircraft Systems (ICUAS)*, 2023, pp. 1168–1174.

[12] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[13] E. Niddam, J. Dumon, L. Cuvillon, S. Durand, S. Querry, A. Hably, and J. Gangloff, "Design of a suspended manipulator with aerial elliptic winding," *IEEE Robotics and Automation Letters*, vol. 9, no. 9, pp. 7939–7946, 2024.

[14] S. Li, N. Hendrich, H. Liang, P. Ruppel, C. Zhang, and J. Zhang, "A Dexterous Hand-Arm Teleoperation System Based on Hand Pose Estimation and Active Vision," *IEEE Transactions on Cybernetics*, vol. 54, no. 3, pp. 1417–1428, 2024.

[15] L. Xie, D. Huang, Z. Lu, N. Wang, and C. Yang, "Handheld Device Design for Robotic Teleoperation based on Multi-Sensor Fusion," in *2023 IEEE Int. Conference on Mechatronics (ICM)*, 2023, pp. 1–6.

[16] D. Zhang, Y. Guo, J. Chen, J. Liu, and G.-Z. Yang, "A Handheld Master Controller for Robot-Assisted Microsurgery," in *2019 IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 394–400.

[17] R. E. Roberson and R. Schwertassek, *Dynamics of Multibody Systems*. Berlin, Heidelberg: Springer, 1988.

[18] P. Corke, *Robotics, Vision and Control*, ser. Springer Tracts in Advanced Robotics. Cham: Springer Int. Publishing, 2017, vol. 118.

[19] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics*, ser. Advanced Textbooks in Control and Signal Processing, M. J. Grimble and M. A. Johnson, Eds. London: Springer London, 2009.

[20] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear Complementary Filters on the Special Orthogonal Group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[21] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *2011 IEEE Int. Conference on Rehabilitation Robotics*, 2011, pp. 1–7.

[22] M. W. Achtelik, "Advanced closed loop visual navigation for micro aerial vehicles," Doctoral Thesis, ETH Zurich, 2014.

[23] A. Martinelli, A. Renzaglia, and A. Oliva, "Cooperative visual-inertial sensor fusion: Fundamental equations and state determination in closed-form," *Autonomous Robots*, vol. 44, no. 3-4, pp. 339–357, 2020.

[24] G. Huang, "Visual-Inertial Navigation: A Concise Review," in *2019 Int. Conference on Robotics and Automation (ICRA)*, 2019, pp. 9572–9582.

[25] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge: Cambridge University Press, 2004.

[26] M. Tognon and A. Franchi, "Omnidirectional Aerial Vehicles With Unidirectional Thrusters: Theory, Optimal Design, and Control," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2277–2282, 2018.

[27] J. Gangloff, A. Yiğit, and M. Lesellier, "RPIt," 2020. [Online]. Available: https://github.com/jacqu/RPIt/