



HAL
open science

A Force-based Bilateral Teleoperation Framework for Aerial Robots in Contact with the Environment

Guido Gioioso, Mostafa Mohammadi, Antonio Franchi, Domenico Prattichizzo

► **To cite this version:**

Guido Gioioso, Mostafa Mohammadi, Antonio Franchi, Domenico Prattichizzo. A Force-based Bilateral Teleoperation Framework for Aerial Robots in Contact with the Environment. 2015 IEEE Int. Conf. on Robotics and Automation, May 2015, Seattle, United States. hal-01135204

HAL Id: hal-01135204

<https://hal.science/hal-01135204>

Submitted on 25 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Force-based Bilateral Teleoperation Framework for Aerial Robots in Contact with the Environment

Guido Gioioso^{1,2}, Mostafa Mohammadi^{1,2}, Antonio Franchi^{3,4} and Domenico Prattichizzo^{1,2}

Abstract—In this paper a novel teleoperation framework for aerial robots that physically interact with the environment is presented. This framework allows to teleoperate the robot both in contact-free flight and in physical contact with the environment in order, e.g., to apply desired forces on objects of the environment. The framework is build upon an impedance-like indirect interaction force controller that allows to use standard underactuated aerial robots as force effectors. Haptic (bilateral) feedback from the master side enables the user to feel the contact forces exerted by the robot. An automatic potential field-based slowing-down policy is used by the robot to ensure a smooth transition between the contact-free motion phase and the force interaction phase. The effectiveness of the approach has been shown in extensive human-in-the-loop simulations including remote pressing of buttons on a surface and pushing a cart until it touches a wall.

I. INTRODUCTION

Most of the applications designed for aerial robots, also known as Unmanned Aerial vehicles (UAVs), concern inspection and surveillance of places inaccessible by humans or grounded vehicles. The possibility of letting aerial robot interact with the environment opens an additional wide set of potential applications for aerial robots like, e.g., maintenance, construction, cooperative grasping and transportation.

The problem of modeling and control of an aerial robot interacting with objects and surfaces in the surroundings has been faced in some very recent projects [1], [2]. In [3] a delta-shaped structure has been mounted on a quadrotor to apply forces on the environment using a passivity-based controller. One-dimensional manipulation of a cart performed by one or two quadrotors, equipped with rigid and passive tools, has been studied in [4]. A hybrid force/motion control for a quadrotor with a rigid tool attached on it applying forces on external environment has been presented in [5]. A standard near-hovering controller has been used instead in [6] to let a quadrotor UAV apply a desired 3D force on a surface by means of a rigidly attached passive tool.

Most of the UAVs' real applications, requiring or not the physical interaction of the robots with the surrounding objects, take place in unstructured, uncertain and unknown environments. This could make the fully-autonomous control of the UAVs practically unreliable. The human intervention thus becomes advisable to handle uncertainty issues.

Moreover, providing the human operators with an intuitive and natural interface to control the vehicle would open

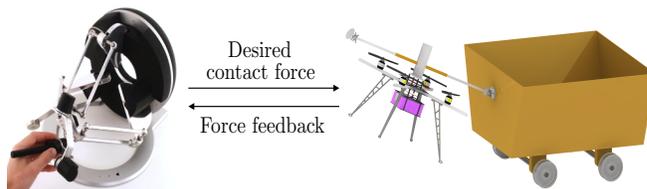


Fig. 1: Schematic representation of the proposed teleoperation framework.

the possibility of tele-operating UAVs also to users with no particular piloting skills. Several possible bilateral teleoperation schemes using haptic feedback to steer single or multiple UAVs have been proposed in the last decade. In [7] an impedance controller is presented that allows quadrotor teleoperation with haptic feedback to the pilot that was able to steer the vehicle, avoiding collisions with obstacles in the environment. In [8] an admittance control mode, dual to the impedance mode, has been developed. In [9], impedance control was used for velocity control while a force feedback was computed using the distance from obstacles.

In [10] a multi-layer architecture was developed to steer a formation of UAVs while perceiving the state of all (or some) of the robots along with the presence of obstacles in the environment. A decentralized passivity-based approach was used in [11] in order to cope with the presence of possible destabilizing time-varying topologies. A reactive planning approach to bilaterally teleoperate a UAV is presented in [12] where the human modifies the future path and receives a integral haptic feedback related to the path flyability.

To the best of our knowledge the problem of teleoperating UAVs able to establish contact and apply forces on objects in the surroundings has not been yet deeply investigated. Our main contribution is to propose a novel teleoperation framework based on the controller presented in [6], suitable for both contact-free tasks and arbitrary 3D force exertion on the environment using a rigidly attached tool. The user is provided with a force feedback proportional to the force applied by the vehicle at the contact point (Fig. 1). An automatic slowing-down policy is introduced to ensure a smooth transition between free motion and physical interaction.

The rest of this paper is organized as follows. In Sec. II the models of the master and the slave systems involved in the teleoperation are described. The adopted near-hovering controller is presented in Sec. III along with the main results obtained in [6]. In Sec. IV the teleoperation architecture is described. The proposed framework has been validated in simulations whose results are described in Sec. V, while in Sec. VI conclusions and possible future developments of the

¹Department of Information Engineering and Mathematics, University of Siena, via Roma 56, 53100 Siena, Italy. [gioioso, mohammadi, prattichizzo]@dii.unisi.it

²Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego 30, 16163 Genova, Italy.

³CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France

⁴Univ de Toulouse, LAAS, F-31400 Toulouse, France afranchi@laas.fr

work are outlined.

II. SYSTEM MODEL

The proposed teleoperation scheme consists of a fully actuated haptic joystick (master), an underactuated aerial robot (slave), and a communication link between them. In the present paper communication time-delays will not be taken into account. The models considered for master and slave systems are described in the following.

A. Master Model

The dynamics of a gravity/inertial compensated haptic interface can be simply described by:

$$\mathbf{M}_m \ddot{\mathbf{q}} = \mathbf{f}_h + \mathbf{f}_c \quad (1)$$

where $\mathbf{q} \in \mathbb{R}^3$ is the position of the haptic device end effector, $\mathbf{M}_m \in \mathbb{R}^{3 \times 3}$ is a diagonal matrix, $\mathbf{f}_h \in \mathbb{R}^3$ is the human force on the haptic probe, and $\mathbf{f}_c \in \mathbb{R}^3$ is the control force.

B. Slave Model

In this section we present the standard dynamical model of an underactuated aerial robot, of which the quadrotor is the most popular example. Let us consider a reference world frame $\mathcal{W} : \{\mathcal{O}_{\mathcal{W}}, \vec{w}_1, \vec{w}_2, \vec{w}_3\}$ and a robot body frame $\mathcal{B} : \{\mathcal{O}_{\mathcal{B}}, \vec{b}_1, \vec{b}_2, \vec{b}_3\}$. The origin of this second frame is placed at the center of mass of the UAV and coincides with the geometric center of the four propellers. The position of $\mathcal{O}_{\mathcal{B}}$ in \mathcal{W} and the corresponding linear velocity will be indicated with $\mathbf{p} = (p_1, p_2, p_3)^T \in \mathbb{R}^3$ and with $\mathbf{v} = (v_1, v_2, v_3)^T \in \mathbb{R}^3$ respectively. Let us denote by $m \in \mathbb{R}$ the mass of the vehicle and by $\mathbf{J} \in \mathbb{R}^{3 \times 3}$ its inertia matrix with respect to the body frame. The matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ represents the rotation matrix from the body frame to the inertial world frame and $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity vector representing the rotation of \mathcal{B} with respect to \mathcal{W} , expressed in \mathcal{B} . The orientation of the vehicle is denoted by $\boldsymbol{\eta} := (\phi, \theta, \psi)^T \in \mathbb{R}^3$, being ϕ, θ, ψ respectively the roll, pitch and yaw angles (RPY). The matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ can be thus written as $\mathbf{R}(\boldsymbol{\eta}) = \mathbf{R}_z(\psi) \mathbf{R}_y(\theta) \mathbf{R}_x(\phi)$, where $\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z$ are the elementary rotation matrices about the three coordinate axis. The total thrust acting on the $-\vec{b}_3$ direction, denoted by $\lambda \in \mathbb{R}^+$, and the total moment expressed in \mathcal{B} , denoted by $\boldsymbol{\tau} = (\tau_1, \tau_2, \tau_3)^T$, are assumed to be the control inputs of this system.

The aerial robot is equipped with a tool used to establish contact with the environment. Its tip position in \mathcal{B} is expressed by: $\mathbf{d} = (d_x, d_y, d_z)^T \in \mathbb{R}^3$. The position $\mathbf{y} \in \mathbb{R}^3$ of the tool-tip in the world frame is:

$$\mathbf{y} = \mathbf{p} + \mathbf{R}\mathbf{d}. \quad (2)$$

The force exerted by the environment on the tool-tip, expressed in the world frame, will be denoted by $\mathbf{f}^e \in \mathbb{R}^3$. According to the hard finger contact model [13], we assume a negligible interaction moment at the contact point.

Given these assumptions, the equations of motion of the slave robot can be written as:

$$\dot{\mathbf{p}} = \mathbf{v} \quad (3)$$

$$m\dot{\mathbf{v}} = -\lambda \mathbf{R}\hat{\mathbf{z}} + mg\hat{\mathbf{z}} + \mathbf{f}^e \quad (4)$$

$$\dot{\boldsymbol{\eta}} = \mathbf{T}(\boldsymbol{\eta})\boldsymbol{\omega} \quad (5)$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\tau} + \mathbf{d} \times \mathbf{R}^T \mathbf{f}^e, \quad (6)$$

where $\hat{\mathbf{z}} = (0, 0, 1)^T$ and $\mathbf{T}(\boldsymbol{\eta}) \in \mathbb{R}^{3 \times 3}$ is a matrix that transforms $\boldsymbol{\omega}$ to the RPY rates $\dot{\boldsymbol{\eta}}$. Another common assumption will be adopted in the rest of the paper: the aerial robot is never in a singular configuration (i.e., $-\frac{\pi}{2} < \phi < \frac{\pi}{2}$ and $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$). In the singular configuration, in fact, the total thrust cannot be used to balance the gravity force and this represents a useless case for the purposes of this work.

III. NEAR-HOVERING CONTROLLER ADAPTED FOR PHYSICAL INTERACTION

The aerial robot in exam is an underactuated system with its six configuration parameters, $(\mathbf{p}, \boldsymbol{\eta})$ and four control inputs $(\lambda, \boldsymbol{\tau})$. Despite its underactuation it is able to apply arbitrary 3D forces on the environment, as shown in [6]. In fact, in [6] a mathematical derivation has been presented which allows to adopt a widely used near-hovering controller not only during contact-free flight but also to apply desired contact forces on the surrounding environment. In this section the structure of the mentioned controller, will be recalled along with the main results obtained in [6], which represent the core of the proposed framework.

A fast inner-loop PID controller is aimed at letting the robot track a desired attitude $\boldsymbol{\eta}_d = (\phi_d, \theta_d, \psi_d)^T \in \mathbb{R}^3$, relying on the fact that the rotational dynamics is fully actuated. Differentiating (5) and substituting (6), we get

$$\dot{\boldsymbol{\eta}} = \mathbf{T}(\boldsymbol{\eta})\mathbf{J}^{-1}(-\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\tau} + \mathbf{d} \times \mathbf{R}^T \mathbf{f}^e) + \dot{\mathbf{T}}(\boldsymbol{\eta})\boldsymbol{\omega}.$$

We can thus define the attitude controller as:

$$\boldsymbol{\tau} = \mathbf{J}\mathbf{T}^{-1}(\boldsymbol{\eta})\mathbf{K}_A \boldsymbol{\eta}_e \quad (7)$$

where:

$$\mathbf{K}_A^T = \begin{bmatrix} 0 & 0 & k_{DA}^\phi \\ 0 & 0 & k_{PA}^\phi \\ 0 & 0 & k_{IA}^\phi \\ 0 & k_{DA}^\theta & 0 \\ 0 & k_{PA}^\theta & 0 \\ 0 & k_{IA}^\theta & 0 \\ k_{DA}^\psi & 0 & 0 \\ k_{PA}^\psi & 0 & 0 \\ k_{IA}^\psi & 0 & 0 \end{bmatrix}, \quad \boldsymbol{\eta}_e = \begin{bmatrix} -\dot{\phi} \\ \phi_d - \phi \\ \int (\phi_d - \phi) dt \\ -\dot{\theta} \\ \theta_d - \theta \\ \int (\theta_d - \theta) dt \\ -\dot{\psi} \\ \sin(\psi_d - \psi) \\ \int \sin(\psi_d - \psi) dt \end{bmatrix} \quad (8)$$

leading to the closed-loop dynamics:

$$\dot{\boldsymbol{\eta}} + \mathbf{K}_A \boldsymbol{\eta}_e = \mathbf{T}\mathbf{J}^{-1}(-\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{d} \times \mathbf{R}^T \mathbf{f}^e) + \dot{\mathbf{T}}\boldsymbol{\omega}.$$

Local exponential stability has been shown for $(\boldsymbol{\eta}_e, \dot{\boldsymbol{\eta}}_e)$ under suitable conditions of the initial state and suitable gains.

The expression (7) can be simplified considering that in the near-hovering conditions assumed for the contact-free flight $\mathbf{T}^{-1}(\boldsymbol{\eta})$ can be approximated by a 3×3 identity matrix and \mathbf{J} can be assumed to be diagonal, as common practice, and hidden in the gain matrix \mathbf{K}_A . The expression of $\boldsymbol{\tau}$ can be thus written as:

$$\boldsymbol{\tau} = \mathbf{K}_A \boldsymbol{\eta}_e. \quad (9)$$

These approximations have been proven to work very well in reality under near-hovering conditions [11], [10].

An outer loop controller is then designed to track a desired vertical dynamics as:

$$\lambda = -\frac{m}{\cos \phi \cos \theta} (-g - b_p^z v_3 + k_{pp}^z e_3) \quad (10)$$

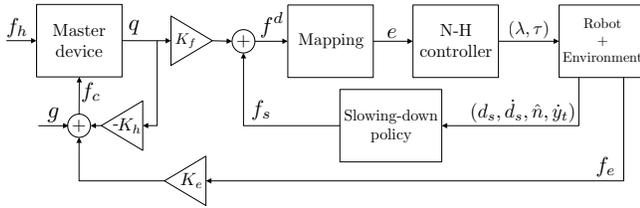


Fig. 2: Block diagram of the proposed teleoperation framework.

where b_p^z and k_{pp}^z are positive scalar gains and e_3 is an additional control input. In order to act on the horizontal dynamics, the desired roll ϕ_d and pitch θ_d are set such that

$$\begin{pmatrix} \sin \phi_d \\ \sin \theta_d \end{pmatrix} = \frac{m}{\lambda} \begin{pmatrix} -s_\psi & c_\psi \\ -c_\psi/c_\phi & -s_\psi/c_\phi \end{pmatrix} \begin{pmatrix} -b_{pv1} + k_{pp}e_1 \\ -b_{pv2} + k_{pp}e_2 \end{pmatrix} \quad (11)$$

where e_1, e_2 are two additional control inputs, and s_\bullet and c_\bullet denote $\sin(\bullet)$ and $\cos(\bullet)$ respectively. The reader is referred to [10] for a detailed analytical derivation of (11). Let us gather the three additional control inputs in the vector $\mathbf{e} = (e_1, e_2, e_3)^T \in \mathbb{R}^3$.

The presented controller has been used in literature to steer aerial robots in contact-free motion by choosing $\mathbf{e} = \mathbf{p}_d - \mathbf{p}$ in order to regulate \mathbf{p} to a desired position $\mathbf{p}_d \in \mathbb{R}^3$ (see [10] for further details).

What has been shown in [6] is that, during the contact phase, a mapping exists between the control input $\mathbf{e} = (e_1, e_2, e_3)^T \in \mathbb{R}^3$ and the contact force \mathbf{f}^e at steady state. The analytical expression of this mapping depends on the presence of integral terms in the attitude controller. If non-zero integral gains are considered for roll and pitch regulation in the inner loop (i.e., $k_{IA}^\phi > 0, k_{IA}^\theta > 0, k_{IA}^\psi$ any), the mapping assumes the simplest form and can be written as:

$$e_1 = -\frac{f_1^e}{mk_{pp}} \quad e_2 = -\frac{f_2^e}{mk_{pp}} \quad e_3 = -\frac{f_3^e}{k_{pp}}. \quad (12)$$

For the more complex expressions of the mapping obtained in the other cases (when k_{IA}^ϕ and k_{IA}^θ are equal to zero) the reader is referred to [6].

The mapping between \mathbf{e} and \mathbf{f}^e has been used in [6] to let an autonomous UAV apply a desired steady-state contact force $\mathbf{f}^d \in \mathbb{R}^3$ on the environment (equal to $-\mathbf{f}^e$). Given the desired force \mathbf{f}^d , the error \mathbf{e} was then computed (still considering the simple case of non-zero integral terms for the inner-loop) as:

$$e_1 = \frac{f_1^d}{mk_{pp}} \quad e_2 = \frac{f_2^d}{mk_{pp}} \quad e_3 = \frac{f_3^d}{k_{pp}} \quad (13)$$

and given to the outer-loop controller to result in the desired contact force.

Remark 1: It is worth noting that the presented framework allows an underactuated aerial robot to apply an arbitrary 3D force on the environment, relying on its four actuators, as shown in [6]. The underactuation of the robot appears in the fact that the position and the orientation cannot be freely chosen when applying the arbitrary 3D force

The aim of the present paper is to exploit this result to build a teleoperation framework in which a human operator can manually regulate \mathbf{f}^d by changing the position of the haptic probe and is provided with a force feedback computed

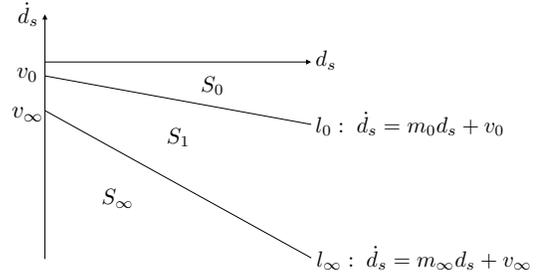


Fig. 3: Graphical representation of the adopted slowing-down policy.

on the basis of the contact force that arises from the robot-environment interaction.

IV. TELEOPERATION FRAMEWORK

The proposed teleoperation framework is schematically represented by the block diagram in Fig. 2 and described in the following. In this work we assume that the yaw is automatically regulated by the robot and set such that the projection of the tool on the x-y plane of \mathcal{W} is normal to the target surface and the tool is pointing to it.

A. User Command

The human operator applies a force \mathbf{f}_h on the haptic device changing its position \mathbf{q} . The desired force \mathbf{f}^d is computed as:

$$\mathbf{f}^d = \mathbf{K}_f \mathbf{q} + \mathbf{f}_s \quad (14)$$

where $\mathbf{K}_f \in \mathbb{R}^{3 \times 3}$ is a diagonal matrix with positive entries, chosen to produce a desired mapping between the workspace of the haptic device and the range of possible forces applicable by the robot on the environment.

The term $\mathbf{f}_s \in \mathbb{R}^3$ in (14) constitutes a potential force whose role is to automatically slow down the vehicle when it is approaching the part of the environment on which the user wants to apply a force. At the beginning of the interaction phase, in fact, the speed of the vehicle has to be sufficiently small, in order to avoid undesirable high impact forces that might damage both the robot and the environment. The vector \mathbf{f}_s is thus designed as:

$$\mathbf{f}_s = b_n(d_s, \dot{d}_s) \hat{\mathbf{n}} - b_t(d_s) \dot{\mathbf{y}}_t, \quad (15)$$

where $\hat{\mathbf{n}} \in \mathbb{R}^3$ is the unitary vector normal to the target surface, $\dot{\mathbf{y}}_t \in \mathbb{R}^3$ is the velocity of the tooltip projected on the surface plane and $b_n(d_s, \dot{d}_s) \in \mathbb{R}$ and $b_t(d_s) \in \mathbb{R}$ are positive scalar functions defined in the following.

The scalar $b_n(d_s, \dot{d}_s)$ is a function of the distance $d_s \in \mathbb{R}$ between the tooltip and the target surface and its derivative $\dot{d}_s \in \mathbb{R}$. Let l_0 and l_∞ be two straight lines on the half plane represented in Fig. 3 which are described by the equations:

$$l_0: \dot{d}_s = m_0 d_s + v_0 \quad l_\infty: \dot{d}_s = m_\infty d_s + v_\infty, \quad (16)$$

where $m_\infty < m_0 < 0$ and $v_\infty < v_0 < 0$ are negative scalar parameters, properly chosen. Denote with $\tilde{p} = (\tilde{d}_s, \tilde{\dot{d}}_s)$ the actual state of the tooltip on that half plane, and define the two points \tilde{p}_0 and \tilde{p}_∞ as the intersections of the vertical lines passing through \tilde{p} with l_0 and l_∞ , respectively.

Now consider a function $\alpha(d_s, \dot{d}_s) \in \mathbb{R}$ defined as:

$$\alpha(d_s, \dot{d}_s) = \frac{\|\tilde{p} - \tilde{p}_\infty\|}{\|\tilde{p}_0 - \tilde{p}_\infty\|}, \quad (17)$$

which represents the vertical distance of the point \tilde{p} from the line l_∞ , normalized in such a way that it is equal to one when \tilde{p} lies on l_0 . The function b_n is finally defined as follows:

$$b_n(\alpha) = \begin{cases} \frac{1}{\cos^2(\frac{\pi}{2}\alpha - \frac{\pi}{2})} - \cos(\frac{\pi}{2}\alpha - \frac{\pi}{2}) & \text{if } 0 < \alpha \leq 1 \\ 0 & \text{otherwise} \end{cases}. \quad (18)$$

Function b_n is nonzero only in the zone denoted by S_1 in Fig. 3. In the zones S_0 and S_∞ it is equal to zero and thus in those regions \mathbf{f}_s does not act in the direction $\hat{\mathbf{n}}$ normal to the target surface.

The function $b_t(d_s)$ serves as a damping factor that acts on the projection of the tooltip velocity on the plane tangential to the target surface. Its value is defined as

$$b_t(d_s) = \begin{cases} b_{max}(1 - \frac{d_s}{d^*}) & \text{if } 0 \leq d_s \leq d^* \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

where b_{max} is the maximum damping value when $d_s = 0$ (i.e., when the contact phase starts) and d^* is the value of the distance d_s at which the damping action starts.

Remark 2: Note that the term \mathbf{f}_s defined by (15), (17) (18) and (19) only acts on \mathbf{f}^d during the approaching phase while it is zero during the contact. This allows to apply the user-commanded force on the surface without interference. Furthermore \mathbf{f}^s is zero during the detaching phase (i.e., when $\dot{d}_s > 0$) and the possible traveling phases used to move between several contact points.

The mapping recalled in Section III is then used to transform \mathbf{f}^d in the control input \mathbf{e} which is the input for the near-hovering controller that computes the corresponding control inputs (λ, τ) for the UAV. If the robot is in contact with the environment, the controller presented in Section III, fed with the input \mathbf{e} , will produce, at steady state, a contact force \mathbf{f}^d at the tooltip. Conversely, if no contact occurs, the same control input \mathbf{e} will produce a motion of the vehicle in the direction of \mathbf{f}^d . The user can thus intuitively control the motion of the robot in the environment and also its interaction with it.

B. Force Feedback

The control force for the master device \mathbf{f}_c is computed as:

$$\mathbf{f}_c = -\mathbf{K}_h \text{sat}(\mathbf{q}, q_{max}) - \mathbf{B}_h \dot{\mathbf{q}} + \mathbf{K}_e \mathbf{f}^e \quad (20)$$

where $\mathbf{K}_h \in \mathbb{R}^{3 \times 3}$, $\mathbf{B}_h \in \mathbb{R}^{3 \times 3}$ and $\mathbf{K}_e \in \mathbb{R}^{3 \times 3}$ are diagonal matrices with positive entries, $q_{max} \in \mathbb{R}$ is a positive scalar, and the function $\text{sat}(x, y) \in \mathbb{R}^3 \times \mathbb{R}^3$ is defined as:

$$\text{sat}(\mathbf{q}, q_{max}) = \begin{cases} \mathbf{q} & \text{if } |\mathbf{q}| \leq q_{max} \\ \frac{\mathbf{q}}{|\mathbf{q}|} q_{max} & \text{otherwise} \end{cases}. \quad (21)$$

The role of the first term in the rhs of (20) is to ensure the haptic probe stabilization at $(\mathbf{q} = [0, 0, 0])$ in absence of external forces. This term also gives to the user the feeling of the distance from zero-commanded force. The second term is a damping factor that prevents the system from oscillatory behaviors. The third term is used to compensate



Fig. 4: A picture of the experimental setup. The human operator teleoperates the UAV using the haptic device and relies on a fixed camera (left) and an onboard camera (right) to see the virtual environment.

the gravitational force. Finally $\mathbf{K}_e \mathbf{f}^e$ represents the force feedback provided to the user that gives the perception of the contact force applied by the robot on the environment.

In order to implement this kind of feedback we assume to have a measure of the external force \mathbf{f}^e that occurs at the contact point during the interaction. This measure can be obtained by means of a force sensor placed at the tooltip or implementing an observer for the external force. A possible choice for the observer is represented by the algorithm presented in [14]. Note that the matrices of gains \mathbf{K}_h , \mathbf{B}_h , the scalars q_{max} and \dot{q}_{max} and \mathbf{K}_e in (20) should be designed s.t. the stabilization terms $-\mathbf{K}_h \text{sat}(\mathbf{q}, q_{max})$ and $\mathbf{B}_h \dot{\mathbf{q}}$ become negligible w.r.t. $\mathbf{K}_e \mathbf{f}^e$ during the interaction phase (i.e. when $\mathbf{f}^e \neq 0$), according to the performed task.

In this paper we do not provide a formal proof of the stability of the teleoperation system, which is left as a future development of this work. However we want to briefly discuss some stability-related facts about the system in exam. The stability of the slave system (3)-(6) under the control law (9)-(11) has been proven in [10]. Moreover in [6] the stability of the equilibrium points of the system in contact with the environment has been proven by means of numerical analysis using the indirect Lyapunov method. The structure of the teleoperation framework proposed in this work has some affinity with the scheme presented in [9], even though there physical interaction was not allowed. In that paper the stability of the teleoperation system, subject to bounded forces applied by the human operator on the master and bounded external forces acting on the slave, was proven. In that case the external forces acting on the slave were preventing the vehicle from colliding with obstacles in the environment. Nevertheless, the proof shown in [9] gives us grounded hopes on the existence of a formal proof for our teleoperation framework in which the external forces acting on the slave arise from its interaction with the environment. Furthermore a stable behavior of the proposed force-based teleoperation scheme was practically observed during the human-in-the-loop simulations described in the following section.

V. HUMAN-IN-THE-LOOP SIMULATIONS

To evaluate the efficiency of the proposed bilateral teleoperation scheme, a set of human-in-the loop simulations has been performed, in which the human operator teleoperates a quadrotor aerial robot in a virtual environment. The experimental setup is shown in Fig. 4. The operator

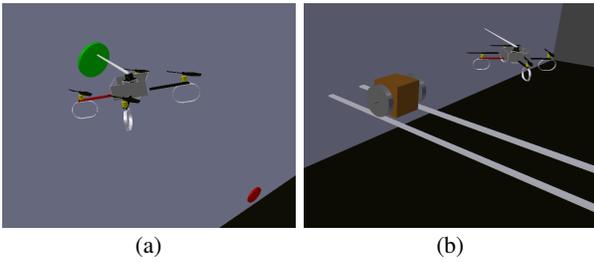


Fig. 5: Screenshots of the first (a) and second (b) scenarios designed for the human-in-the-loop simulations.

sends the commands using an Omega.3 (Force Dimension) haptic device to a simulated quadrotor. The dynamic model of the quadrotor and the proposed controller are simulated using Matlab and Simulink. The interaction between the robot and the environment is also computed in Simulink. In particular the presence of a force sensor at the end of the tool is emulated by computing the interaction forces between the tooltip and the environment using the god-object model. The visualization of robot and environment has been implemented using the open source game engine Blender. Control loop and data communication runs with a rate of 1000 Hz. Position and orientation of the simulated vehicle and other objects are computed in Simulink at 1000 Hz and sent to Blender with a rate of 25 Hz. Two cameras (one fixed and one onboard) give the visual feedback to the operator. The inertial parameters of the simulated quadrotor are: $m = 1\text{kg}$, $J = \text{diag}(0.13, 0.13, 0.22)\text{kgm}^2$. Two different scenarios have been designed to validate the approach in which the operator was asked to perform different tasks. A detailed description of the two tasks is given in Secs.V-A and V-B. The results obtained during the simulations performed in the first and in the second scenario are summarized in the plots shown in Fig. 7 and 8, respectively. In the plots the following correspondence between colors and cartesian axis has been established: blue $\rightarrow x$, green $\rightarrow y$, red $\rightarrow z$. For the plots regarding angles the adopted convention is: blue $\rightarrow \text{roll}$, green $\rightarrow \text{pitch}$, red $\rightarrow \text{yaw}$. In Figures 7a and 8a the dashed lines show the force commanded by the user $\mathbf{K}_f \mathbf{q}$, while the continuous lines represent the desired force \mathbf{f}^d . Figures 7b and 8b show the force commanded by the user (dashed lines) and the force exerted by the robot on the environment \mathbf{f}^e (continuous lines). In Figures 7c and 8c the continuous lines represent \mathbf{f}^e while the dashed lines are used to plot the control force \mathbf{f}_c . The error \mathbf{e} is shown in Figures 7d and 8d. Figures 7e and 8e show the desired (dashed lines) and actual (continuous lines) attitude of the vehicle. In Figures 7f and 8f the control inputs λ (black line) and τ (colored lines) can be observed. The plots have been divided by vertical dashed black lines into *phases* denoted by P1, P2 and P3. The phases denoted by P1 represent free motion phases in which the slowing-down policy does not act on \mathbf{f}^d (i.e., $\mathbf{f}_s = 0$). The ones denoted by P2 represent the phases of the task in which the slowing-down policy acts on \mathbf{f}^d (i.e. $\mathbf{f}_s \neq 0$). Finally, the contact phases are denoted by P3. The three different phases can be also interpreted, considering Fig. 3, as it follows. The phases denoted by P1 can be interpreted as the ones in which $(d_s, \dot{d}_s) \in S_0$. The phases denoted by P2 as the ones in

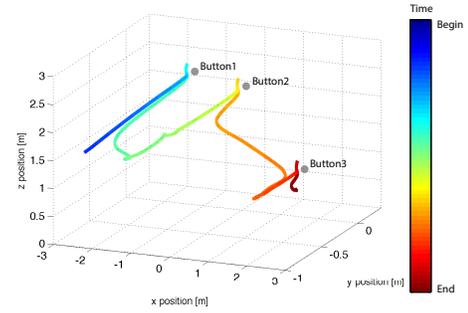


Fig. 6: 3D position of the robot during the simulation performed in the first scenario.

which $(d_s, \dot{d}_s) \in S_1$ and $\mathbf{f}_s \neq 0$. During these phases the robot approaches the target surface leading to small values of d_s and \dot{d}_s until the contact is reached and we enter in the phases denoted by P3, in which $(d_s, \dot{d}_s) = (0, 0)$. The current phase can be thus univocally determined, based on the value of d_s and \dot{d}_s . The reader is encouraged to watch the video attached to this paper in which a full human-in-the-loop simulation is shown.

A. First Scenario - Pressing buttons on a wall

In the first scenario a human operator was asked to drive the quadrotor to push three buttons on a wall. The positions of the buttons were randomly chosen. In order to fulfill the task, the UAV had to apply a force on the three buttons greater than a certain threshold (5N). This experiment has been designed to show the applicability of the proposed framework for both contact-free flight and interaction phases. Two screenshots of the simulation are given in Fig. 5a.

The results of this experiment are plotted in Fig. 7. Figure 6 shows the 3D trajectory of the UAV during the test. The first phase is a free motion phase in which the user controls the motion of the robot and aligns the tool position with the first button to be pressed on the wall. The approaching phase then starts, denoted by P2. The action of \mathbf{f}_s on \mathbf{f}^d is shown in Fig. 7a: the difference between \mathbf{f}_d and the force commanded by the user increases while the robot gets closer to the button. The third phase, denoted by P3, is the first contact phase in which the robot presses the first button on the wall. In Fig. 7b the contact force \mathbf{f}^e is indeed different from zero in this phase, and it tracks the desired force \mathbf{f}^d . At the beginning of this phase, a positive peak of force on the z axis can be observed. This transient force is due to the underactuation of the considered robotic platform. In order to apply the pushing force on the button, in fact, the UAV has to tilt with a negative pitch (shown in Fig. 7e). This results in an undesired force read by the sensor during the transient phase which is not considered by the mapping recalled in Section III which only takes into account steady-state forces. Let us consider the Fig. 7c for what concerns the force feedback. It can be seen how \mathbf{f}_c follows the external force \mathbf{f}^e despite of an offset due to the presence of the stabilization term $-\mathbf{K}_h \mathbf{q}$ in (20). In these simulations the value of the stabilization term cannot be considered negligible w.r.t. to the contact force. Nevertheless it was found to be appropriate for the considered conditions

and led to a comfortable teleoperation system. The value of the force feedback is considerably higher in the contact phase denoted by P3 w.r.t. to the one recorded during the previous phases. During the rest of the experiment the same sequence of phases is replicated twice. Reasonable values of thrust and torques have been computed by the controller during the simulation (see Fig. 7f), in correspondence of the (bounded) values of the input e shown in Fig. 7c.

B. Second Scenario - Pushing a cart toward a wall

The second virtual environment consists of a cart machine moving on rails (see Fig. 5b). The user was asked to push the cart machine toward a wall placed at the end of the rails and stop the movement when the contact with it was reached. The inertia and the friction of the cart on the rails prevent it to move freely. The human operator applies a certain amount of force on it, by teleoperating the UAV, to reach the wall. The aim of the experiment is to show the ability of the system to apply forces on moving objects and to show the effectiveness of the proposed force feedback strategy in rendering forces of the environment to the human operator.

Let us consider the Fig. 8. In the first phase (denoted by P1) of the simulation, the UAV is teleoperated by the user in order to approach the cart that has to be pushed on the rails. The distance between the tool and the target surface of the cart starts decreasing until $(d_s, \dot{d}_s) \in S_1$ and the second phase (denoted by P2) starts. In this second phase, due to the action of the slowing-down control action, the distance between the dashed and the continuous blue lines of Fig. 8a starts to be greater than zero and it increases until the contact is reached. When the contact phase (denoted by P3) starts, a peak of force on the z axis is read by the sensor and shown in Fig. 8b and 8c. This peak can be explained with the same arguments used in Section V-A to explain the peaks recorded during the simulation performed in the first scenario. After the transient phase, also in this test, the force \mathbf{f}^e tracks the desired force \mathbf{f}^d (see Fig. 8b). At the end of the contact phase a peak of force in the x direction is observable in Fig. 8b. This important haptic cue happens when the cart touches the wall. This peak is also felt by the user (as shown in 8c). During the experiment the UAV has a stable behavior (as shown in Fig. 8e) and the control inputs of the near-hovering controller, in correspondence to the error e shown in Fig. 8d, reach reasonable values (see Fig. 8f) w.r.t. a typical quadrotor platform of that size and weight. The positions of both the UAV and the cart in the three different phases of the simulation can be observed in Fig. 9. It can be seen from this plot that the cart moves under the action of the UAV and subject to the friction force until it reaches the wall.

VI. CONCLUSIONS AND FUTURE WORK

In this paper a novel teleoperation framework for aerial robots able to establish contact with the surrounding environment has been presented. The framework is based on the mathematical derivation previously presented in [6], in which the problem of an autonomous aerial robot applying an arbitrary 3D force on the environment was addressed.

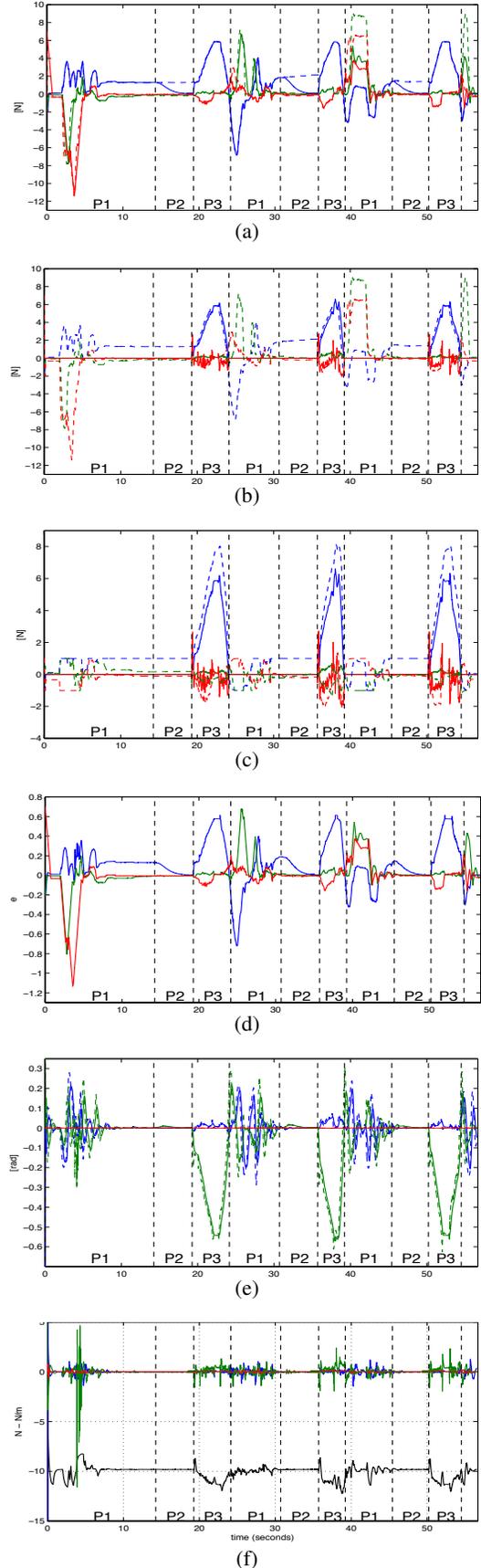


Fig. 7: Results of the simulation performed in the first scenario.

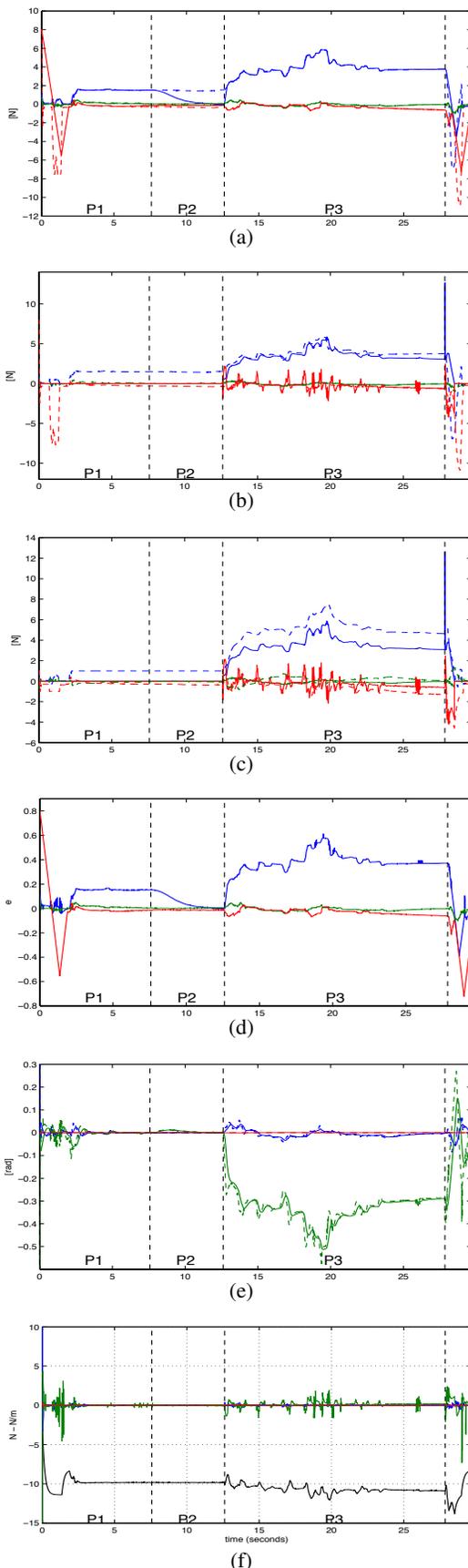


Fig. 8: Results of the simulation performed in the second scenario.

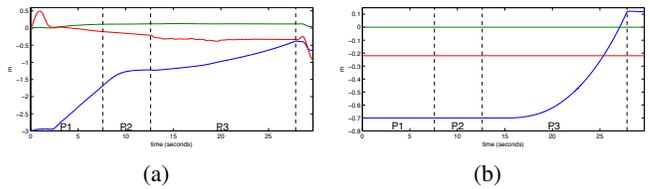


Fig. 9: Position of the UAV (a) and position of the cart (b) recorded during the simulation performed in the second scenario.

The proposed teleoperation architecture allows a human operator to intuitively drive the vehicle during the free motion and also to apply a desired force on a surface of the environment by means of a passive tool rigidly attached on the robot. The approach has been validated in extensive human-in-the-loop simulations aimed at showing its effectiveness. The validation of the framework in a real experimental setup is the future development of this work. Work is ongoing to formally prove the stability of the proposed teleoperation loop.

REFERENCES

- [1] AIRobots, "EU Collab. Project ICT-248669." www.airobots.eu.
- [2] ARCAS, "EU Collab. Project ICT-287617." www.arcas-project.eu.
- [3] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi, "Modeling and control of a flying robot for contact inspection," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, (Vilamoura, Portugal), pp. 3532–3537, Oct 2012.
- [4] M. B. Srikanth, A. Soto, A. Annaswamy, E. Lavretsky, and J.-J. Slotine, "Controlled manipulation with multiple quadrotors," in *AIAA Conf. on Guidance, Navigation and Control*, (Portland, OR), Aug. 2011.
- [5] H. Nguyen and D. Lee, "Hybrid force/motion control and internal dynamics of quadrotors for tool operation," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, (Tokyo, Japan), pp. 3458–3464, November 2013.
- [6] G. Gioioso, M. Ryll, D. Prattichizzo, H. H. Bühlhoff, and A. Franchi, "Turning a near-hovering controlled quadrotor into a 3D force effector," in *2014 IEEE Int. Conf. on Robotics and Automation*, (Hong Kong, China), pp. 6278–6284, May 2014.
- [7] R. Mahony, F. Schill, P. Corke, and Y. S. Oh, "A new framework for force feedback teleoperation of robotic vehicles based on optical flow," in *2009 IEEE Int. Conf. on Robotics and Automation*, (Kobe, Japan), pp. 1079–1085, May 2009.
- [8] F. Schill, X. Hou, and R. Mahony, "Admittance mode framework for haptic teleoperation of hovering vehicles with unlimited workspace," in *2010 Australasian Conf. on Robotics & Automation*, (Brisbane, Australia), December 2010.
- [9] H. Rifa, M. D. Hua, T. Hamel, and P. Morin, "Haptic-based bilateral teleoperation of underactuated unmanned aerial vehicles," in *18th IFAC World Congress*, (Milano, Italy), pp. 13782–13788, Aug. 2011.
- [10] D. J. Lee, A. Franchi, H. I. Son, H. H. Bühlhoff, and P. Robuffo Giordano, "Semi-autonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles," *IEEE/ASME Trans. on Mechatronics, Focused Section on Aerospace Mechatronics*, vol. 18, no. 4, pp. 1334–1345, 2013.
- [11] A. Franchi, C. Secchi, H. I. Son, H. H. Bühlhoff, and P. Robuffo Giordano, "Bilateral teleoperation of groups of mobile robots with time-varying topology," *IEEE Trans. on Robotics*, vol. 28, no. 5, pp. 1019–1033, 2012.
- [12] C. Masone, P. Robuffo Giordano, H. H. Bühlhoff, and A. Franchi, "Semi-autonomous trajectory generation for mobile robots with integral haptic shared control," in *2014 IEEE Int. Conf. on Robotics and Automation*, (Hong Kong, China), pp. 6468–6475, May 2014.
- [13] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.), pp. 671–700, Springer, 2008.
- [14] B. Yüksel, C. Secchi, H. H. Bühlhoff, and A. Franchi, "A nonlinear force observer for quadrotors and application to physical interactive tasks," in *2014 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, (Besançon, France), pp. 433–440, Jul. 2014.