

Bilateral Haptic Teleoperation of an Industrial Multirotor UAV

Sammy Omari¹, Minh-Duc Hua², Guillaume Ducard³, and Tarek Hamel³

¹ Autonomous Systems Lab (ASL), ETH Zurich, Switzerland &
Skybotix AG, Luegislandstrasse 105, Zurich, Switzerland
`omaris@ethz.ch`

² Institut des Systèmes Intelligents et de Robotique (ISIR), CNRS-UPMC, France
`hua@isir.upmc.fr`

³ Laboratoire d'Informatique, Signaux et Systèmes (I3S), CNRS-UNS, France
`ducard(thamel)@i3s.unice.fr`

Abstract. This chapter presents an intuitive laser-based teleoperation scheme to enable the safe operation of a multirotor UAV by an untrained user in a cluttered environment using a haptic joystick. An obstacle avoidance strategy is designed and implemented to autonomously modify the position setpoint of the UAV if necessary. This scheme includes a novel force-feedback algorithm to enable the user to feel surrounding environment of the UAV as well as the disturbances acting on it. The stability analysis of the whole teleoperation loop, including the nonlinear dynamics of both UAV and joystick, is provided. The implementation of the teleoperation scheme on the Flybox hexacopter platform by the company Skybotix is described. Finally, experimental results and videos are reported to demonstrate the successful implementation and the performance of the overall system.

Keywords: Aerial Robotics, Teleoperation, Haptics and Haptic Interfaces.

1 Introduction

The next generation of unmanned aerial vehicles (UAVs) will be capable of executing missions that are too dangerous, too difficult or simply impossible for humans. This is the case, for example, when it is necessary to inspect power lines, buildings, bridges, tunnels, hydraulic dam walls, pipelines, etc. Also, after a natural or industrial disaster, the inspection of the inside of a collapsed house, building or plant, by a small UAV would represent a considerable advantage in terms of time-saving, cost and human risk.

There are different modes of inspection with a UAV. 1) Fully autonomous mode: the UAV takes off, reaches a target, performs the task at target (delivers payload or inspects target), and finally returns to base. 2) Teleoperation mode: the aerial robot is remotely operated by a human pilot while the teleoperation loop is ensuring the stability and safety of the robot.

The teleoperation mode is particularly relevant when the visual inspection requires a human to make the assessment. In the teleoperation scheme, a human pilot controls the UAV with an input device, such as a joystick, using visual cues. However, visual evaluation of the distance between the vehicle and the obstacles is usually difficult. Therefore, even experienced pilots may fail to safely pilot the flying robot without collisions with obstacles. Moreover, in the vicinity of structures, complex aerodynamic effects induced by strong and unpredictable wind gusts may complicate the task of the human pilot.

To promote the use of UAVs as a tool for professional inspectors, the operation of an aerial robot should be straight-forward without the need of extensive training. Therefore, the goal of an efficient teleoperation scheme is to enable an inexperienced human pilot to perform complex and accurate inspection manoeuvres without touching obstacles despite aerological disturbances. To this end, at the core of every teleoperation system, there is 1) a controller that stabilizes the UAVs (unstable) rotational and translational dynamics, 2) an obstacle avoidance algorithm to ensure safe operation in presence of obstacles 3) feedback to the user, in order to provide the user accurate information on the surroundings of the UAV.

Several teleoperation schemes have been proposed to solve these requirements [1–7]. For example, in [8] awareness of obstacles in the UAV environment is rendered to the pilot through changing the stiffness of the joystick. In [3] an artificial force field translates the environmental constraints into force-feedback in the user's joystick. In [1], haptic force feedback is generated based on optical flow measurement data. Vision from onboard cameras is used to close the teleoperation feedback loop in [9] and [10], for example. The concept of a virtual slave UAV is introduced in [5] and extended in [7] to multidimensional and underactuated case. In [7], the concept of multi-state energy tank is introduced to ensure the passivity property of the teleoperation loop, by associating every action of the slave UAV with an energy expense, made available by the multi-state energy tank. A hierarchical control strategy is employed in which the high-level controller handles the teleoperation loop, whereas the low-level controller regulates the dynamics of the vehicle. The stability of each control level ensures the stability of the complete system.

The key contribution of this chapter is the design of an intuitive laser-based teleoperation scheme to enable the safe operation of a multirotor UAV by an untrained user in a cluttered environment. This scheme includes 1) a novel, laser-based force-feedback algorithm that enables the user to feel the texture of the environment, 2) a novel mapping function that allows to teleoperate the UAV in an unlimited workspace in position control mode with a joystick which has a limited workspace, and 3) a laser-based obstacle avoidance strategy which autonomously modifies the position setpoint of the UAV, independently of the pilot's commands. Moreover, a stability analysis proves the stability of the complete teleoperation loop made of the subsystems 1) master joystick and 2) slave UAV. Additionally, the description and implementation of the hardware and software architecture used aboard the Flybox hexacopter platform by the

company Skybotix is discussed. Finally, experiments demonstrate the performance of the teleoperation loop using an haptic joystick and a hexacopter UAV equipped with a 2D laser-range scanner as input for both obstacle avoidance and haptic rendering.

2 Bilateral Teleoperation Scheme

2.1 Architecture of the Teleoperation Loop

The considered bilateral haptic teleoperation scheme, as depicted in Fig. 1, consists of a fully-actuated 3 degrees of freedom (dof) haptic joystick (master) and an underactuated VTOL UAV (slave). The user interacts with the UAV and its surrounding environment using the haptic joystick by imposing a force \mathbf{F}_h on it.

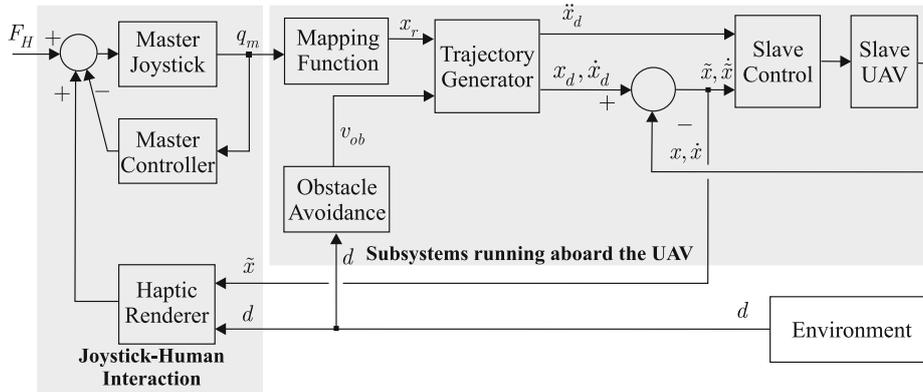


Fig. 1. Scheme of the bilateral teleoperation loop

A mapping function translates the position of the end effector of the haptic joystick $\mathbf{q}_m \in \mathbb{R}^3$ into a reference position for the UAV $\mathbf{x}_r \in \mathbb{R}^3$. To avoid obstacles, the mapping function also incorporates the output of the obstacle avoidance algorithm into the computation of the reference position \mathbf{x}_r . This reference position is then translated into a dynamically feasible reference trajectory that the position controller of the UAV tracks. In order to give the user some feedback about the operation of the UAV, a force vector \mathbf{F}_e is haptically generated on the joystick. The force vector is a function of the surrounding obstacles, the control error, etc.

2.2 Modeling of the Master Haptic Joystick

Let $\mathbf{q}_m \in \mathbb{R}^3$, $\dot{\mathbf{q}}_m \in \mathbb{R}^3$, and $\ddot{\mathbf{q}}_m \in \mathbb{R}^3$ respectively denote the position, velocity, and acceleration of the end effector of the haptic joystick, expressed in the

joystick frame \mathcal{F}^m . The joystick under consideration is a fully-actuated system. Thus, it can be described by the following Euler-Lagrange equation:

$$\mathbf{M}_m(\mathbf{q}_m)\ddot{\mathbf{q}}_m + \mathbf{C}_m(\mathbf{q}_m, \dot{\mathbf{q}}_m)\dot{\mathbf{q}}_m + \mathbf{g}_m(\mathbf{q}_m) = \mathbf{F}_m + \mathbf{F}_h + \mathbf{F}_e \quad (1)$$

where $\mathbf{M}_m(\mathbf{q}_m)$ the joystick inertia matrix, $\mathbf{C}_m(\mathbf{q}_m, \dot{\mathbf{q}}_m)$ representing the Coriolis and centrifugal effects, $\mathbf{g}_m(\mathbf{q}_m)$ the vector of gravitational forces, \mathbf{F}_m the local control force, \mathbf{F}_h the human force acting on the joystick, and \mathbf{F}_e denoting a haptic force generated on the joystick. For the sake of simplicity, the enclosed parameter(s) of $\mathbf{M}_m(\mathbf{q}_m)$, $\mathbf{C}_m(\mathbf{q}_m, \dot{\mathbf{q}}_m)$ and $\mathbf{g}_m(\mathbf{q}_m)$ are omitted subsequently. For later use, let us recall the following well-known property:

Property 1. (see [11]) The resulting matrix $\dot{\mathbf{M}}_m - 2\mathbf{C}_m$ is skew symmetric, i.e. $\forall \mathbf{x} \in \mathbb{R}^3, \mathbf{x}^\top (\dot{\mathbf{M}}_m - 2\mathbf{C}_m)\mathbf{x} = 0$.

2.3 Modeling of the Slave Multirotor UAV

The most basic multirotor helicopter configuration consists of a rigid airframe with two pairs of counter-rotating rigid propellers attached to it. The control of this platform is achieved by varying the rotational speed of the rotors. While such a four-rotor configuration already allows for full actuation of the vehicle's attitude, this approach can be easily extended to six- or eight-rotor configurations. In general, the configuration can be scaled up to an arbitrary number of rotors, however, the configuration should always consist of a multiple of counter-rotating rotor pairs for torque balancing reasons. In Fig. 2, a schematic of the Flybox hexacopter, described in Sec. 4, is depicted.

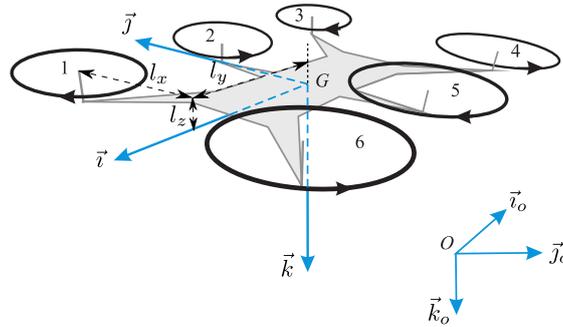


Fig. 2. Schematic representation of a six-rotor UAV platform. Inertial frame \mathcal{F}^0 with origin O and the body-fixed frame \mathcal{B} with origin G .

The following notation is introduced. The vehicle's center of mass (CoM) is denoted as G , its mass m , and its inertia matrix \mathbf{J} . Let $\mathcal{F}^0 = \{O; \vec{i}_o, \vec{j}_o, \vec{k}_o\}$ and $\mathcal{F}^s = \{G; \vec{i}, \vec{j}, \vec{k}\}$ denote the inertial frame and the body frame attached

to the vehicle, respectively. Let $\mathbf{x} \in \mathbb{R}^3$ denote the position of the vehicle's CoM expressed in \mathcal{F}^0 . The rotation matrix representing the orientation of the frame \mathcal{F}^s relative to the frame \mathcal{F}^0 is $\mathbf{R} \in SO(3)$. The vehicle's velocity and the wind velocity are both expressed in the frame \mathcal{F}^0 are denoted as $\dot{\mathbf{x}} \in \mathbb{R}^3$ and $\dot{\mathbf{x}}_w \in \mathbb{R}^3$, respectively. Let $\boldsymbol{\omega} \in \mathbb{R}^3$ be the angular velocity of the frame \mathcal{F}^s expressed in \mathcal{F}^s . The canonical basis of \mathbb{R}^3 is denoted $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. Let $\mathbf{d}_i = [d_{1,i} \ d_{2,i} \ d_{3,i}]^\top \in \mathbb{R}^3$ be the position of the i -th rotor expressed in the body-fixed frame \mathcal{F}^s . We define $\mathbf{d}_i^\perp = [d_{1,i} \ d_{2,i} \ 0]^\top \in \mathbb{R}^3$ as the component of \mathbf{d}_i perpendicular to \mathbf{e}_3 . Let the thrust direction of all rotors be parallel to \mathbf{e}_3 in \mathcal{F}^s . The notation \times represents the skew-symmetric matrix associated with the cross product, i.e. $\mathbf{u} \times \mathbf{v} = \mathbf{u} \times \mathbf{v}$, $\forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^3$. The Euclidean norm in \mathbb{R}^n is denoted as $|\cdot|$.

Dynamic Model of Vehicle

Following the model proposed in [26], the i -th rotor, turning at velocity ϖ_i , generates a thrust force $\mathbf{F}_{t,i} = c_T \varpi_i^2 \mathbf{e}_3$ and an aerodynamic torque $\mathbf{Q}_i = \lambda_i c_Q \varpi_i^2 \mathbf{e}_3$ with the aerodynamic constants c_T , c_Q and $\lambda_i = \{-1, 1\}$, depending on the direction of rotation of the rotor (cw: $\lambda_i = 1$, ccw: $\lambda_i = -1$). The remaining aerodynamic forces and torques (mostly due to drag by the fuselage) are summed up in a vector $\mathbf{F}_{aero} \in \mathbb{R}^3$ and $\boldsymbol{\Gamma}_{aero} \in \mathbb{R}^3$ respectively. The vehicle is subject to gravity $m\mathbf{g}\mathbf{e}_3$.

Applying the Newton-Euler formalism, one obtains the following equations of motion of the vehicle [21]:

$$\begin{cases} m\ddot{\mathbf{x}} = \mathbf{R} \sum_i \mathbf{F}_{t,i} + m\mathbf{g}\mathbf{e}_3 + \mathbf{F}_{aero} & (2a) \\ \dot{\mathbf{R}} = \mathbf{R}\boldsymbol{\omega}_\times & (2b) \\ \mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega}_\times \mathbf{J}\boldsymbol{\omega} + \sum_i (\mathbf{Q}_i + \mathbf{d}_i \times \mathbf{F}_{t,i}) + \boldsymbol{\Gamma}_{aero} & (2c) \end{cases}$$

2.4 Mapping of the Joystick Workspace to the UAV Workspace

Because of the limited joystick workspace and the unlimited UAV workspace, recent teleoperation schemes directly map the joystick position to the velocity setpoint of the UAV [6], [7]. This comes at the cost of not being able to perform precise, position controlled flights as, for example, needed in inspection tasks. We propose a novel mapping function between the joystick workspace and the UAV workspace that overcomes this limitation and enables position controlled UAV flight in an arbitrarily large UAV workspace.

In this scheme, as shown in Fig. 3, the position of the joystick's end effector \mathbf{q}_m is mapped to a reference position \mathbf{x}_r of the slave UAV using the mapping

$$\begin{cases} \mathbf{x}_r(t) & = \mathbf{K}_m \mathbf{q}_m(t) \min\left(1, \frac{r_*}{|\mathbf{q}_m(t)|}\right) + \mathbf{x}_c(t) \\ \mathbf{x}_c(t) & = \int_0^t \boldsymbol{\Psi}_m(\mathbf{q}_m(s)) ds \\ \boldsymbol{\Psi}_m(\mathbf{q}_m(t)) & = \mathbf{K}_v \mathbf{q}_m(t) \max\left(0, 1 - \frac{r_*}{|\mathbf{q}_m(t)|}\right) \end{cases} \quad (3)$$

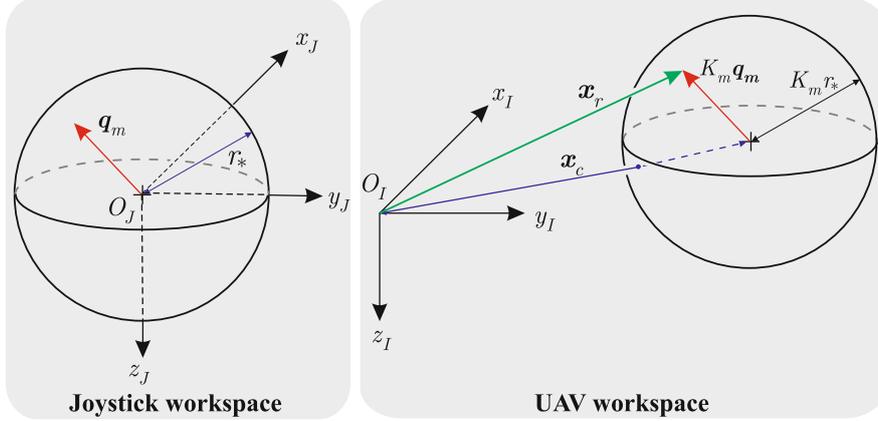


Fig. 3. Mapping of the joystick position \mathbf{q}_m in the joystick workspace (left) into the position reference \mathbf{x}_r of the UAV in the UAV workspace (right). In this figure, the diagonal scaling matrix \mathbf{K}_m has identical elements.

with \mathbf{K}_m , \mathbf{K}_v being two diagonal, positive definite scaling matrices. The term \mathbf{x}_c is the center of operation in the UAV workspace, around which the user can operate the UAV. While the user is operating the joystick end effector closer than the distance r_* to the origin of the joystick workspace, the joystick is within the “position sphere”. In this state, the center of operation \mathbf{x}_c remains constant since $\Psi_m(\mathbf{q}_m) = 0$. The mapping directly relates the joystick position \mathbf{q}_m to the UAV position setpoint around the center of operation by $\mathbf{x}_r = \mathbf{K}_m \mathbf{q}_m \frac{r_*}{|\mathbf{q}_m|} + \mathbf{x}_c$.

When the position of the joystick \mathbf{q}_m is moved outside of the “position sphere”, the center of operation starts moving in the direction of \mathbf{q}_m since \mathbf{x}_c is now evolving as the integral of $\Psi_m(\mathbf{q}_m) = \mathbf{K}_v \mathbf{q}_m (1 - \frac{r_*}{|\mathbf{q}_m|})$. The speed at which the center of operation is moving is determined by the matrix \mathbf{K}_v . When the end effector is moved back inside the sphere, the term \mathbf{x}_c remains constant again since $\Psi_m(\mathbf{q}_m) = 0$, thus resulting in a new center of operation around which the UAV can be operated.

2.5 Trajectory Generator

The implementation of the position controller of the UAV requires a position setpoint up to its second derivative¹. Since commercially available haptic joysticks only provide position and velocity of the end effector, the acceleration setpoint cannot be computed directly. Instead, we propose a low-pass filter to generate the setpoints

¹ For the computation of the term $\dot{\eta}_d$ of Eq. 20, the trajectory is required up to its third derivative (see e.g. [12] for explanations). Since we do not fly acrobatic maneuvers, we can neglect this term and therefore only estimate the setpoint up to its second derivative.

$$\begin{cases} \dot{\hat{\mathbf{x}}}_r := \text{sat}_{v_{\max}} \dot{\hat{\mathbf{x}}}_r \\ \ddot{\hat{\mathbf{x}}}_r := \text{sat}_{a_{\max}} (-k_d \dot{\hat{\mathbf{x}}}_r - k_p (\hat{\mathbf{x}}_r - \mathbf{x}_r)). \end{cases} \quad (4)$$

with the classical saturation function $\text{sat}_{\Delta}(\mathbf{x}) := \mathbf{x} \min(1, \Delta/|\mathbf{x}|)$. Using the saturation functions, we define maximal desired accelerations and velocities. Additionally, the aggressiveness of the trajectory generation can be tuned by adjusting the gains k_p and k_d .

Similarly, for the obstacle avoidance, we augment the obstacle velocity using a low-pass filter (see [4])

$$\dot{\hat{\mathbf{v}}}_{\text{ob}} = -k_p \hat{\mathbf{v}}_{\text{ob}} + \mathbf{v}_{\text{ob}}, \quad (5)$$

where \mathbf{v}_{ob} is specified in the next subsection. Finally, the UAV's reference trajectory is expressed as

$$\begin{cases} \mathbf{x}_d = \hat{\mathbf{x}}_r + \int \hat{\mathbf{v}}_{\text{ob}} dt \\ \dot{\mathbf{x}}_d = \dot{\hat{\mathbf{x}}}_r + \hat{\mathbf{v}}_{\text{ob}} \\ \ddot{\mathbf{x}}_d = \ddot{\hat{\mathbf{x}}}_r + \dot{\hat{\mathbf{v}}}_{\text{ob}} \end{cases} \quad (6)$$

2.6 Obstacle Avoidance

By assuming that the position error of the UAV remains small, we can work directly on the UAV setpoint as input to the obstacle avoidance scheme. The underlying idea of the obstacle avoidance scheme is to reshape the velocity setpoint of the UAV in a way that it avoids the surrounding obstacles [4].

For the operation of the obstacle avoidance algorithm, we assume to have a knowledge of the metric distance of the UAV setpoint to the surrounding obstacles. In this approach, the obstacles are represented using a sparse cloud of obstacle features. The obstacle features can come from a variety of sensors, e.g. measurement data from an onboard laser scanner or the point correspondences of a sparse monocular or stereo vision SLAM algorithm [16].

As depicted in Fig. 4, for each obstacle feature that is closer than some distance $d_i < d_{\star}$ from the reference position \mathbf{x}_d , a repelling velocity is computed as

$$\mathbf{v}_{\text{rep},i} = -\chi(d_i) \boldsymbol{\eta}_i, \quad (7)$$

where $\boldsymbol{\eta}_i$ is the unit vector pointing from the reference position \mathbf{x}_d to the obstacle feature. The function $\chi(d_i)$ is a smooth, non-increasing function that approaches infinity for d_i approaching the radius r_u of the UAV. An example of such a function is provided in Section 4.3. The resulting reference obstacle avoidance velocity \mathbf{v}_{ob} is computed as the average of all repelling velocities

$$\mathbf{v}_{\text{ob}} = \frac{1}{N} \sum_{i=1}^N \mathbf{v}_{\text{rep},i}. \quad (8)$$

Remark 1. In a particular case where only one planar obstacle (wall) is detected, the average repelling velocity can be approximated by $\mathbf{v}_{\text{ob}} = -\chi(d) \boldsymbol{\eta}$, where d is the distance between the reference position to the wall, and $\boldsymbol{\eta}$ is the unit normal

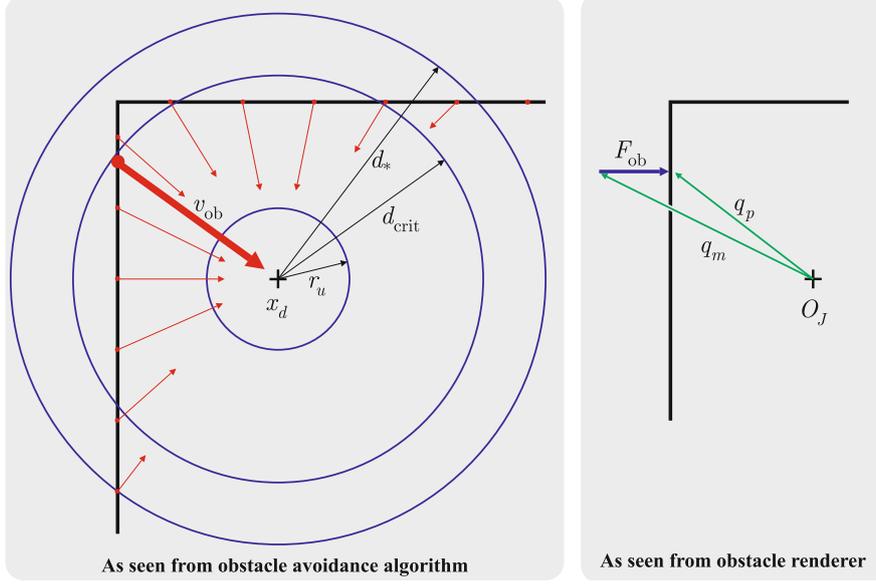


Fig. 4. Top-Down view of scenario where UAV is operated close to a corner. The left plot displays the obstacle avoidance algorithm. Each laser measurement (red dot) inside the circle with radius d_* generates a repulsive velocity \mathbf{v}_{rep} (red vector). The resulting obstacle velocity \mathbf{v}_{ob} (bold red vector) is the average of all \mathbf{v}_{rep} . On the right, the same scenario as seen from the obstacle renderer in the joystick workspace. The user tries to penetrate the virtual obstacle by setting \mathbf{q}_m inside the obstacle. The resulting haptic feedback \mathbf{F}_{ob} corresponds to a stiff spring between \mathbf{q}_m and its projected position on the obstacle surface \mathbf{q}_p .

vector pointing towards the wall and expressed in the frame \mathcal{F}^i . Noting that $\dot{d} = -\dot{\mathbf{x}}_d^\top \boldsymbol{\eta}$ and denoting $\sigma := -\dot{\mathbf{v}}_{ob}^\top \boldsymbol{\eta}$, one deduces from Eqs. (5) and (6) that

$$\begin{cases} \dot{d} = -\dot{\mathbf{x}}_r^\top \boldsymbol{\eta} + \sigma \\ \dot{\sigma} = -k_p \sigma + \chi(d) \end{cases} \quad (9)$$

From these equations, using the boundedness of $-\dot{\mathbf{x}}_r^\top \boldsymbol{\eta}$ and fact the non-increasing function $\chi(d)$ tends to infinity when d tends to r_u , one can prove the existence of a positive number $d_{min} > r_u$ such that $d(t) > d_{min}, \forall t > 0$, provided that $d(0) > d_{min}$. This means that the collision of the reference UAV and the wall is avoided. The proof of the property is based on a Lyapunov argument and can be found in [4]. On the other hand, one ensures that $\chi(d)$ and, consequently, v_{ob} remain bounded. Besides, one can also easily deduce from Eq. (9) the boundedness of σ , \dot{d} and, consequently, of $\dot{\mathbf{v}}_{ob}$. Then, from Eqs. (5) and (6), one also ensures the boundedness of $\dot{\mathbf{v}}_{ob}$, $\dot{\mathbf{x}}_d$, $\ddot{\mathbf{x}}_d$, and $\mathbf{x}_d^{(3)}$.

2.7 Haptic Rendering

In earlier works on bilateral haptic teleoperation (e.g. [3], [5], [6]), the environment was haptically rendered to the user using a potential wall function. The result was a sluggish and soft sensation of the environment since the user could only feel a gradually increasing force when approaching an obstacle. Therefore, we design a different strategy to give the user the sensation of feeling the rigid environment.

In a first step, a polygonal 3D model is generated from the obstacle features. The obstacle model is then mapped into the joystick workspace using the linear mapping

$$\mathbf{v}_j = \mathbf{K}_m^{-1}(\mathbf{v}_w - \mathbf{x}_d), \quad (10)$$

with \mathbf{v}_w being a vertex of the polygon in the world frame and \mathbf{v}_j the corresponding vertex in the joystick frame. When the user now penetrates this virtual object with the end effector, a stiff spring pulls the end effector back to the surface. This gives the user the sensation of touching the real environment as if he was located directly on top of the position reference of the UAV. This method is closely related to the god-object rendering method that was proposed in [17] to give the user of virtual reality simulation a haptic sensation of the virtual rigid objects. We define the obstacle force \mathbf{F}_{ob} as

$$\mathbf{F}_{ob} = -\text{sat}_{\Delta_{ob}} k(\mathbf{q}_m - \mathbf{q}_p), \quad (11)$$

with $k \gg 1$ and \mathbf{q}_p the projected position of \mathbf{q}_m on the surface. A graphical representation of the rendering process is displayed in Fig. 4. For the computation of \mathbf{q}_p , the reader is referred to [17].

On top of the obstacle force, we propose to haptically display the position control error as a spring force:

$$\mathbf{F}_{err} = -\text{sat}_{\Delta_1} k_1(\mathbf{x} - \mathbf{x}_d). \quad (12)$$

Using this spring force, the user can feel the inertia of the UAV when changing the setpoint. In general, all effects that cause a momentary position control error, such as external disturbances, are displayed using this force.

When doing a transition of the joystick's end effector from inside to outside the "position sphere", the user should feel a sensation that resembles the penetration of a membrane using a needle. Therefore, we construct a membrane force \mathbf{F}_{mem} , as depicted in Fig. 5, with which the user only feels the resistance of the virtual membrane when going outside the sphere but not when entering it back.

In the end, the environment force \mathbf{F}_e is constructed as the sum of all forces:

$$\mathbf{F}_e = \mathbf{F}_{err} + \mathbf{F}_{ob} + \mathbf{F}_{mem}. \quad (13)$$

At this point, it is to be noted that no component of the teleoperation scheme requires the use of a haptic joystick for proper operation. Since we use a velocity setpoint shaping approach for obstacle avoidance, the position reference of the UAV will be pushed away from the obstacle irrespective of the force feedback.

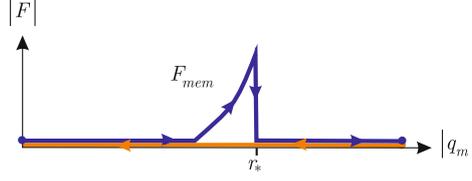


Fig. 5. Force profile to render the virtual membrane sensation

Therefore, the proposed teleoperation scheme can be interfaced and controlled using any input device such as a remote control or a tablet pc, in those cases without the need of generating haptic feedback.

3 Control Design

3.1 Control of the Master Joystick

The local control force \mathbf{F}_m of the master joystick is designed in order to ensure that the joystick's end effector is pushed back into the “position sphere” when the human and environmental forces are null (i.e. $\mathbf{F}_h = \mathbf{F}_e = 0$). Consequently, if the user releases the joystick, the UAV is stabilized in position. The following control expression of \mathbf{F}_m is proposed:

$$\begin{aligned} \mathbf{F}_m = & -(\mathbf{C}_m \mathbf{\Lambda}_1 + \lambda_2 \mathbf{I}_3) \Psi_m(\mathbf{q}_m) \\ & - \left(\mathbf{M}_m \mathbf{\Lambda}_1 \frac{\partial \Psi_m(\mathbf{q}_m)}{\partial \mathbf{q}_m} + \lambda_2 \mathbf{\Lambda}_1^{-1} \right) \dot{\mathbf{q}}_m + \mathbf{g}_m, \end{aligned} \quad (14)$$

with $\mathbf{\Lambda}_1 \in \mathbb{R}^{3 \times 3}$ a positive diagonal gain matrix and $\lambda_2 \in \mathbb{R}$ a positive gain.

3.2 Control of the Slave UAV

Model for Control Design and Rate Control

The UAV model (2) can be rewritten as

$$\begin{cases} \Sigma_1 : \begin{bmatrix} m \ddot{\mathbf{x}} \\ \dot{\mathbf{R}} \end{bmatrix} = \begin{bmatrix} -T \mathbf{R} \mathbf{e}_3 + \mathbf{F}_{ext} \\ \mathbf{R} \boldsymbol{\omega}_\times \end{bmatrix} & (15a) \\ \Sigma_2 : \mathbf{J} \dot{\boldsymbol{\omega}} = -\boldsymbol{\omega}_\times \mathbf{J} \boldsymbol{\omega} + \boldsymbol{\Gamma} + \boldsymbol{\Gamma}_{ext} & (15b) \end{cases}$$

where \mathbf{F}_{ext} and $\boldsymbol{\Gamma}_{ext}$ are the sum of all the acting forces and moments on the vehicle except the thrust force $T = c_T \sum_i \varpi_i^2$ and the torque $\boldsymbol{\Gamma} = \sum_i \lambda_i c_Q \varpi_i^2 \mathbf{e}_3 - c_T \varpi_i^2 \mathbf{d}_i^\perp \times \mathbf{e}_3$ generated by the rotors.

One can view $T \in \mathbb{R}^+$ and $\mathbf{F} \in \mathbb{R}^3$ as control inputs of System (15). For N mounted rotors, we can derive a linear mapping from the square of the propellers' angular velocity to the total thrust T and torque \mathbf{F} :

$$\begin{bmatrix} T \\ \mathbf{F} \end{bmatrix} = \begin{bmatrix} c_T & c_T & \dots & c_T \\ c_T d_{2,1} & c_T d_{2,2} & \dots & c_T d_{2,N} \\ -c_T d_{1,1} & -c_T d_{1,2} & \dots & -c_T d_{1,N} \\ \lambda_1 c_Q & \lambda_2 c_Q & \dots & \lambda_N c_Q \end{bmatrix} \begin{bmatrix} \varpi_1^2 \\ \varpi_2^2 \\ \vdots \\ \varpi_N^2 \end{bmatrix} \quad (16)$$

If $N = 4$ one can determine the desired angular rates of the rotors by inverting (16). When the UAV is actuated by more than four rotors, the set of equations (16) is overdetermined and the Moore-Penrose pseudo-inverse method can be used to determine the desired angular velocities of the propellers [23].

System (15) shows full actuation of the rotational dynamics and underactuation of the translational dynamics. For the rotational motion, exponential convergence of the angular velocity $\boldsymbol{\omega}$ to any bounded desired value $\boldsymbol{\omega}_d$ is easy to obtain, since the subsystem Σ_2 is fully actuated and the angular velocity vector $\boldsymbol{\omega}$ can be measured at high frequency from embedded gyrometers. A possible control solution is [27]

$$\mathbf{F} = -K_\omega(\boldsymbol{\omega} - \boldsymbol{\omega}_d) + \boldsymbol{\omega}_d \times \mathbf{J}\boldsymbol{\omega} + \mathbf{J}\dot{\boldsymbol{\omega}}_d, \quad (17)$$

with a sufficiently large diagonal positive gain matrix K_ω to dominate the disturbance torque \mathbf{F}_{ext} . From here on, all attention of control design can be given to the control of the subsystem Σ_1 using T and $\boldsymbol{\omega} \equiv \boldsymbol{\omega}_d$ as control inputs.

UAV Control Law

The control law applied to the UAV is designed such that the UAV's position x_s is stabilized at the desired value \mathbf{x}_d defined by (6) regardless of the dynamics of the master joystick. This controller is inspired by the one proposed in [12]. Let us denote $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d$ and $\dot{\tilde{\mathbf{x}}} = \dot{\mathbf{x}} - \dot{\mathbf{x}}_d$, the position and velocity error variables, respectively. Define $\boldsymbol{\eta} := \mathbf{R}\mathbf{e}_3$, and

$$\boldsymbol{\gamma} := h_p(|\tilde{\mathbf{x}}|)\tilde{\mathbf{x}} + h_v(|\dot{\tilde{\mathbf{x}}}|)\dot{\tilde{\mathbf{x}}} + \frac{\mathbf{F}_{ext}}{m} - \ddot{\mathbf{x}}_d, \quad (18)$$

with $h_p(\cdot)$ and $h_v(\cdot)$ some smooth bounded positive functions defined on $[0, +\infty)$ such that for some positive constants α_i, β_i (with index i being either p or v) (see e.g. [12])

$$|h_i(s^2)s| < \alpha_i, \quad 0 < \frac{\partial}{\partial s}(h_i(s^2)s) < \beta_i, \quad \forall s \in \mathbb{R}.$$

An example for the functions $h_p(\cdot)$ and $h_v(\cdot)$ is provided in Section 4.5.

Proposition 1. *Apply the control law*

$$\begin{cases} \mathbf{T} = m|\boldsymbol{\gamma}| \\ \boldsymbol{\omega}_{1,2} = \left(\mathbf{R}^\top \left(k_\eta \frac{\boldsymbol{\eta} \times \boldsymbol{\eta}_d}{(1 + \boldsymbol{\eta}^\top \boldsymbol{\eta}_d)^2} - (\boldsymbol{\eta}_\times)^2 (\boldsymbol{\eta}_d \times \dot{\boldsymbol{\eta}}_d) \right) \right)_{1,2} \end{cases} \quad (19)$$

to system (15a), with $\boldsymbol{\eta}_d := \frac{\boldsymbol{\gamma}}{|\boldsymbol{\gamma}|}$, k_η a positive gain, and the notation $\mathbf{u}_{1,2} = (u_1, u_2)^\top$, $\forall \mathbf{u} \in \mathbb{R}^3$. Assume that

- $\dot{\mathbf{x}}_d, \ddot{\mathbf{x}}_d, \mathbf{x}_d^{(3)}$ are known and bounded;
- $\boldsymbol{\gamma}$ is always different from the null vector;
- $\boldsymbol{\omega}_3$ is bounded.

Then, the equilibrium point $(\tilde{\mathbf{x}}, \dot{\tilde{\mathbf{x}}}, \boldsymbol{\eta}) = (0, 0, \boldsymbol{\eta}_d)$ of the controlled system is almost-globally asymptotically stable provided that $\boldsymbol{\eta}(0) \neq -\boldsymbol{\eta}_d(0)$.

Proof. Using (2) and (18) one verifies that the translational error dynamics satisfies

$$\begin{aligned} \ddot{\tilde{\mathbf{x}}} &= -h_p(|\tilde{\mathbf{x}}|)\tilde{\mathbf{x}} - h_v(|\dot{\tilde{\mathbf{x}}}|)\dot{\tilde{\mathbf{x}}} - \frac{1}{m_s}T\boldsymbol{\eta} + \boldsymbol{\gamma} \\ &= -h_p(|\tilde{\mathbf{x}}|)\tilde{\mathbf{x}} - h_v(|\dot{\tilde{\mathbf{x}}}|)\dot{\tilde{\mathbf{x}}} + \boldsymbol{\varepsilon} \end{aligned} \quad (20)$$

with $\boldsymbol{\varepsilon} := |\boldsymbol{\gamma}|(\boldsymbol{\eta} - \boldsymbol{\eta}_d)$. First, according to [12] one ensures the almost exponential convergence of $\boldsymbol{\eta}$ to $\boldsymbol{\eta}_d$. This is based on the analysis of the storage function $\mathcal{V}_\eta = 1 - \boldsymbol{\eta}^\top \boldsymbol{\eta}_d$ whose time-derivative along any solution to the controlled system satisfies (see [12])

$$\dot{\mathcal{V}}_\eta = -k_\eta \frac{|\boldsymbol{\eta} \times \boldsymbol{\eta}_d|^2}{(1 + \boldsymbol{\eta}^\top \boldsymbol{\eta}_d)^2}.$$

Due to the fact that $\boldsymbol{\gamma}$ is bounded, it follows that $\boldsymbol{\varepsilon}$ is bounded and converges exponentially to zero. Consequently, application of Lemma 1 in [15] on Eq. (20) ensures the convergence of $\tilde{\mathbf{x}}$ and $\dot{\tilde{\mathbf{x}}}$ to zero.

3.3 Stability of the Teleoperation Loop

The stability of the teleoperated system is studied next. First, we show that the master system is input-to-state stable (see [13]) in the presence of bounded operator force \mathbf{F}_h and environment force \mathbf{F}_e . Finally, for the free moving systems where the master and slave systems operate in free space, i.e. $\mathbf{F}_h = 0$ and $\mathbf{F}_e = \mathbf{F}_{err}$, we show that the joystick's position will be pushed back to the "position sphere" of radius r_\star and the slave UAV will asymptotically stop. For simplicity, the effect of the membrane force \mathbf{F}_{mem} is neglected in the stability analysis of the free moving systems case.

Proposition 2. *Consider the teleoperation system with the master system (1) controlled by the controller (14) and the slave system (15a) controlled by the controller (19). Assume that all assumptions in Proposition 1 are satisfied. Then, the results of Proposition 1 hold. Furthermore,*

- *In the case where the human and environment forces are bounded (i.e. $\exists \alpha_h$ and α_e such that $|\mathbf{F}_h(t)| \leq \alpha_h$ and $|\mathbf{F}_e(t)| \leq \alpha_e$, $\forall t$), the master system is input-to-state stable (I.S.S.) with respect to \mathbf{F}_h and \mathbf{F}_e .*

- In the case of free moving systems (i.e. $\mathbf{F}_h = 0$, $\mathbf{F}_e = \mathbf{F}_{err}$, $\mathbf{v}_{ob} = 0$), one ensures the existence of a constant vector $\mathbf{q}_\infty \in \mathbb{R}^3$ such that $|\mathbf{q}_\infty| \leq r_\star$ and $\lim_{t \rightarrow \infty} \mathbf{q}_m(t) = \mathbf{q}_\infty$. Moreover, the UAV will asymptotically stop.

Proof. Consider the following non-negative function

$$\begin{aligned} \mathcal{V}_m &= \frac{1}{2}(\dot{\mathbf{q}}_m + \mathbf{A}_1 \Psi_m(\mathbf{q}_m))^\top \mathbf{M}_m(\dot{\mathbf{q}}_m + \mathbf{A}_1 \Psi_m(\mathbf{q}_m)) \\ &\quad + 2\lambda_2 \mathbf{K}_m \int_0^{|\mathbf{q}_m(t)|} \max(0, s - r_\star) ds \end{aligned} \quad (21)$$

The time-derivative of \mathcal{V}_m satisfies

$$\begin{aligned} \dot{\mathcal{V}}_m &= (\dot{\mathbf{q}}_m + \mathbf{A}_1 \Psi_m(\mathbf{q}_m))^\top \mathbf{M}_m(\ddot{\mathbf{q}}_m + \mathbf{A}_1 \frac{\partial \Psi_m(\mathbf{q}_m)}{\partial \mathbf{q}_m} \dot{\mathbf{q}}_m) \\ &\quad + \frac{1}{2}(\dot{\mathbf{q}}_m + \mathbf{A}_1 \Psi_m)^\top \dot{\mathbf{M}}_m(\dot{\mathbf{q}}_m + \mathbf{A}_1 \Psi_m) \\ &\quad \quad \quad \underbrace{= (\dot{\mathbf{q}}_m + \mathbf{A}_1 \Psi_m(\mathbf{q}_m))^\top \mathbf{C}_m(\dot{\mathbf{q}}_m + \mathbf{A}_1 \Psi_m(\mathbf{q}_m))}_{=2\lambda_2 \Psi_m(\mathbf{q}_m)^\top \dot{\mathbf{q}}_m} \\ &\quad + k_X \mathbf{K}_m \max(0, |\mathbf{q}_m| - r_\star) \frac{\mathbf{q}_m^\top \dot{\mathbf{q}}_m}{|\mathbf{q}_m|} \\ &= (\dot{\mathbf{q}}_m + \mathbf{A}_1 \Psi_m)^\top (\mathbf{F}_e + \mathbf{F}_h - \lambda_2 (\mathbf{A}_1^{-1} \dot{\mathbf{q}}_m + \Psi_m)) + 2\lambda_2 \Psi_m^\top \dot{\mathbf{q}}_m \\ &= -\lambda_2 \dot{\mathbf{q}}_m^\top \mathbf{A}_1^{-1} \dot{\mathbf{q}}_m - \lambda_2 \Psi_m(\mathbf{q}_m)^\top \mathbf{A}_1 \Psi_m(\mathbf{q}_m) \\ &\quad - (\dot{\mathbf{q}}_m + \mathbf{A}_1 \Psi_m(\mathbf{q}_m))^\top (\mathbf{F}_e + \mathbf{F}_h) \end{aligned} \quad (22)$$

In view of (22) and the quadratic form of (21) there exists some positive constants α_1, α_2 such that

$$\dot{\mathcal{V}}_m \leq -\alpha_1 \mathcal{V}_m + \alpha_2 (|\mathbf{F}_e| + |\mathbf{F}_h|)$$

From here, if the environment force \mathbf{F}_e and the human force are bounded, then the function \mathcal{V}_m remains bounded. This in turn implies the boundedness of the master state variables \mathbf{q}_m and $\dot{\mathbf{q}}_m$. Besides, if \mathbf{F}_e and \mathbf{F}_h converge to zero (or are equal to zero), the application of the singular perturbation theorem (see e.g. [13]) ensures the convergence of \mathcal{V}_m to zero which in turn implies the convergence of $\Psi_m(\mathbf{q}_m)$ and $\dot{\mathbf{q}}_m$ to zero. This is the input-to-state stability property of the master loop with respect to \mathbf{F}_h and \mathbf{F}_e .

Now, consider the case of free moving systems where the human does not act on the joystick (i.e. $\mathbf{F}_h = 0$) and the slave is not in contact with the environment (i.e. $\mathbf{v}_{ob} = 0$, $\mathbf{F}_e = \mathbf{F}_{err}$). As a result of Proposition 1, the UAV's position error remains bounded and converges to zero, i.e. $\lim_{t \rightarrow \infty} \tilde{\mathbf{x}}(t) = 0$. This is independent on the master joystick's dynamics. From here one deduces that the environment force \mathbf{F}_e converges to zero since $\mathbf{F}_e = -\text{sat}_{\Delta_1}(k_1 \tilde{\mathbf{x}})$. As a consequence, one deduces (as proved previously) that $\dot{\mathbf{q}}_m$ and $\Psi_m(\mathbf{q}_m)$ converge to zero which means that the joystick's end effector converges to the “position sphere” and



Fig. 6. The Flybox hexacopter by Skybotix equipped with a Hokuyo UTM-30LN laser scanner. A mirror, mounted on top of the laser, is used to deflect some rays towards the ground for altitude estimation. The low-level autopilot and the high-level computer are inside the housing.

asymptotically stops. Then, by definition (3) the reference position x_r specified by the joystick also converges to a constant value. Since \mathbf{x}_r tends to a constant value, its augmented value $\hat{\mathbf{x}}_r$ converges to it. Besides, since $\mathbf{v}_{ob} = 0$ one has $\dot{\hat{\mathbf{v}}}_{ob} = -\kappa\hat{\mathbf{v}}_{ob}$. This implies the exponential convergence of $\hat{\mathbf{v}}_{ob}$ to zero. From here, using the definition (6) of \mathbf{x}_d one deduces that \mathbf{x}_d tends to a constant value. Finally, the UAV controller ensures that it will asymptotically stop at \mathbf{x}_d .

4 Implementation

4.1 System Setup

The teleoperation setup consists of

- The ground station (GS) computer.
- The 3 DoF fully-actuated haptic joystick Novint Falcon.
- The hexacopter UAV platform Flybox by Skybotix.
- The 2D laser range scanner Hokuyo UTM-30LN.

The Flybox UAV, shown in Fig. 6, is equipped with a low-level (LL) autopilot and a high-level (HL) computer. The LL autopilot is built around a Cortex M3 32-bit microprocessor and is equipped with a custom-made IMU. It controls the UAV's attitude by tracking desired thrust vector $\boldsymbol{\eta}_d$ from the HL computer. The autopilot also provides attitude and inertial sensor information at 1 kHz to the HL computer. The HL computer is an off-the-shelf Atom 1.6 Ghz Single-Core computer running Ubuntu 12.04 and consuming less than 7 W.

The Hokuyo UTM-30LN laser scanner is rigidly fixed to the UAV and is connected to the HL computer via USB. The laser scanner provides 1080 points per scan up to 30 m in a 270 degree window at 40 Hz.

The ground station is connected to the HL computer using WiFi. The Robotic Operating System (ROS) [14] is used as a communication layer between GS and HL and to run the different components of the teleoperation scheme. All the system-critical tasks, such as obstacle avoidance and control algorithms, are run aboard the UAV, in order to ensure the proper operation of the UAV even in case of loss of communication link between GS and HL computer. The GS only interfaces the joystick in order to 1) provide the haptic feedback to the user and 2) send the position and velocity setpoints from the joystick to the HL computer.

4.2 Laser Preprocessing

The onboard 2D laser scanner is used as input to the obstacle avoidance scheme as well as the haptic rendering loop. Both algorithms require a spatial model of the environment. While there are algorithms available to build 3D models from 2D laser range data, they come with the drawback that they are either not suited for real-time use [19] or only offer a coarse spatial resolution [18]. As a trade-off between speed and spatial resolution, it was decided to generate a 2D environment model by assuming that the environment consists of vertical walls. Using this assumption, the laser range measurements can be projected down from the UAV frame onto the x-y plane in the inertial frame using the attitude information from the onboard IMU. It is assumed that the ground coincides with the x-y plane in the inertial frame. As a consequence, laser measurements intersecting the ground plane are rejected, as in the case when the UAV is tilted and flying close to ground.

4.3 Obstacle Avoidance

The repelling velocity of each obstacle feature in the x-y plane of the inertial frame is computed using Eq. 7. For the derivation of the function $\chi(d)$, we assume to have a circular UAV with radius r_u ². We set

$$\chi(d) = \begin{cases} v_{\max} \frac{d-d_\star}{d_{\text{crit}}-d_\star} & \text{if } d_{\text{crit}} < d < d_\star \\ v_{\max} \frac{d_{\text{crit}}-r_u}{\max(\epsilon, d-r_u)} & \text{if } d < d_{\text{crit}} \end{cases} \quad (23)$$

with v_{\max} being the maximal allowed translational velocity of the trajectory generator and d_{crit} the distance at which we want the vehicle to stop. The term ϵ is present in (23) to prevent a division by zero. Let us quickly consider the 1D case where the UAV is perpendicularly approaching an infinitely long static vertical wall. Since the maximal allowed velocity is v_{\max} , the UAV will come to a stop at latest at d_{crit} because at this point $\chi(d_{\text{crit}}) = v_{\max}$ (see [4] for proof). However, since the UAV is operated in 3D, the average of all repulsive velocities

² When only considering obstacle avoidance during translational motion, this approach can be easily extended for non-circular UAVs by approximating them using an ellipsoid. However, in this case, obstacle avoidance during a yawing motion is not guaranteed with the current scheme.

are acting on the UAV. As a consequence, the UAV might get closer to the obstacle than d_{crit} . Therefore, when the $d < d_{\text{crit}}$, we shape $\chi(d)$ as a hyperbola approaching infinity as $d \rightarrow r_u$. This ensures that the position reference \mathbf{x}_r will, under no circumstances, get closer to an obstacle than r_u .

Since the laser scanner only covers 270 degrees field of view around the UAV, special precautions have to be taken when the user commands a position change into this blind spot. In such a case, the UAV is rotated first until the demanded position setpoint lies in a field of view of 180 degrees. Only then is the UAV allowed to approach the setpoint. The reduction from 270 to 180 degrees is necessary for safety reasons since the UAV is not a point and could therefore hit an obstacle if no safety margin would be introduced.

4.4 Haptic Rendering

The haptic rendering loop runs at 1 kHz to provide the user with a believable haptic sensation. To interface the Novint Falcon haptic joystick, the open source library *HAPI* by SenseGraphics is used. The library provides an interface where we can set the sum of the force vectors \mathbf{F}_e defined by (13) and \mathbf{F}_m defined by (14) as input. Note that here the matrix \mathbf{C}_m is set equal to zero for implementation simplicity. Finally, the total force vector is internally mapped into the corresponding motor torques of the joystick.

The library also provides an implementation of the god-object rendering algorithm. Using the projected laser measurements on the x-y plane in the inertial frame, it is straightforward to generate a polygonal 3D model with vertical surfaces.

4.5 Control

The LL autopilot developed by the company Skybotix tracks the defined thrust vector $\boldsymbol{\gamma}$ using Eq. 19. As a consequence, when implementing the position controller, the UAV can be considered a fully-actuated point-mass with 3 DoF force control inputs. This is a valid assumption, as long as the time scale separation between attitude and position controller is ensured. As a rule of thumb, the time constant of the attitude controller should be one magnitude larger than the time constant of the position controller.

Following Eq. 18, we define the thrust vector as

$$\boldsymbol{\gamma} = \text{sat}_{\Delta_p}(k_p \tilde{\mathbf{x}}) + \text{sat}_{\Delta_v}(k_d \dot{\tilde{\mathbf{x}}}) + g\mathbf{e}_3 - \ddot{\mathbf{x}}_d \quad (24)$$

An integral term could be included in the expression. The interested reader is referred to [12]. The control gains are determined via a pole placement procedure performed on the linearized system of system (2) at hovering. Details on the gain-tuning process can be found in [20, Ch. 2].

The gain k_p of the trajectory generator relates to the responsiveness of the generator to position inputs. The gain k_d relates to the damping of the generator and is set to $k_d = 2\sqrt{k_p}$ to ensure critically-damped trajectory tracking without overshoot.

5 Experimental Results

The performance of the proposed teleoperation scheme is evaluated via several experiments in an indoor environment. For the stabilization of the UAV position, a Vicon motion tracking system is used. While the position of the UAV could be stabilized using the laser scanner, we use the Vicon system instead, since we want to evaluate the performance of the teleoperation scheme rather than the laser position estimator. For laser-based UAV stabilization we refer the reader to [22] and to the widely used open source implementation of the algorithms [24].

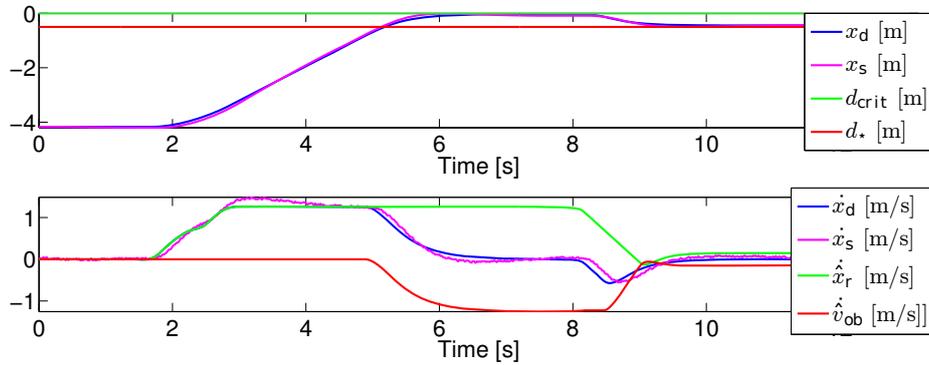


Fig. 7. 1D representation of UAV approaching a static vertical wall. The upper plot represents the approach of the UAV position setpoint \mathbf{x}_d (blue) towards the wall at $\mathbf{x} = 1\text{ m}$. The red and green line in the upper plot represent the distance d_* and d_{crit} from the wall. The middle plot shows the velocity input from the user $\dot{\mathbf{x}}_r$ (green), the obstacle velocity $\dot{\mathbf{v}}_{ob}$ (red) and the resulting reference velocity for the UAV $\dot{\mathbf{x}}_d$ (blue). The magenta line corresponds to the position \mathbf{x} and velocity $\dot{\mathbf{x}}$ of the UAV.

The UAV velocity is estimated from the Vicon position data using a linear observer. The onboard laser scanner is used for both obstacle avoidance and haptic rendering. The numerical values used in the teleoperation scheme for the experiments are depicted in Table 5. A video recording of all experiments are on the homepage [25].

Table 1. Numerical values of the obstacle avoidance scheme used in the experiments

d_*	d_{crit}	r_u	v_{max}
1.5 m	1.0 m	0.5 m	2.0 m/s

In a first experiment, we evaluate the performance of the obstacle avoidance algorithm when approaching a vertical wall perpendicularly. The wall is at $\mathbf{x} = 1\text{ m}$ in the Vicon frame. A 1D representation of the experiment is depicted in Fig. 7.

The experiment starts with the UAV approaching the wall from 4 m . At time $t = 5.0\text{ s}$, the UAV setpoint gets closer than d_\star to the obstacle. The obstacle avoidance starts injecting a repelling velocity $\hat{\mathbf{v}}_{ob}$ in normal direction of the wall. Then, the translation of the UAV setpoint in direction of the wall comes to a stop at $t = 6.5\text{ s}$ at distance d_{crit} without any oscillation. Now, the repelling velocity $\hat{\mathbf{v}}_{ob}$ is counteracting the user defined velocity $\hat{\mathbf{x}}_r$. At $t = 8.0\text{ s}$, the user releases the joystick. The UAV is pushed back to distance d_\star because of the velocity term $\hat{\mathbf{v}}_{ob}$. Concerning the performance of the position controller, the UAV is able to track the position reference without overshoot. The lowest plot shows the angular control error of thrust vector $\boldsymbol{\eta}_d$ and $\boldsymbol{\eta}$. Attitude control is asymptotically stable as required for the derivation of the position controller.

In a second experiment, the performance of the obstacle avoidance algorithm is evaluated when entering a narrow hallway. This is often problematic with potential-wall like obstacle avoidance approaches. A top-down view of the experiment is depicted in Fig. 8.

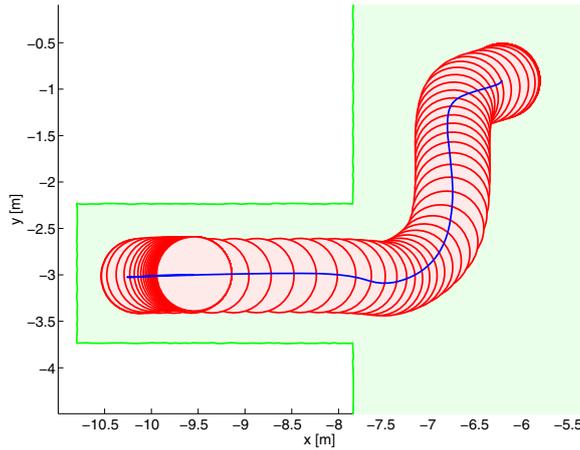


Fig. 8. Top-Down view of UAV entering a narrow hallway with a width of 1.5 m . The green line represents the laser scan taken at startup of the experiment. The red spheres represent snapshots of the position setpoint \mathbf{x}_d in 0.5 s intervals. The blue line is the trajectory of the UAV position \mathbf{x}_s .

As the UAV is approaching the wall, the obstacle avoidance starts injecting a repelling velocity $\hat{\mathbf{v}}_{ob}$ in normal direction of the wall. As a consequence, the position setpoint is sliding along the wall until it enters the hallway opening. When the UAV is inside the hallway, it is able to fly at maximal velocity in direction along the hallway. This is due to the fact that the average repelling velocity has no component along the hallway when the UAV is completely in this one. When the end of the hallway is reached, the user releases the joystick. The UAV position setpoint stabilizes at the position sufficiently away from the obstacles.

We also evaluate the performance of the obstacle avoidance algorithm when exposed to moving human obstacles. For space reasons, we omit this experiment in the chapter and refer the reader to the video on [25].

6 Conclusions

In this chapter, a safe teleoperation scheme has been presented for a wide range of VTOL UAVs operated by untrained pilots. A force-feedback algorithm generates a force to a haptic joystick that enables the user to feel the texture of the environment. Although the joystick has a limited workspace, a novel mapping function enables the teleoperation of the UAV in an unlimited workspace in position control mode. The obstacle avoidance strategy presented autonomously modifies the position setpoint of the UAV independently of the pilot's commands. The stability analysis of the whole teleoperation loop is proven. Experiments showed the successful teleoperation of a UAV using a haptic joystick and a hexacopter UAV equipped with a 2D laser-range scanner.

Acknowledgment. This work was supported by the EU project EC FP7-ICT-231143 ECHORD. Additionally, the authors acknowledge the financial support by the SCAR ANR-ASTRID fund.

References

1. Mahony, R., Schill, F., Corke, P., Oh, Y.S.: A new framework for force feedback teleoperation of robotic vehicles based on optical flow. In: *IEEE Int. Conf. on Robotics and Automation*, pp. 1079–1085 (2009)
2. Schill, F., Mahony, R., Corke, P., Cole, L.: Visual Force Feedback Teleoperation of the Insectbot Using Optic Flow. In: *Australian Conf. on Robotics and Automation* (2009)
3. Lam, T.M., Boschloo, H.W., Mulder, M., van Passen, M.M.: Artificial Force Field for Haptic Feedback in UAV Tele-operation. *IEEE Trans. on Systems, Man, and Cybernetics- Part A: Systems and Humans* 39(6), 1316–1330 (2009)
4. Hua, M.-D., Rifai, H.: Obstacle Avoidance for Teleoperated Underactuated Aerial Vehicles using Telemetric Measurement. In: *IEEE Conf. on Control and Decision*, pp. 262–267 (2010)
5. Stramigioli, S., Mahony, R., Corke, P.: A Novel Approach to Haptic Teleoperation of Aerial Robot Vehicles. In: *IEEE Int. Conf. on Robotics and Automation*, pp. 5302–5308 (2010)
6. Rifai, H., Hua, M.-D., Hamel, T., Morin, P.: Haptic-based Bilateral Teleoperation of Underactuated Unmanned Aerial Vehicles. In: *IFAC World Congress, Italy*, pp. 13782–13788 (2011)
7. Mersha, A.Y., Stramigioli, S., Carloni, R.: Bilateral Teleoperation of Underactuated Unmanned Aerial Vehicles: The virtual Slave Concept. In: *IEEE Int. Conf. on Robotics and Automation, Minnesota*, pp. 4614–4620 (2012)
8. Lam, T.M., Mulder, M., Van Paassen, M.M.: Collision Avoidance in UAV Teleoperation with Time Delay. In: *IEEE Int. Conf. on Systems, Man and Cybernetics*, pp. 997–1002 (2007)

9. Mettler, B., Andersh, J., Papanikolopoulos, N.: A First Investigation into the Teleoperation of a miniature Rotorcraft. In: Khatib, O., Kumar, V., Pappas, G.J. (eds.) *Experimental Robotics*. STAR, vol. 54, pp. 191–199. Springer, Heidelberg (2009)
10. Suzuki, R., Matsumoto, T., Konno, A., Hoshino, Y., Go, K., Oosedo, A., Uchiyama, M.: Teleoperation of a Tail-sitter VTOL UAV. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1618–1623 (2010)
11. Spong, M.W., Hutchinson, S., Vidyasagar, M.: *Robot modeling and control*. John Wiley & Sons, Inc. (2006)
12. Hua, M.-D., Hamel, T., Morin, P., Samson, C.: A Control Approach for Thrust-Propelled Underactuated Vehicles and its Application to VTOL Drones. *IEEE Trans. on Automatic Control* 59(8), 1837–1853 (2009)
13. Khalil, H.K.: *Nonlinear Systems*, 3rd edn. Prentice Hall (2002)
14. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A.: ROS: An open-source robot operating system. In: *ICRA Workshop on Open Source Software* (2009)
15. Abdessameud, A., Tayebi, A.: Formation control of VTOL Unmanned Aerial Vehicles with communication delays. *Automatica* 47, 2383–2394 (2011)
16. Weiss, S., Achtelik, M., Kneip, L., Scaramuzza, D., Siegwart, R.: Intuitive 3D Maps for MAV Terrain Exploration and Obstacle Avoidance. *Journal of Intelligent and Robotic Systems* 61, 473–493 (2011)
17. Zilles, C.B.: A constraint-based god-object method for haptic display. In: *IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 146–151 (1995)
18. Dryanovski, I., Morris, W., Xiao, J.: Multi-Volume Occupancy Grids: an Efficient Probabilistic 3D Mapping Model for Micro Aerial Vehicles. In: *IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 1553–1559 (2010)
19. Hostenstein, C., Zlot, R., Bosse, M.: Watertight surface reconstruction of caves from 3D laser data. In: *IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 3830–3837 (2011)
20. Hua, M.-D.: *Contributions to the Automatic Control of Aerial Vehicles*. Ph.D. thesis, University of Nice-Sophia Antipolis (2009)
21. Hua, M.-D., Hamel, T., Morin, P., Samson, C.: A control approach for thrust-propelled underactuated vehicles and its application to VTOL drones. *IEEE Transactions on Automatic Control* 54(8), 1837–1853 (2009)
22. Achtelik, M., Bachrach, A., He, R., Prentice, S., Roy, N.: Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments. In: *Unmanned Systems Technology XI* (2009)
23. Ducard, G., Hua, M.-D.: Discussion and practical aspects on control allocation for a multi-rotor helicopter. In: *Conference on Unmanned Aerial Vehicle in Geomatics*, pp. 1–6 (2011)
24. Dryanovski, I., Morris, W., Xiao, J.: An open-source pose estimation system for micro-air vehicles. In: *IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 4449–4454 (2011)
25. I3S Institute Homepage, <http://tinyurl.com/i3s-cnrs> (accessed 11:16 CET, May 30, 2013)
26. Hamel, T., Mahony, R., Lozano, R., Ostrowski, J.: Dynamic modelling and configuration stabilization for an X4-flyer. In: *IFAC World Congress*, pp. 200–212 (2002)
27. Rudin, K., Hua, M.-D., Ducard, G., Bouabdallah, S.: A Robust Attitude Controller and its Application to Quadrotor Helicopters. In: *IFAC World Congress*, pp. 10379–10384 (2011)