Preprint version 2nd IFAC Workshop RED-UAS, France (2013)

Vision-based Autonomous Control of a Quadrotor UAV using an Onboard RGB-D Camera and its Application to Haptic Teleoperation

Paolo Stegagno * **Massimo Basile** * **Heinrich H. Bülthoff** *
Antonio Franchi *

* *Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany,*
(e-mail: {paolo.stegagno,antonio.franchi,hhb}@tuebingen.mpg.de)

Abstract: In this paper we present the design of a platform for autonomous navigation of a quadrotor UAV based on RGB-D technology. The proposed platform can safely navigate in an unknown environment while self-stabilization is done relying only on its own sensor perception. We developed an estimation system based on the integration of IMU and RGB-D measurements in order to estimate the velocity of the quadrotor in its body frame. Experimental tests conducted as teleoperation experiments show the effectiveness of our approach in an unstructured environment.

1. INTRODUCTION

The unconstrained workspace and versatility of Micro UAVs allow to use them as flying sensors and actuators to reach and operate on places that are out of the range of more classical ground mobile robots. Hence, they constitute the ideal platform for many robotic tasks, such as exploration, mapping and surveillance. Many recent works highly rely on the presence of precise external tracking systems. However, for real autonomy the UAVs must be able to navigate relying only on the perceptions from their on-board sensors.

The development of a platform for safe autonomous or semi-autonomous navigation includes estimation of the current state of the UAV, control, sensing, obstacle avoidance and others. In addition, depending on the application, it may also require high level control and planning in case of a fully autonomous system, or visual and haptic feedback in case of teleoperation, e.g., search-and-rescue missions (Murphy et al., 2008).

Many research groups around the world have proposed their own systems. Several works (Weiss et al., 2011; Engel et al., 2012) relies on monocular SLAM paradigm in order to obtain information on the status of the robot and navigate in the environment. Shaojie et al. (2011) employ a richer sensor equipment (kinect, range sensor) but still integrates a SLAM module. Similarly, Grzonka et al. (2012) developed a quadrotor able to navigate in a map obtained through the measurements of a laser range sensor. Many other works employs sensor fusion. Schmid et al. (2012) relies on integration of IMU, range measurements and stereo vision in order to navigate both indoor and outdoor. Bry et al. (2012) performs the state estimation of a UAV based on range-IMU fusion through Gaussian Particle Filtering, while Nieuwenhuisen et al. (2013) presents their platform equipped with GPS, two stereo cameras, a rotating laser scanner and multiple ultrasonic sensors.

The goal of this paper is therefore to present our UAV platform designed for autonomous or semi-autonomous navigation using velocity control in real unstructured scenarios providing safety against obstacles and relying on onboard sensors only, namely IMU and RGB-D measurements. This essential sensorial equipment, thanks to the presence of a depth camera, is relatively richer with respect to the standard IMU-(mono)camera integration setting. This choice brings several advantages but also some drawbacks.

First, the depth measurements are extremely useful because they allow a metric estimation of the velocity. Monocular camera methods, as, e.g., the ones based on PTAM (Scaramuzza et al., 2013), do not provide metric information directly and typically need additional sensor fusion with the accelerometer reading, thus requiring a persistently accelerated motion to properly work metrically. In addition, the measurements coming from an RGB-D sensor can be easily used to perform reliable obstacle avoidance directly using the dense cloud of obstacle points obtained from the sensor. On the contrary, RGB-D sensors are usually sensible to natural light, so our system is specifically designed for indoor navigation. To overcome this issue, we are considering to substitute the RGB-D sensor with a stereo camera.

In order to test the developed platform, we will employ it in teleoperation experiments with visual and haptic feedback. Recent works have investigated the role of haptic feedback and the fact that it can be successfully used in order to increase the operator situational awareness (see, e.g., Lam et al. (2009) and references therein) and therefore to have a positive impact on the human decisions. For this reason haptic shared control of UAVs represents an emerging topic attracting the attention of many research groups.

Concerning the single-UAV case, an extensive study has been already done, with special regard on the theoretical point of view. Lam et al. (2009) have proposed the use of

Fig. 1. The quadrotor setup.

artificial force fields, while Schill et al. (2010) presented the design of an admittance control paradigm from the master side with position feedback. Single-UAV teleoperation control based on the port-Hamiltonian approach has been presented by Stramigioli et al. (2010) and extended by Mersha et al. (2012). Rifa et al. (2011) designed a strategy to generate the haptic feedback as a virtual force based on both telemetric and optic flow sensors. A novel force feedback user interface for mobile robotic vehicles with dynamics has been shown by Hou et al. (2013), and a novel force feedback algorithm that allows the user to feel the texture of the environment has been recently presented by Omari et al. (2013).

Concerning the, more challenging, multi-UAV case, Franchi et al. (2012b,a,c); Lee et al. (2013) presented an extensive framework to control a group of UAVs that can be interfaced with multiple operators by means of haptic devices. Riedel et al. (2012) have shown how that framework can be applied in the real world to perform teleoperation over intercontinental distances.

The majority of the works never addressed the problem in a real world scenario, either employing simulation or external motion capture systems. Even though in (Omari et al., 2013) the obstacles are detected through a laser scanner the state for control purpose is still retrieved by an external camera system. Similarly, Franchi et al. (2012a) used on-board cameras to measure the relative bearings, but the velocities were obtained through an external motion capture system. At the best of our knowledge none of the approaches dealing with haptic-teleoperation of UAVs have been experimentally proven on a platform that uses onboard sensors only.

The rest of the paper is organized as follows. Section 2 presents the hardware architecture. Section 3 presents the flight controller, velocity estimator and teleoperation system. Section 4 presents some experimental results and Section 5 concludes the paper.

2. HARDWARE ARCHITECTURE

The quadrotor configuration of this work comprises the mechanical frame, actuators, microcontrollers, and inertial measurement unit (IMU) of the MK-Quadro, a relatively low cost platform. [1] Its actuation system consists of four plastic propellers with a diameter of 0.254 m, and a total span and weight of the frame of 0.5 m and 0.12 kg, respectively. The on-board electronics consists of:

i. a single board mini PC Odroid U2 [2] for high level estimation, control, and interfacing purposes. Its features include a 1.7 GHz Quad core ARM Cortex-A9 MPCore, a MicroSD slot and three USB connectors;
ii. a low-level 8-bit Atmega1284p microcontroller, clocked at 20 MHz, connected to the mini-computer through two RS232 serial ports and a MAX232 converter. The serial connections operate at a baud-rate of 115 200 Bd. The manufacturer provides the board preinstalled with its own firmware to drive the quadrotor with a remote control, which we have substituted adding new features and changing the interfacing to control the robot through the serial connection;
iii. four brushless controllers connected to the low-level controller through a standard I2C bus;
iv. three 3D LIS344alh accelerometers (0.0039g0 m/s2 resolution and 2g0 m/s2 range) and three ADXRS610 gyros (0.586 deg/s resolution and 300 deg/s range), directly connected to the analog to digital 10 bit converters of the low-level microcontroller;
v. a pressure sensor MPX4115A.

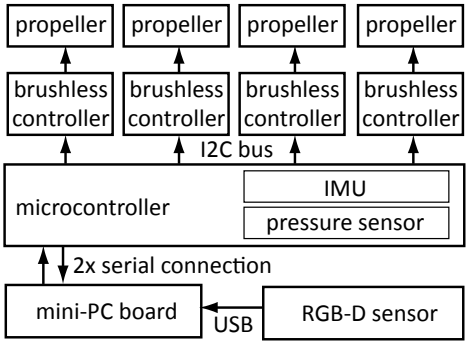

Fig. 2. A block scheme representation of the main components. At the actual implementation stage the mini-PC board is not mounted on the quadrotor yet. On the other hand, all the used sensors are mounted onboard (no external motion capture system is used).

In addition, we have retrofitted the MK-Quadro frame with an Asus XtionRGB-D sensor to obtain exteroceptive measurements of the environment. The RGB-D sensor, from now on referred to simply as 'camera', is rigidly attached to the frame through three 5 mm diameter plastic bars, heading approximatively at 45° on the right of the quadrotor and tilted by approximatively 30° downward, vertically mounted to increase the field of view on the Z axis. In Fig. 1 we report a picture of the whole system and a block scheme of the electronic components is given in Fig. 2.

The whole system is powered by a 2600 mAh LiPo battery which guarantees an endurance of around 10 min of flight in normal regimes. The complete system has a weight of approximately 1.000 kg.

At current state of development, the Odroid board is still not integrated in the system. Its role is temporarily exploited by an external computer which communicates with the quadrotor through two XBee channels instead of wired serial connectors.

3. FLIGHT CONTROL, ESTIMATION AND TELEOPERATION

Figure 3 gives a representation of the relevant frames used in the platform development and discussed in the

[1] http://www.mikrokopter.de/
[2] http://www.hardkernel.com/

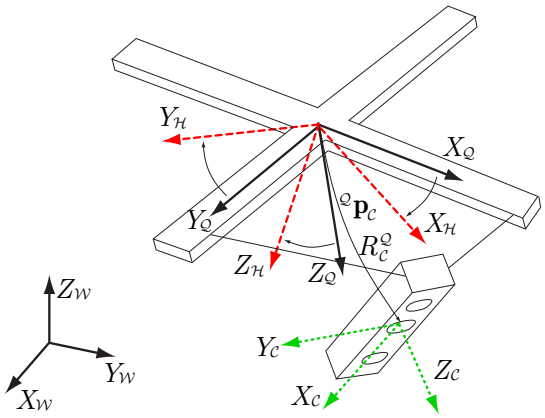

Fig. 3. A representation of all frames involved in the control and estimation of the velocity of the quadrotor.

following. Let $\mathcal{W} : \{O_{\mathcal{W}}, X_{\mathcal{W}}, Y_{\mathcal{W}}, Z_{\mathcal{W}}\}$ be the inertial (world) frame defined with the North-West-Up (NWU) convention, hence with $Z_{\mathcal{W}}$ pointing in the opposite direction of the gravity vector, and let be $\mathcal{Q} : \{O_{\mathcal{Q}}, X_{\mathcal{Q}}, Y_{\mathcal{Q}}, Z_{\mathcal{Q}}\}$ a frame attached to a representative point of the quadrotor (ideally its center of mass), which conforms to the North-East-Down (NED) convention as common in the aerospace field. In general, we will denote with ${}^{\mathcal{A}}\mathbf{p}_{\mathcal{B}}$ the position of the origin of a frame $\mathcal{B}$ in another frame $\mathcal{A}$ and with $\mathbf{R}^{\mathcal{A}}_{\mathcal{B}} \in SO(3)$ the rotation matrix expressing the orientation of the frame $\mathcal{B}$ in $\mathcal{A}$. With reference to the frames $\mathcal{W}$ and $\mathcal{B}$ we then define ${}^{\mathcal{W}}\mathbf{p}_{\mathcal{Q}} \in \mathbb{R}^3$ and $\mathbf{R}^{\mathcal{W}}_{\mathcal{Q}} \in SO(3)$. Finally, denote with ϕ, θ, ψ respectively the roll, pitch and yaw angles that represent the orientation of the quadrotor in $\mathcal{W}$, i.e., such that $\mathbf{R}^{\mathcal{W}}_{\mathcal{Q}} = \mathbf{R}_x(\pi)\mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi)$, where $\mathbf{R}_x(\cdot)$, $\mathbf{R}_y(\cdot)$, $\mathbf{R}_z(\cdot)$ represent the canonical rotation matrices about the axes X, Y, and Z respectively.

It is well known that roll and pitch cannot be chosen independently from the cartesian motion of the quadrotor center of mass. Any yaw motion can instead be commanded while following a 3D trajectory. Therefore, the external motion commands [3] are expressed in a (NED) horizontal frame $\mathcal{H} : \{O_{\mathcal{H}}, X_{\mathcal{H}}, Y_{\mathcal{H}}, Z_{\mathcal{H}}\}$ such that $O_{\mathcal{H}} \equiv O_{\mathcal{Q}}$ and $Z_{\mathcal{H}} \parallel -Z_{\mathcal{W}}$. Then, the rotation matrix between $\mathcal{H}$ and $\mathcal{Q}$ is $\mathbf{R}^{\mathcal{Q}}_{\mathcal{H}} = \mathbf{R}_z(\theta)\mathbf{R}_x(\phi)$.

Finally, consider the camera frame $\mathcal{C} : \{O_{\mathcal{C}}, X_{\mathcal{C}}, Y_{\mathcal{C}}, Z_{\mathcal{C}}\}$. Since the camera is rigidly attached to the quadrotor, ${}^{\mathcal{Q}}\mathbf{p}_{\mathcal{C}}$ and $\mathbf{R}^{\mathcal{Q}}_{\mathcal{C}}$ are constant extrinsic parameters.

3.1 Obstacle avoidance and velocity tracker

Assume the UAV is performing a given task which requires to track a desired velocity $\mathbf{v}^*_{des}$ at each time instant t. In order to guarantee safe navigation and avoid contacts with the objects in the environment we have implemented a simple obstacle detection and avoidance module able to modify the desired velocity before it is passed to the velocity tracker. In particular, at each time t, the relevant parts of the surrounding obstacles are detected by finding the local minima in the depth-component of the camera image. Then, a standard repulsive potential is implemented for each local minimum in order to avoid contact, which generates a total repulsive velocity term denoted with $\mathbf{v}_{obs}$.

[3] External motion commands can be, e.g., generated by a guidance algorithm or a human operator.

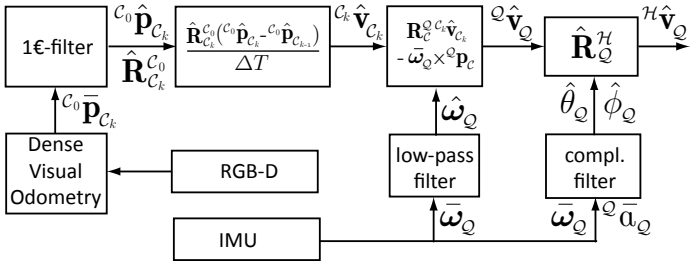

Fig. 4. A block scheme representation of the velocity estimation system.

The modified desired velocity $\mathbf{v}_{des} = \mathbf{v}^*_{des} + \mathbf{v}_{obs}$ is provided to the flight controller (referred to as 'velocity tracker' in the following) that uses also the current estimated velocity of the robot to compute the tracking error. The velocity tracker, described in (Lee et al., 2013), is a simple PID controller with gravity compensation regulating the thrust and the roll and pitch angles in order to accelerate as requested and therefore be able to track the desired velocity. Note that, unlike most PID controllers employed in navigation issues, the integral error is computed numerically integrating the error on the velocity.

3.2 Velocity estimation

The working principle of our estimation system is summarized in Fig. 4, and uses both measurements of the IMU and of the depth-camera. The first ones are used in a complementary filter to compute estimates $\hat{\phi}_{\mathcal{Q}}$, $\hat{\theta}_{\mathcal{Q}}$ of the roll and pitch angles as described in (Mahony et al., 2008; Martin and Salaün, 2010). In addition, a low-pass filter improves the angular velocity measurements ${}^{\mathcal{Q}}\bar{\omega}_{\mathcal{Q}}$ from the gyros producing an estimate denoted with ${}^{\mathcal{Q}}\hat{\omega}_{\mathcal{Q}}$.

Once the attitude (i.e., roll and pitch) of the quadrotor is known, the images from the depth camera can be used to obtain an estimate of the velocity of the quadrotor in the frame $\mathcal{H}$ as described in the following.

At each time-step k the images are used to feed the dvo [4] algorithm (Kerl et al., 2013) which provides the estimates ${}^{\mathcal{C}_0}\bar{\mathbf{p}}_{\mathcal{C}_k}$, $\bar{\mathbf{R}}^{\mathcal{C}_0}_{\mathcal{C}_k}$ of the position ${}^{\mathcal{C}_0}\mathbf{p}_{\mathcal{C}_k}$ and orientation $\mathbf{R}^{\mathcal{C}_0}_{\mathcal{C}_k}$ of the camera frame $\mathcal{C}_k$ at time-step k w.r.t. the camera frame $\mathcal{C}_0$ at time-step 0. Obviously, since dvo performs a visual odometry algorithm, the estimates will eventually diverge from the true value and cannot be used for a long time to obtain absolute position and orientation measurements. Nevertheless, it is possible to extract a noisy but non-drifting measurement of the velocity ${}^{\mathcal{C}_k}\mathbf{v}_{\mathcal{C}_k}$, i.e., the velocity of the origin $O_{\mathcal{C}_k}$ of the frame $\mathcal{C}_k$ expressed in $\mathcal{C}_k$, through the equation:

$$ {}^{\mathcal{C}_k}\mathbf{v}_{\mathcal{C}_k} = \frac{\mathbf{R}^{\mathcal{C}_k}_{\mathcal{C}_0}({}^{\mathcal{C}_0}\mathbf{p}_{\mathcal{C}_k} - {}^{\mathcal{C}_0}\mathbf{p}_{\mathcal{C}_{k-1}})}{\Delta T} \tag{1} $$

where $\mathcal{C}_{k-1}$ denotes the camera frame at time $k-1$ and ΔT is the elapsed time between time-steps $k-1$ and k. However, since (1) corresponds to a first order numerical derivation of the position ${}^{\mathcal{C}_0}\mathbf{p}_{\mathcal{C}}$ it would be considerably affected by noise. For this reason, instead of (1), we use

$$ {}^{\mathcal{C}_k}\hat{\mathbf{v}}_{\mathcal{C}_k} = \frac{\hat{\mathbf{R}}^{\mathcal{C}_k}_{\mathcal{C}_0}({}^{\mathcal{C}_0}\hat{\mathbf{p}}_{\mathcal{C}_k} - {}^{\mathcal{C}_0}\hat{\mathbf{p}}_{\mathcal{C}_{k-1}})}{\Delta T} \tag{2} $$

[4] `https://github.com/tum-vision/dvo`

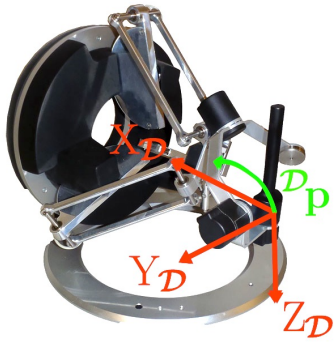

Fig. 5. The haptic device and its frame of reference.

where ${}^{\mathcal{C}_0}\hat{\mathbf{p}}_{\mathcal{C}_k}$, and $\hat{\mathbf{R}}^{\mathcal{C}_0}_{\mathcal{C}_k}$ are the 1€-filtered (Casiez et al., 2012) versions of ${}^{\mathcal{C}_0}\bar{\mathbf{p}}_{\mathcal{C}_k}$, and $\bar{\mathbf{R}}^{\mathcal{C}_0}_{\mathcal{C}_k}$ respectively.

The velocity ${}^{\mathcal{C}}\mathbf{v}_{\mathcal{C}}$ of $O_{\mathcal{C}}$ in $\mathcal{C}$ can be written as

$$ {}^{\mathcal{C}}\mathbf{v}_{\mathcal{C}} = \mathbf{R}^{\mathcal{C}}_{\mathcal{Q}}\,{}^{\mathcal{Q}}\mathbf{v}_{\mathcal{C}} = \mathbf{R}^{\mathcal{C}}_{\mathcal{Q}}({}^{\mathcal{Q}}\mathbf{v}_{\mathcal{Q}} + {}^{\mathcal{Q}}\omega_{\mathcal{Q}} \times {}^{\mathcal{Q}}\mathbf{p}_{\mathcal{C}}). \tag{3} $$

Therefore we compute an estimate of ${}^{\mathcal{Q}}\mathbf{v}_{\mathcal{Q}}$ at time-step k as

$$ {}^{\mathcal{Q}}\hat{\mathbf{v}}_{\mathcal{Q}} = \mathbf{R}^{\mathcal{Q}\mathcal{C}_k}_{\mathcal{C}}\hat{\mathbf{v}}_{\mathcal{C}_k} - {}^{\mathcal{Q}}\hat{\omega}_{\mathcal{Q}} \times {}^{\mathcal{Q}}\mathbf{p}_{\mathcal{C}} \tag{4} $$

Finally, given the estimates $\hat{\phi}_{\mathcal{Q}}$, $\hat{\theta}_{\mathcal{Q}}$, we obtain the sought velocity in the $\mathcal{H}$ frame

$$ {}^{\mathcal{H}}\hat{\mathbf{v}}_{\mathcal{C}} = \hat{\mathbf{R}}^{\mathcal{H}}_{\mathcal{Q}}\,{}^{\mathcal{Q}}\hat{\mathbf{v}}_{\mathcal{Q}} \tag{5} $$

which is then used in the velocity tracker in order to follow the velocity commanded by the operator and the obstacle avoidance module.

3.3 Human operator

The testing of the platform has been carried out on teleoperation experiments. Hence, we have considered the situation in which an operator is required to drive the robot receiving only the feedback from the onboard sensors of the UAV. In particular, the operator receives a visual feedback from the camera.

In addition, the human operator interfaces with the system through an haptic device, the omega.6 [5], shown in Fig. 5. The device provides six degrees of freedom (DOFs), three translational and three rotational, in order to offer complete motion to a 3D rigid body. However, we have limited our system to use only the three translational DOFs. Let be $\mathcal{D} : \{O_{\mathcal{D}}, X_{\mathcal{D}}, Y_{\mathcal{D}}, Z_{\mathcal{D}}\}$ the NED frame of reference whose origin is located in the steady position of the end effector of the haptic device, then we define ${}^{\mathcal{D}}\mathbf{p} = (p_x\, p_y\, p_z)^T$ the configuration of the three translational DOFs of the haptic interface in $\mathcal{D}$. The commanded velocity for the quadrotor, expressed in $\mathcal{H}$, is then computed as:

$$ \tilde{v}_x = k_v p_x \cos(\alpha) \tag{6} $$

$$ \tilde{v}_y = k_v p_x \sin(\alpha) \tag{7} $$

$$ \tilde{v}_z = -k_v p_z \tag{8} $$

where k_v is a positive gain and α is a parameter expressing the direction of the desired forward motion of the quadrotor. For safety reason, we want to force the quadrotor to move only in the direction in which the operator can see through the camera and the obstacles can be perceived, hence α is selected as the yaw angle of the camera in $\mathcal{Q}$:

$$ \alpha = \mathrm{atan2}(\mathrm{r}_{21}, \mathrm{r}_{11}) \tag{9} $$

[5] `http://www.forcedimension.com/`

where $\mathbf{R}^{\mathcal{Q}}_{\mathcal{C}} = [r_{ij}]_{i=1,\dots,3,\, j=1,\dots,3}$.

Notice that the commanded velocities are computed in the frame $\mathcal{H}$ instead of $\mathcal{Q}$ in order to let the command of the operator be independent from the roll and pitch motions that naturally arise when the quadrotor has to accelerate in the horizontal plane.

Finally the commanded yaw rate is obtained as

$$ \tilde{\dot{\psi}} = -k_\psi p_y \tag{10} $$

where k_ψ is also a positive gain.

Given the commands of the human operator, the haptic feedback is generated as the difference between the commands and their actual execution as estimated by the onboard measurement systems, see, e.g., Franchi et al. (2012b) for an analytical expression of the haptic cue.

4. IMPLEMENTATION AND EXPERIMENTS

The main framework in which the platform is developed is TeleKyb (Grabe et al., 2013), a ROS-based project specifically designed for the design of applications on UAVs and oriented to multi-robot execution.

In addition to TeleKyb other general purpose tools as Matlab and Openni have been used in order to accomplish preliminary tasks, as the calibration of the camera and quadrotor frames, and online camera stream acquisition.

At current stage of development not all computation is performed on-board, being the camera directly connected to an external PC which also hosts the execution of dvo and TeleKyb. The connection between the microcontroller and the PC is demanded to two pairs of XBee transmitters/receivers. Nevertheless, we plan to employ an Odroid U2 mini-board to replace the PC, and make the system really able to work in complete autonomy.

We have conducted several experiments in order to evaluate the performance of the proposed algorithms and obtain useful data for their improvement. In all the experiments we used the estimated quantities in the flight controller. Additionally, in order to numerically evaluate the accuracy of the estimation algorithm, we used an external motion capture system as ground truth.

We report here the results of a representative experiment. A video of the experiment can be watched at `antoniofranchi.com/videos/onboard_hapteleop.html`.

The experiment has been performed in a 4 m $\times$ 6 m arena with a cardboard-box obstacle placed approximately at its center.

In Fig. 6 we show the plots of the estimated (blue), ground truth (red) and commanded $\mathbf{v}^*_{des}$ (green) values of the three velocity components expressed in $\mathcal{H}$. Fig. 7 shows the terms of the velocities due to the obstacle avoidance, i.e., $\mathbf{v}_{obs}$. Commanded and measured yaw rate from the onboard gyroscope are shown in Fig. 8. All plots show that the velocity estimate is very similar to the ground truth counterpart. In addition, the quadrotor reproduces quite faithfully the commanded velocity and yaw-rate.

At the beginning the quadrotor is on the ground and at 15 s it takes off. Then the human operator commands the UAV to rotate in the direction of the obstacle and starts driving the UAV toward the central obstacle at constant

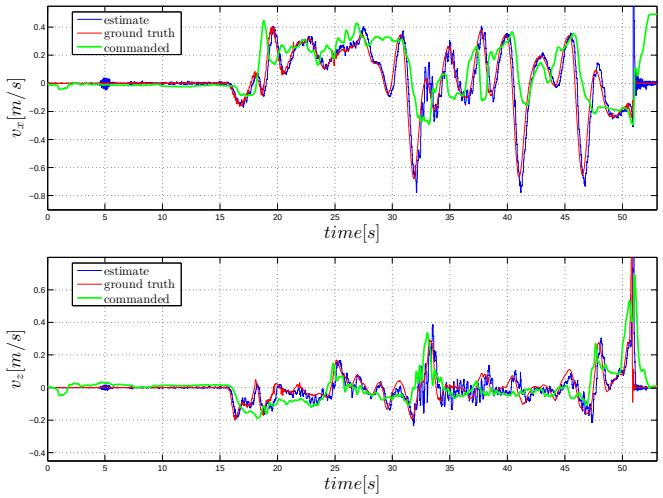

Fig. 6. Comparison between the x and z components of: the estimated velocity (blue plots), the ground truth velocity measured by an external motion capture system (red plots), and the velocity commanded by the human operator $\mathbf{v}^*_{des}$ (green plots). All the velocities are expressed in the horizontal body frame $\mathcal{H}$.

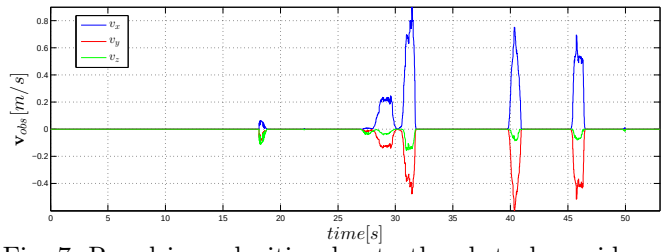

Fig. 7. Repulsive velocities due to the obstacle avoidance on the three axes.

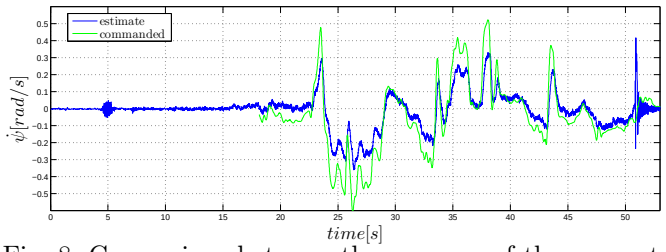

Fig. 8. Comparison between the measure of the yaw rate obtained from the gyroscope (blue plot) and the commanded yaw rate (green plot).

speed (the phase lasts approximately from 20 s to 28 s). In this phase the commanded and actual velocities in the horizontal plane are approximately constant and there is no evident evident error between the two signals.

At time 28 s the UAV comes close enough to the obstacle and the obstacle potential starts rising thus adding to the commanded velocity a repulsive component that lets the actual velocity greatly deviate from the commanded one. Between 28 s and 32 s the operator pushes the UAV against the obstacle two times thus generating two peaks in the actual velocities. In these phase the operator feels a high opposing forces informing him/her about the presence of the obstacle. In the rest of the experiment the operator tries again to drive the quadrotor straight against the obstacle, in particular at times 41 s and 46 s. Also in correspondence of those times, it is possible to recognize big spikes in the real velocities, which also significantly

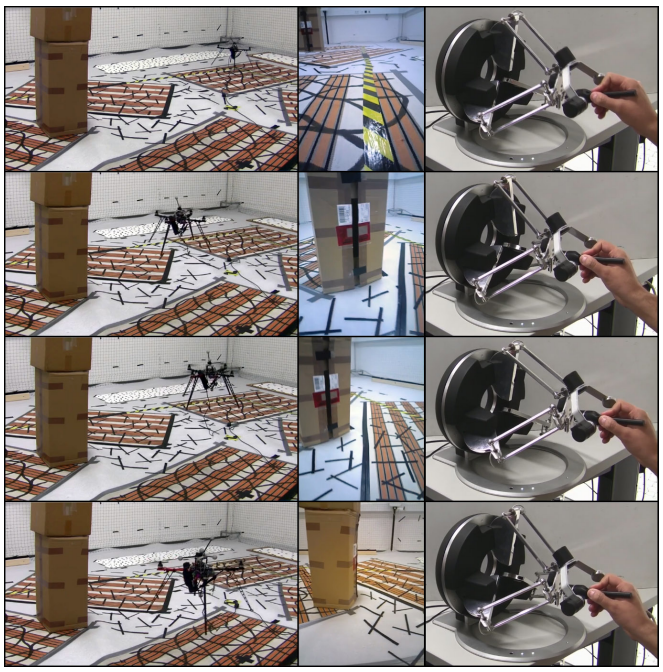

Fig. 9. Snapshots of an experiment. Each row refers to a different time instant. Left column: global views of the environment with the quadrotor on the right and the obstacle box on the left. Middle column: onboard views from the depth-camera sensor. Right column: haptic interface used by the human operator.

differ from commanded velocities due to the velocity correction term added by the obstacle avoidance algorithm of Fig. 7. Finally, the UAV lands at time 51 s.

Fig. 9 reports some significant snapshots of the described experiment, with both global and onboard views, plus the haptic interface operated by the human.

5. DISCUSSION AND CURRENT WORK

In this paper we have presented the development of a semi-autonomous UAV platform that is used for indoor haptic teleoperation control and is able to exploit only onboard sensors, thus being independent from any motion capture system. No assumptions on the environment are needed, such, e.g., the presence of planar surfaces or objects of known sizes.

The ongoing project is currently at a stage in which we are able to drive a quadrotor indoor without the help of external navigation systems and specific environment assumptions. However, not all computation is performed onboard, being the execution of dvo and TeleKyb software demanded to an external PC. The main drawback of the current system configuration is to have an USB cable connecting the onboard depth camera and the external PC, clearly limiting the motion of the quadrotor and disturbing its dynamics. In fact, the microcontroller alone is not able to acquire the output of the depth camera, nor to send it to the base station. Nevertheless, in our goal configuration we plan to connect the RGB-D sensor directly to the Odroid board, hence removing this issue.

Other improvements will consider different filtering strategies for the angular velocities and for the whole state. Once the platform is complete, we plan to employ it to

perform teleoperation experiments over the internet, hence introducing significant delay on the commanded velocities.

REFERENCES

A. Bry, A. Bachrach, and N. Roy. State estimation for aggressive flight in gps-denied environments using onboard sensing. In *2012 IEEE Int. Conf. on Robotics and Automation*, May 2012. doi: 10.1109/ICRA.2012.6225295.

G. Casiez, N. Roussel, and D. Vogel. 1 euro filter: a simple speed-based low-pass filter for noisy input in interactive system. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 2527–2530, Austin, Texas, May 2012.

J. Engel, J. Sturm, and D. Cremers. Camera-based navigation of a low-cost quadrocopter. In *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Oct. 2012.

A. Franchi, C. Masone, V. Grabe, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. Modeling and control of UAV bearing-formations with bilateral high-level steering. *The International Journal of Robotics Research, Special Issue on 3D Exploration, Mapping, and Surveillance*, 31 (12):1504–1525, 2012a.

A. Franchi, C. Secchi, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs. *IEEE Robotics & Automation Magazine, Special Issue on Aerial Robotics and the Quadrotor Platform*, 19(3):57–68, 2012b.

A. Franchi, C. Secchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano. Bilateral teleoperation of groups of mobile robots with time-varying topology. *IEEE Trans. on Robotics*, 28(5):1019–1033, 2012c.

V. Grabe, M. Riedel, H. H. Bülthoff, P. Robuffo Giordano, and A. Franchi. The TeleKyb framework for a modular and extendible ROS-based quadrotor control. In *6th European Conference on Mobile Robots*, Barcelona, Spain, Sep. 2013.

S. Grzonka, G. Grisetti, and W. Burgard. A Fully Autonomous Indoor Quadrotor. *IEEE Trans. on Robotics*, 8(1):90–100, 2 2012.

X. Hou, R. Mahony, and F. S. Schill. Representation of vehicle dynamics in haptic teleoperation of aerial robots. In *2013 IEEE Int. Conf. on Robotics and Automation*, pages 1477–1483, Karlsruhe, Germany, May 2013.

C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for rgb-d cameras. In *2013 IEEE Int. Conf. on Robotics and Automation*, May 2013.

T. M. Lam, H. W. Boschloo, M. Mulder, and M. M. Van Paassen. Artificial force field for haptic feedback in UAV teleoperation. *IEEE Trans. on Systems, Man, & Cybernetics. Part A: Systems & Humans*, 39(6):1316–1330, 2009.

D. J. Lee, A. Franchi, H. I. Son, H. H. Bülthoff, 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*, 18(4):1334–1345, 2013.

R. Mahony, T. Hamel, and J.-M. Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Trans. on Automatic Control*, 53(5):1203–1218, 2008.

P. Martin and E. Salaün. The true role of accelerometer feedback in quadrotor control. In *2010 IEEE Int. Conf. on Robotics and Automation*, pages 1623–1629, Anchorage, AK, May 2010.

A. Y. Mersha, S. Stramigioli, and R. Carloni. Switching-based mapping and control for haptic teleoperation of aerial robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2629–2634, Vilamoura, Portugal, Oct. 2012.

R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset, and A. Erkmen. Search and rescue robotics. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 1151–1173. Springer, 2008.

M. Nieuwenhuisen, D. Droeschel, J. Schneider, D. Holz, T. Läbe, and S. Behnke. State estimation for highly dynamic flying systems using key frame odometry with varying time delays. In *6th European Conference on Mobile Robots*, Barcelona, Spain, Sep. 2013.

S. Omari, M. D. Hua, G. J. J. Ducard, and T. Hamel. Bilateral haptic teleoperation of VTOL UAVs. In *2013 IEEE Int. Conf. on Robotics and Automation*, pages 2385–2391, Karlsruhe, Germany, May 2013.

M. Riedel, A. Franchi, H. H. Bülthoff, P. Robuffo Giordano, and H. I. Son. Experiments on intercontinental haptic control of multiple UAVs. In *12th Int. Conf. on Intelligent Autonomous Systems*, pages 227–238, Jeju Island, Korea, Jun. 2012.

H. Rifa, M. D. Hua, T. Hamel, and P. Morin. Haptic-based bilateral teleoperation of underactuated unmanned aerial vehicles. In *18th IFAC World Congress*, pages 13782–13788, Milano, Italy, Aug. 2011.

D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Fraundorfer, E. B. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. A. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. H. Lee, S. Lynen, L. Meier, M. Pollefeys, A. Renzaglia, R. Siegwart, J. C. Stumpf, P. Tanskanen, C. Troiani, and S. Weiss. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments. *Accepted to IEEE Robotics & Automation Magazine*, 2013.

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.

K. Schmid, F. Ruess, M. Suppa, and D. Burschka. State estimation for highly dynamic flying systems using key frame odometry with varying time delays. In *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Oct. 2012.

S. Shaojie, N. Michael, and V. Kumar. Autonomous multi-floor indoor navigation with a computationally constrained MAV. In *2011 IEEE Int. Conf. on Robotics and Automation*, pages 20–25, Shanghai, China, May 2011.

S. Stramigioli, R. Mahony, and P. Corke. A novel approach to haptic tele-operation of aerial robot vehicles. In *2010 IEEE Int. Conf. on Robotics and Automation*, pages 5302–5308, Anchorage, AK, May 2010.

S. Weiss, D. Scaramuzza, and R. Siegwart. Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011.