The Flying Hand: a Formation of UAVs for Cooperative Aerial Tele-Manipulation

Guido Gioioso[1,2], Antonio Franchi[3], Gionata Salvietti[2], Stefano Scheggi[1], Domenico Prattichizzo[1,2]

Abstract— The flying hand is a robotic hand consisting of a swarm of UAVs able to grasp an object where each UAV contributes to the grasping task with a single contact point at the tooltip. The swarm of robots is teleoperated by a human hand whose fingertip motions are tracked, e.g., using an RGB-D camera. We solve the kinematic dissimilarity of this unique master-slave system using a multi-layered approach that includes: a hand interpreter that translates the fingertip motion in a desired motion for the object to be manipulated; a mapping algorithm that transforms the desired object motions into a suitable set of virtual points deviating from the planned contact points; a compliant force control for the case of quadrotor UAVs that allows to use them as indirect 3D force effectors. Visual feedback is also used as sensory substitution technique to provide a hint on the internal forces exerted on the object. We validate the approach with several human-in-the-loop simulations including the full physical model of the object, contact points and UAVs.

I. INTRODUCTION

Aerial mobile manipulation is a growing area of robotics research that claims to bring in the air the results obtained on the terrain by the mobile manipulation community using UGVs. Exploiting the ubiquity of low cost and easy to fly UAVs, great steps have been made in a relatively short time. Different approaches have been pursued to let a flying robot grasp and manipulate an object. A solution is to equip the vehicles with an arm and a gripper or more sophisticated end-effectors. In [1], for instance, a fixed gripper enabling grasping and object retrieval at high speeds is attached to a quadrotor UAV. A similar goal has been then pursued in [2] with an actuated appendage. In [3] a small–scaled autonomous helicopter is equipped with a robotic arm.

Having on-board a hand-arm system can be efficient in specific tasks, but presents several drawbacks. The whole UAV-arm system results in a higher weight requiring higher energy and decreasing the efficiency and autonomy of the system. This makes such equipped UAVs difficult to utilize for other tasks such as exploration or inspection. Moreover simple gripper cannot accomplish complex manipulation tasks, while robotic hands present a limited workspace and are still difficult to realize and control.

A possible way to overcome all these issues could be to grasp and carry objects using more than one UAV. This is a challenging problem due to the complexity of the task and the limitations of aerial vehicles when compared to grounded robots. In [4], a team of quadrotors are rigidly attached to a payload in order to grasp and transport it. In [5] and in [6] the problem of controlling multiple aerial robots manipulating and transporting a payload via cables is presented. However, the use of cables or others fixed handles is possible only if the load to be carried is accessible and it is possible to fix the cable on it in a specified position.

In this paper we aim at merging the two approaches considering cooperative UAVs grasping an object, where each UAV makes a single contact with the object through a rigidly attached tool. Practically, we can assume each UAV acting as a *finger* of an N-fingered hand and collaborating with the other UAVs to realize a complex and unique hand-system that is able to grasp and move an object. Some preliminary results in this promising direction have been presented in [7] and in [8]. In [7], a dynamic model for a single and double quadrotor UAV manipulating a cart on a track is derived. In [8] a novel second order sliding mode controller is used considering a N quadrotors system grasping and manipulating an object. In this work we consider a control design different from the sliding mode, since this last typically suffers from undesired phenomena, e.g., chattering, that might severely degrade the real-world performances.

The first contribution of this paper is the formalization and the study of the problem of grasping an object by a swarm of N UAVs establishing N contacts with the grasped object, one contact per each UAV, and then cooperatively manipulating it. In this scenario, the UAV team can be used as grasping device only if necessary, keeping its versatility and efficiency. For instance, the robot formation can be guided through a narrow passage varying its shape [9] and then be used to grasp and move an obstacle. Moreover, with this solution, complex grasps can be achieved since contact points can be arbitrarily placed on the object and because UAVs can sense disturbance forces without additional sensors.

The second contribution refers to teleoperation. Teleoperation of multiple aerial vehicles, and in general mobile robots, is a recent promising trend in robotics, see, e.g., [9], [10]. In those works the aerial robots were used mainly as mobile sensors and were not allowed to physically interact with the environment. In the framework presented in this paper, the UAVs physically interact with the grasped object and are teleoperated by a human hand. This poses interesting theoretical issues on how to relate the movements and the forces exerted by the swarm of robots with an extremely different structure such as the human hand [11]. In fact, the free motion of the human hand must be able to control both the motion of the grasped object and the forces used to

[1]Department of Information Engineering and Mathematical Sciences, University of Siena, via Roma 56, 53100 Siena, Italy. {gioioso, scheggi, prattichizzo}@dii.unisi.it

[2]Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy gionata.salvietti@iit.it

[3]Centre National de la Recherche Scientifique (CNRS), Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS), 7 Avenue du Colonel Roche, 31077 Toulouse CEDEX 4, France and Max Planck Institute for Biological Cybernetics, Spemanstr. 38, 72076, Tübingen, Germany. antonio.franchi@laas.fr

establish the grasp.

We propose in this work a layer structured approach. The first layer concerns the tracking of the human hand; in the second layer the features extracted from the human hand are mapped into virtual attraction points; finally the last layer lets the UAVs tooltip implement a force stiffness control that acts in a compliant way with respect to the error between the contact and the attraction points.

The results described in this paper are intended as a guideline for future real implementation where the role of UAV can be played by quadrotors or by other devices. In particular, we believe that this approach will be notably suitable for micro-UAVs, since the reduced dimension and weight allows for better performances in terms of power efficiency [12]. As a first step in this direction we recently validated the ability of the Force Control Layer to work in reality in [13] where the behavior of a single UAV in contact with an object is analyzed.

The paper is organized as it follows. In Sec. II the problem modeling, the objectives and the the structure of the proposed framework are presented. In Sec. III the multi-layer organization of the proposed framework is described in detail. Section IV deals with the human-in-the-loop simulations performed to validate the proposed approach. Finally in Sec. V conclusions and future work are outlined.

II. MODELING, OBJECTIVES AND STRUCTURE OF THE TELEOPERATION FRAMEWORK

Consider a rigid object, denoted by $\mathscr{O}$, and a group of N UAVs, denoted by $\mathscr{U}_1 \ldots \mathscr{U}_N$. The configuration of $\mathscr{O}$ is defined by the position $p_o \in \mathbb{R}^3$ of the center of mass O_b in an inertial frame $\mathscr{W} : \{O_w, \vec{x}_w, \vec{y}_w, \vec{z}_w\}$ and by the rotation matrix $R_o \in SO(3)$ defining the orientation of a moving frame $\mathscr{B} : \{O_b, \vec{x}_b, \vec{y}_b, \vec{z}_b\}$ that is rigidly attached to $\mathscr{O}$.

We assume that a hand-tracking system records in real-time the motion of the fingertips of a human operator (e.g., using the technique described in Sec. III-A). The first goal of our system can be then stated as

Goal 1. To use the hand/fingertip motion to remotely manipulate $\mathscr{O}$ in the sense of changing its location p_o and attitude R_o (i.e., to perform translations and rotations) by exploiting the action of the N UAVs as if they were a *remote flying hand*.

To this aim, each UAV possesses a rigidly attached tool whose tooltip acts as a finger to exert the needed interaction forces with the object. In particular in this framework we consider the N tooltips acting as N fingertips of a flying robotic hand that performs, borrowing the terminology from grasping literature, a *precision grasp* [14] The contact between the tooltip and the object is modeled as a contact point with friction, also referred to as hard finger contact model [15]. According to this model no torsional moments can be exchanged at the contact point. Furthermore, the contact force at the tooltip of $\mathscr{U}_i$ expressed in $\mathscr{B}$ and denoted with $\lambda_i \in \mathbb{R}^3$, must satisfy the friction constraints described by the Coulomb model [15].

Denote by $w \in \mathbb{R}^6$ the resulting wrench, expressed in $\mathscr{B}$, that is applied to $\mathscr{O}$ by $\lambda_1 \ldots \lambda_N$. The relation between w and $\lambda_1 \ldots \lambda_N$, depends on the geometry of the contact points and is given by

$$w = G\lambda \tag{1}$$

where $\lambda = (\lambda_1^T \ldots \lambda_N^T)^T \in \mathbb{R}^{3N}$ is the stacked vector of contact forces and $G \in \mathbb{R}^{6 \times 3N}$ is the grasp matrix [15]. Solving (1) for the contact forces, it is possible to distinguish two contributions. The first, called internal forces [16], does not contribute to the motion of $\mathscr{O}$, i.e., $\lambda_{int} \in \ker G$, and can be obtained through the null-space projector

$$\lambda_{int} = (I - G^\# G)\lambda, \tag{2}$$

where $G^\#$ is the Moore-Penrose pseudoinverse of G and I is the identity matrix. These forces, and in general the nullspace of G, play an important role in maintaining a stable grasp, as detailed in [17].

The second component of λ is defined as

$$\lambda_{ext} = \lambda - \lambda_{int},$$

i.e., the part of the force that produces the motion of $\mathscr{O}$ [18]. It has been proven [19] that the choice of the internal forces λ_{int} and the control of the object motion via λ_{ext} can be considered independently in the control design.

In grasping the optimal choice of the contact points on the object strongly affects the grasp properties. The optimal choice of the contact points is known as grasp planning [16] and responds to many criteria including the task to be performed with the object.

We assume that a preliminary grasp planning phase determines the contact points and internal forces to be applied according to some criteria, e.g., the one presented in [20] to guarantee force closure [15]. The grasp planner returns N contact points, whose positions in $\mathscr{B}$ are denoted by $\bar{p}_1 \ldots \bar{p}_N$ and a set of *nominal internal force*, denoted by $\bar{\lambda} = (\bar{\lambda}_1 \ldots \bar{\lambda}_N)^T$ expressed in $\mathscr{B}$. The nominal internal forces guarantee a stable grasp in the static case and with nominal environment conditions. At the starting time t_0 the object is static and each UAV has approached the object. Each tooltip coincides with the corresponding contact point provided by the grasp planner.

The second goal of our system is then related to the control of the grasping force.

Goal 2. Let the human operator be able to change the intensity of the internal forces, i.e., to choose the actual internal forces in span$\{\bar{\lambda}\}$.

By fulfilling Goal 2 the object can be, for example, grasped or released depending on the operator desire. Notice that the hard finger contact model assumptions is, on purpose, very general since it does not assume that the robot is able to dock to the object at the tooltip. In fact, the possibility of docking assumes the presence of some particular conditions that might not be fulfilled in many real-world scenarios, e.g., the availability of an actuated gripper on the UAV, the presence of a suitable docking surface or handle on the object, etc. Nevertheless, the proposed approach is still valid if some docking mechanism is provided. In this particular case Goal 2 looses its relevance and only Goal 1 must be pursued.
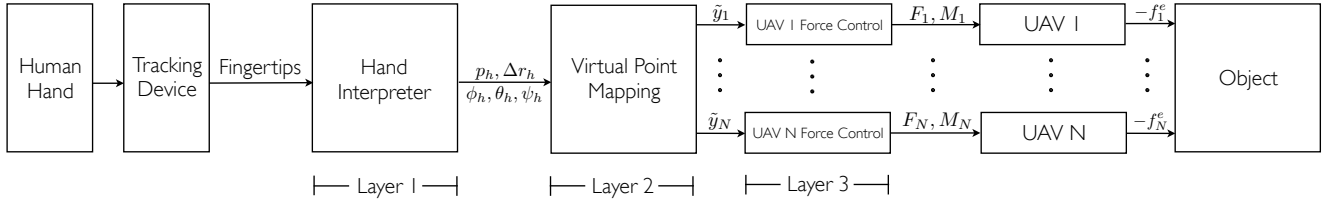
Fig. 1: Schematic representation of the framework. After the approaching phase where the UAVs autonomously get closer to the contact points, the hand motion is captured by a tracking system and translation and rotation of the hand palm and the radial deformation are computed (Layer 1). Virtual attraction points are then computed from the human interpreted commands in order to induce a suitable contact force (Layer 2). Contact forces are exerted by the tooltips on the object using a stiffness control and allow the formation to grasp and manipulate the object (Layer 3). In order to close the loop of the teleoperation, force feedback is visually provided to the human user.

Due to the kinematic dissimilarity between the number of human fingertips and the number of UAV tooltips, the two goals cannot be achieved just mapping the finger action one-to-one to the UAV commands. A suitable 'translation' is therefore needed, which also takes into account possible actuation limits of the UAVs.

In order to overcome all these issues and allow for a transparent and flexible application of the human motion directives to the remote object we designed a framework that is based on a 3-layer approach (refer to Fig. 1 for a block-diagram representation). In the first layer, called *Hand Interpreter* and described in Sec. III-A, the human hand/finger motion is abstracted in order to retrieve high level motion commands, i.e., desired translational velocity, desired rotation rates (roll-rate, pitch-rate, and yaw-rate), and desired grasping force intensity.

In the second layer, called *Virtual Point Mapping* (described in Sec. III-B) we translate these desired actions into virtual 'attraction' points for the UAV tooltips. The role of the virtual attraction points is to indirectly generate a set of forces as in the indirect force control paradigm. In fact, the third layer, called *UAV Force Control*, implements a force stiffness control that is in charge of controlling the UAV actuation in a way that the tooltip behaves as a non-linear spring with a known stiffness map.

Notice that this approach is substantially different from a pure abstraction-layer formation control approach. In fact, the virtual points do not represent in this case a desired trajectory for the UAVs but only some minimum points for the exerted forces, which actually generate the motion of $\mathcal{O}$.

III. COOPERATIVE AERIAL TELEMANIPULATION FRAMEWORK

A. Hand Interpreter

Denote by $\mathcal{H} : \{O_h, \vec{x}_h, \vec{y}_h, \vec{z}_h\}$ a moving frame that is rigidly attached to the tracked hand, and by $\mathcal{H}_0 : \{O_h^0, \vec{x}_h^0, \vec{y}_h^0, \vec{z}_h^0\}$ a fixed frame that coincides with $\mathcal{H}$ at t_0. A possible choice for $\mathcal{H}$ is to set O_h at the centroid of hand, the axis $\vec{x}_h$ passing to a given finger, $\vec{z}_h$ orthogonal to the palm, and $\vec{y}_h$ accordingly. Denote by $p_h \in \mathbb{R}^3$ the position of O_h expressed in $\mathcal{H}_0$, and by $R_h \in SO(3)$ the rotation matrix that expresses the orientation of $\mathcal{H}$ w.r.t. $\mathcal{H}_0$. Furthermore, denote by $(\phi_h, \theta_h, \psi_h)$ the roll-pitch-yaw (RPY) representation of such orientation. Finally, define with r_h the radius of the circle that best fits the detected positions of the fingertips, and with Δr_h the difference between the current value of r_h and its value at t_0.

The *hand parameters*: p_h, $\eta_h = (\phi_h \ \theta_h \ \psi_h)^T$, and Δr_h are used to represent the configuration of the hand, and are provided as input to the Virtual Point Mapping Layer.

Many tracking devices and algorithms can be used to record in real-time the motion of the fingertips, e.g., data gloves, haptic interfaces, and vision-based systems. In our specific implementation we tracked the hand movements using a RGB-D camera. Given the point cloud of the hand, the palm and the fingers can be detected via cluster analysis. Then for each cluster, the centroid and the direction of maximum variance are computed. The fingertips position is estimated from the centroid and direction of the finger point clouds, considering that in our setup the fingertips usually represent the points of the cluster with smaller depth. Then, a RANSAC-based plane fitting is performed on the fingertips. The fingertips and the finger point clouds are projected on such plane and the related bounding circle, and its radius, are evaluated.

At t_0 the hand should be wide open in order to correctly detect the finger clusters and the related fingertips. In the successive frames, if the estimated fingertips are close enough (up to a given threshold) to the old ones, then the fingertip positions are updated. This control allows us to overcome cases in which one or more point clouds are wrongly detected. In this case the point clouds of the fingers are not used in the computation of the bounding circle. Moreover, if one or more fingers are not detected, the pose of the related fingertips is updated accordingly to the overall hand configuration. Figure 2 shows the working principle and result of the proposed hand tracking algorithm used to retrieve the hand parameters.

B. Virtual Point Mapping

The Virtual Point Mapping layer generates the N virtual attraction points on the basis of the human hand parameters provided by the Hand Interpreter (Fig. 3). Denote by $\tilde{y}_i \in \mathbb{R}^3$ the i-th virtual attraction point position expressed in $\mathcal{W}$. At t_0

$$\tilde{y}_i = p_o + R_o \bar{p}_i,$$

i.e., the virtual points coincide with the grasping points.

At the generic time instant t we set

$$\tilde{y}_i(t) = \underbrace{\tilde{p}(t)}_{translation} + \underbrace{\tilde{R}(t)\bar{p}_i}_{rotation} + \underbrace{\alpha_r \Delta r_h \tilde{R}(t)\bar{\lambda}_i}_{grasping}, \quad (3)$$
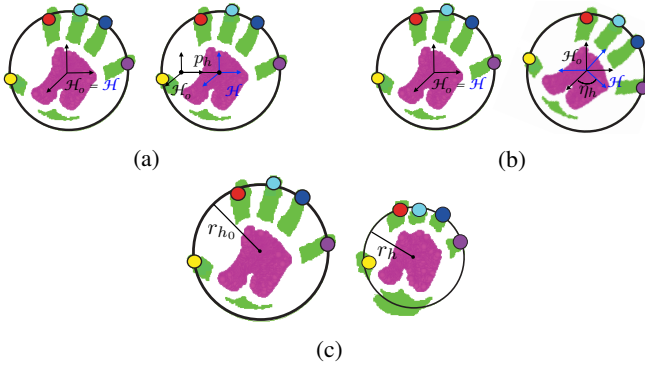
Fig. 2: The hand tracking algorithm retrieves the point clouds related to the palm (magenta) and the fingers (green). Fingertips are then processed in order to retrieve the human commands: (a) translation; (b) rotation and; (c) radius (i.e., grasping force).
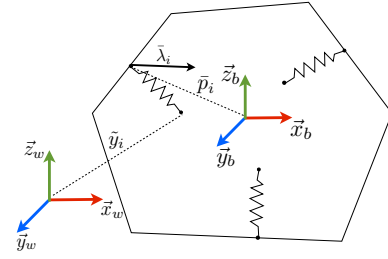

Fig. 3: The Virtual Point Mapping layer. The features extracted from the human hand/fingertip in the first layer are mapped onto attraction points for the UAV tooltips.

$i = 1 \ldots N$, where $\tilde{p}_h \in \mathbb{R}^3$ and $\tilde{R} \in SO(3)$ are generated through the following two independent dynamical systems

$$\dot{\tilde{p}} = \alpha_p p_h, \qquad \tilde{p}(t_0) = p_o(t_0) \tag{4}$$

$$\dot{\tilde{R}} = S(\omega_h)\tilde{R}, \qquad \tilde{R}(t_0) = R_o(t_0) \tag{5}$$

in which

$$\omega_h = \alpha_\omega T(\tilde{\eta})\eta_h$$

being $\tilde{\eta}$ the RPY angles associated to $\tilde{R}$ and $T(\cdot) \in \mathbb{R}^{3\times3}$ the Jacobian matrix mapping RPY angle rates to angular velocity. The positive scalars $\alpha_p, \alpha_\omega, \alpha_r$ represent suitable scale factors.

The following facts should be noted about the algorithm:

- if p_h and η_h are identically zero and $\Delta r_h < 0$ in (3)–(5) then the virtual points at time t are a compressed version of the contact points $\bar{p}_1 \ldots \bar{p}_N$ only in the direction of the internal forces $\bar{\lambda}_i$, the higher the absolute value of Δr_h the higher the compression. Given a certain variation Δr_h, the amount of variation of the internal forces depends on α_r and on the gains of the force controller implemented on the robots;

- if Δr_h is identically zero in (3) then the virtual attraction points at time t are a roto-translated version of the contact points and basically represent a desired position-orientation for $\mathcal{O}$. In particular, p_h is proportional to the translation velocity of the whole points (through (4)) and η_h to the their RPY rotational rates (through (5));

- in the generic case the two kind of actions, compression and roto-translation, sum up. However note that while compression is proportional to the hand radius, roto-translation is, loosely speaking, proportional to the 'integral' of the hand displacement and rotation. This choice is made in order to allow to precisely set the desired internal forces with a *position-position* teleoperation scheme for the hand radius and, at the same time, allow for unlimited space reachability with a *position-rate* scheme for the displacement and rotation.

Now, if we consider $\tilde{y}_1, \ldots, \tilde{y}_N$ to be the attraction points for the UAV tooltips and if the tooltips behave like decoupled linear spring-damper systems, under the action of the UAV Force Control Layer, then the object will be moved

accordingly to the operator directives. This assumes that compression commanded through Δr_h is sufficient to ensure a stable grasp while moving, e.g., by exploiting the coulomb friction. No issues of this sort are instead present if the UAV can dock to $\mathcal{O}$ (see discussion after Goal 2).

Some practical adjustments have to be done to the theoretical algorithm described so far. First of all the scale factors $\alpha_p, \alpha_\omega, \alpha_r$ have to be chosen carefully, in order to avoid that the operator commands motions that are too abrupt for the dynamic capabilities of the UAVs, e.g., taking into account the saturation of their motors. Second of all, being the roto-translation rate commands integrated over time, it might be wise to introduce a saturation that avoids the virtual points to reach regions of the workspace that are not feasible for the mechanical geometry of the UAVs.

C. UAV Force Control

The proposed tele-operation architecture works with different possible UAVs and implementations of the Force Control Layer. In order to show its practicability in a specific case we consider in this section the case of typical underactuated VTOL (Vertical Take-Off and Landing) vehicles, such as, e.g., quadrotors.

1) Quadrotor with a Tool: A quadrotor UAVs is a mechanical system with four propellers placed at the vertices of a square and aligned with the normal of that square, see Fig. 4. We assume that the quadrotor is endowed with a rigidly attached tool whose tip position, expressed in the inertial frame is denoted with y_i. Following the hard finger contact model (see Sec. II) the i-th quadrotor interacts through the tooltip with the environment (i.e., the object $\mathcal{O}$ in our specific case) that generates an interaction force $f_i^e \in \mathbb{R}^3$ expressed in the inertial frame. A complete formal model of this setup can be found in [13].

2) Force Stiffness Control: In [13] we presented a control law designed to allow the tooltip to exert a force on the environment. We derived the analytical expression of the force $-f_i^e$ exerted by the tooltip on the environment for any given $\tilde{y}_i - y_i$. The found relation for typical values of the mechanical and control parameters can be approximated with a linear map (see [13] for more details), thus showing that the tooltip behaves similarly to a linear spring. Therefore the UAV tooltip, under the action of the proposed controller is a compliant system whenever the tooltip is in contact with the object $\mathcal{O}$. This justifies the design, in Sec. III-B, of the virtual points as attraction positions that indirectly generate
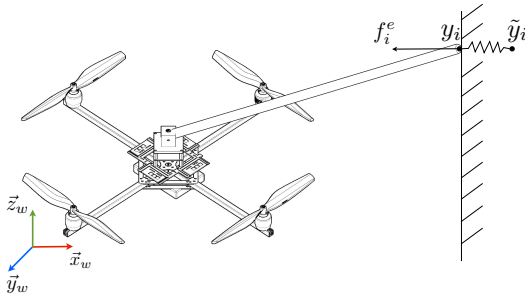
Fig. 4: One of the quadrotor UAVs equipped with the tool, and the main symbols used in the modeling.

the contact forces, implementing the action commanded by the human operator. Note that the same controller can be used also to let y_i track $\tilde{y}_i$ in contact-free motion. More details on the single-UAV control law derivation, its stability analysis, and experimental validation are given in [13].

IV. SIMULATIONS

We conducted several human-in-the-loop (HIL) simulations in order to validate the theoretical framework described so far. In the Hand Interpreter layer we use the Microsoft Kinect RGB-D sensor that sends the depth image to the hand tracker available in the Robotic Operating System (ROS), which runs at an average frame rate of 15 frames per second (fps). This first layer communicates via UDP/IP the finger positions to the Virtual Point Layer that runs in a Matlab program. The UAV Force Control Layer controls several quadrotor UAVs interacting with a object that are both physically simulated within a custom-made environment based on Simulink. Visual information is then provided to the operator by means of the Virtual Reality Toolbox. The whole system runs in real time on a Quad Core i7 3.07Ghz with 16GB of RAM and NVIDiA GForce gtx 180 graphic card.

In order to validate the capabilities of our proposed framework in a realistic scenario, we present here a complete set of plots regarding a prototypical HIL simulation in which the user is asked to remotely move a wooden cubic box between a start and a target location separated by a wall, as depicted in Fig. 5. In the proposed task, the object has to be lifted and rotated during the transport.

The box mass is 1.7 Kg and its edges are 2 m long. The coordinates of the position and the RPY angles describing the starting configuration of the object in the world frame are $(0,0,1)$ m and $(0,0,0)$ deg., respectively. The position coordinates and RPY angles of the target position are $(10,0,1)^T$ m and $(0,0,45)$ deg. The wall is located between the start and target position and its height is 2 m.

Four quadrotors with mass 1 Kg are used to manipulate the object, each one being in charge of pushing one of the four vertical faces of the box. A 4-th order low-pass filter has been employed when mapping the human commands to the quadrotor reference signals. From one side this filter allows to cut off the noise introduced by the RGB-D sensor and the hand tracking algorithm. From the other side such a filter allows to have reference signals that are smooth enough in order to be tracked by a quadrotor UAV.
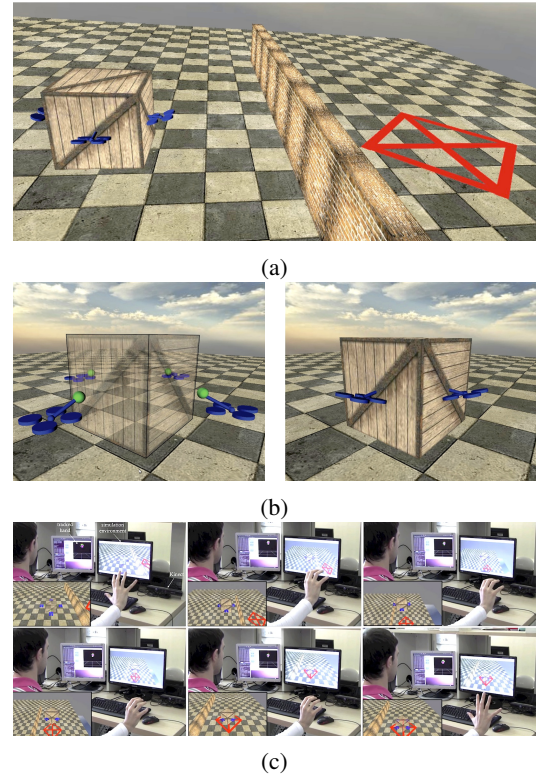

Fig. 5: A human-in-the-loop simulation. (a) The operator task is to bring a wooden cage on the other side of a wall. (b) visual feedback (transparency level) informs the operator about the strength of the internal forces; (c) a human operator performs the task: grasp the box and move it to the red target. The target is rotated and placed beyond the wall.
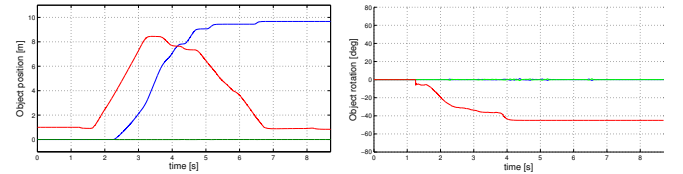

Fig. 6: Object configuration. Left: position (blue $\rightarrow x$, green $\rightarrow y$, red $\rightarrow z$). Right: Euler angles representing its orientation (blue $\rightarrow$ roll, green $\rightarrow$ pitch, red $\rightarrow$ yaw).

In Fig. 6 we show the configuration of the box subject to the teleoperation action. The y coordinate remains constantly at 0 m, while the x coordinate goes monotonically from 0 m to 10 and the z coordinate first goes monotonically from 1 m to about 8 m (when $x \simeq 5$ m) and then goes back to 0 m, showing that the box reaches its highest point in order to overcome the wall and then flies down to the target position. The orientation of the box is represented in terms of RPY angles in the plot at the bottom of Fig. 6. We can see that the pitch and roll are always zero (showing that the attitude of box is kept stable during the aerial transportation) while the yaws goes monotonically from 0 deg to about 45 deg, which is the desired final orientation.

Figure 7a is meant to represent the human translational command (top plot) and the corresponding actual object execution (bottom plot) in terms of linear velocity. A dead zone (delimited by the dashed black horizontal lines) has been introduced in order to facilitate the human task. We

(a) Top: displacement vector o_h of the position of the hand centroid with respect to the rest position (blue $\to x$, green $\to y$, red $\to z$). Horizontal dashed black lines are used to delimit the dead zone. Bottom: velocity of the centroid of the teleoperated box.

(b) Top: angle ω_h representing the rotation of the fingers with respect to the rest position. Horizontal dashed black lines are used to denote a dead zone. Bottom: yaw rate of the teleoperated box.

(c) Top: Hand radius. Bottom: radial forces exerted by the UAVs on the objects (right).
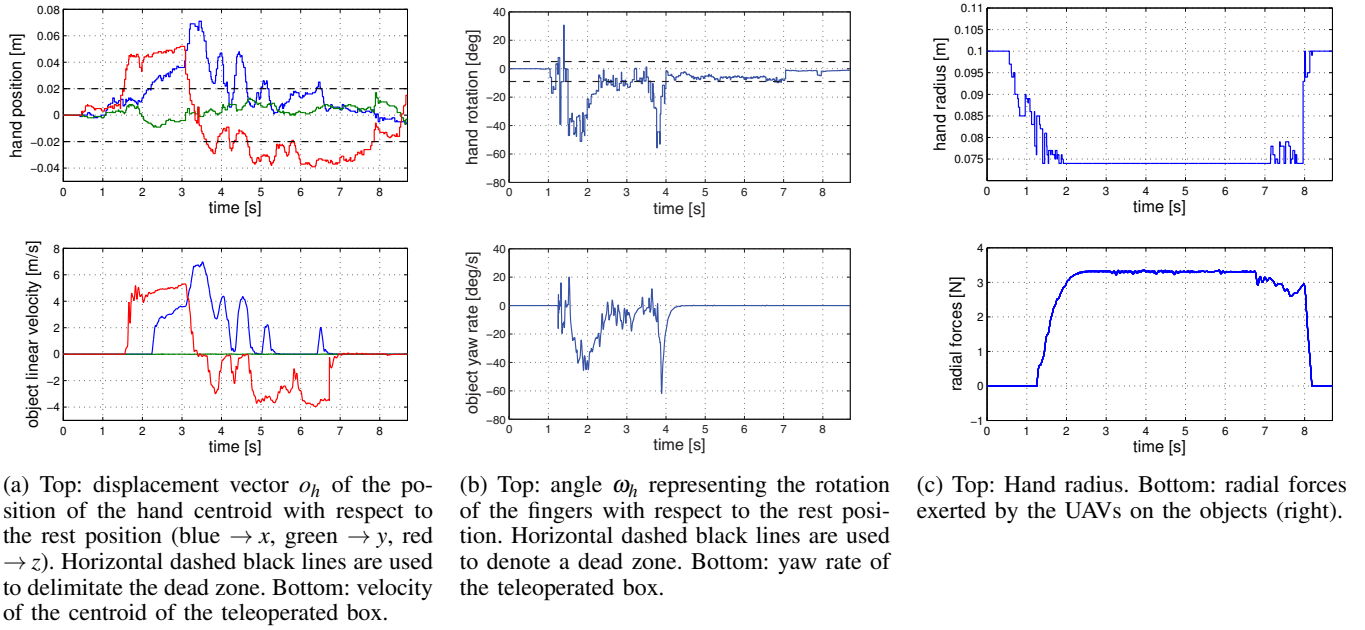
Fig. 7

can see how the human first commands a positive vertical velocity (red line) in order to lift the object. Then a positive horizontal velocity along the x axis is also commanded in order to overcome the wall and reach the target. Finally a negative vertical velocity and a zero horizontal velocity is commanded. The actual velocity of the object (bottom plot) follows the commanded one. The smother behavior of this second one with respect to the commanded velocity is due to the presence of the 4-th order filter described before. Clearly, the integral of this velocity signal is the position of Fig. 6.

As per the rotational commands we artificially limited the rotation to happen only about the vertical axis, being the task only to rotate the yaw of the object. Figure 7b represents the corresponding rotation rate commanded by the human (top) and the actual yaw rate executed by the object under the action of the teleoperated quadrotors. Also in this case we used a dead zone to facilitate the human task. The human first commands a strong negative rotation rate, then stops and then completes the motion with a another negative rotation-rate command. It is possible to appreciate how the actual rotational rate follows the commanded one. Clearly, the integral of this yaw rate signal is the yaw-rate of Fig. 6.

Finally, Fig. 7c represents the radius commanded by the human operator (top) and the average norm of the corresponding internal forces exerted by the quadrotors onto the object (bottom) which are fundamental in order to perform a stable grasp.

In order to show the behaviors of the four quadrotors during the teleoperated grasping, we present in Fig. 8a the tracking errors of the tooltip and in Fig. 8b the quadrotor orientation in terms of RPY angles. At the beginning of the simulation the errors are zero because the quadrotors are not pushing against the box yet. As soon as the push action starts the coordinate of the error which is parallel to the pushing axes of each quadrotor starts to increase in absolute value (i.e., coordinate x for quadrotor 1 and 3, and coordinate y for

2 and 4). When the grasping is performed, the z component of the error becomes positive for each quadrotor, which in turn generates the lifting force. At the same time the pitch of all the quadrotors becomes negative in order to provide the needed grasping and lifting forces (Fig. 8b top). In the plot roll and pitch angles of all the four quadrotors are represented. All the UAVs keep the same roll and pitch angles during the task apart from small variations due to the changes in the operator commands. Given the operator command the quadrotors also start a coordinated yaw rotation (Fig. 8b bottom). The yaw rotation happens because the quadrotor are also asked, as an additional task, to regulate the yaw in order to maintain their tooltip perpendicular to the box faces. In fact, as explained in Sec. III-C, the yaw can be commanded independently from the force that is exerted on the object.

Finally, the coordinated rotation stops and the tracking errors become again zero when the object is released. Notice how the pitch saturates before -20 deg due to a saturation function that as been introduced in order to fulfill real-world requirements (see discussion in Sec. III-B).

The reader is encouraged to watch the video clip attached to the paper where a HIL simulation can be fully appreciated.

V. CONCLUSIONS

An approach to teleoperate a swarm of UAVs, able to grasp and manipulate objects, using the free motion of the human hand has been proposed. Grasping by multi-fingered hands represents the theoretical background of this work since we assumed that each UAV was able to establish a single contact with the object thus simulating a flying hand performing a precision grasp where each quadrotor carries a single finger represented by a tool fixed to the body frame. To the best of our knowledge this is the first time in which a swarm of quadrotors has been telecontrolled by a human hand.

In the HIL simulations we showed the effectiveness of the proposed approach in a realistic virtual environment.
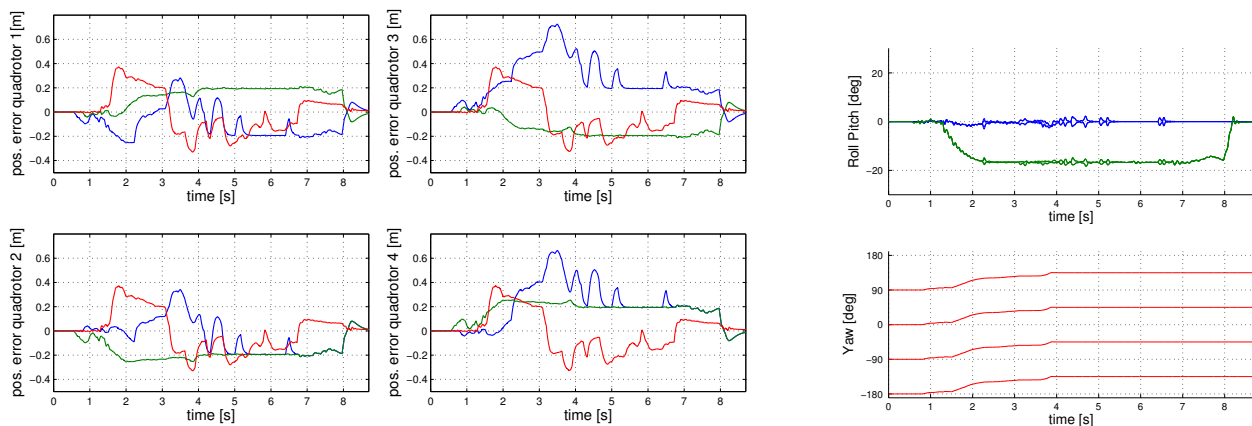
(a) Tracking errors $y_i^d - y_i$ for each quadrotor (blue $\rightarrow x$, green $\rightarrow y$, red $\rightarrow z$).

(b) Rotation angles of the 4 quadrotors. Top: roll (blue) and pitch angles (green). Bottom: yaw angles (red).

Fig. 8

As future work decentralized control techniques will be investigated to improve the stability of the grasp. Real force feedback provided by haptic interfaces will be also introduced to substitute the visual feedback used in the current version of the work. We are also working to improve the ergonomics of the setup in order to make it more comfortable for the human operator and to reduce tiredness effects. In parallel we are also working on the real implementation of a flying hand, with encouraging results [13].

REFERENCES

[1] R. Spica, A. Franchi, G. Oriolo, H. H. Bülthoff, and P. Robuffo Giordano, "Aerial grasping of a moving target with a quadrotor UAV," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, Oct. 2012, pp. 4985–4992.

[2] J. Thomas, J. Polin, K. Sreenath, and V. Kumar, "Avian-Inspired Grasping for Quadrotor Micro UAVs," in *2013 ASME Int. Design Engineering Technical Conf. and Computers and Information in Engineering Conf.*, Portland, OR, Aug. 2013.

[3] K. Kondak, K. Krieger, A. Albu-Schäffer, M. Schwarzbach, M. Laiacker, I. Maza, A. Rodriguez-Castano, and A. Ollero, "Closed-loop behavior of an autonomous helicopter equipped with a robotic arm for aerial manipulation tasks," *International Journal of Advanced Robotic Systems*, vol. 10, pp. 1–9, 2013.

[4] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *10th Int. Symp. on Distributed Autonomous Robotic Systems*, Lausanne, Switzerland, Nov. 2010.

[5] J. Fink, N. Michael, S. Kim, and V. Kumar, "Planning and control for cooperative manipulation and transportation with aerial robots," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 324–334, 2010.

[6] M. Manubens, D. Devaurs, L. Ros, and J. Cortés, "Motion planning for 6-D manipulation with aerial towed-cable systems," in *2013 Robotics: Science and Systems*, Berlin, Germany, May 2013.

[7] 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.

[8] V. Parra-Vega, A. Sanchez, C. Izaguirre, O. Garcia, and F. Ruiz-Sanchez, "Toward aerial grasping and manipulation with multiple UAVs," *Journal of Intelligent & Robotics Systems*, vol. 70, pp. 575–593, 2012.

[9] 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*, vol. 28, no. 5, pp. 1019–1033, 2012.

[10] 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*, vol. 19, no. 3, pp. 57–68, 2012.

[11] G. Gioioso, G. Salvietti, M. Malvezzi, and D. Prattichizzo, "Mapping synergies from human to robotic hands with dissimilar kinematics: an approach in the object domain," *IEEE Trans. on Robotics*, vol. 29, no. 4, pp. 825–837, 2013.

[12] N. Michael and V. Kumar, "Opportunities and challenges with autonomous micro aerial vehicles," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1279–1291, 2012.

[13] G. Gioioso, M. Ryll, D. Prattichizzo, H. H. Bülthoff, 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, May. 2014.

[14] T. Iberall, "The nature of human prehension: Three dextrous hands in one," in *1989 IEEE Int. Conf. on Robotics and Automation*, Scottsdale, AZ, May 1989, pp. 396–401.

[15] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 671–700.

[16] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC, 1994.

[17] A. Bicchi, "Force distribution in multiple whole-limb manipulation," in *1993 IEEE Int. Conf. on Robotics and Automation*, Atlanta, May 1993, pp. 196–201.

[18] G. Salvietti, L. Meli, G. Gioioso, M. Malvezzi, and D. Prattichizzo, "Object-based bilateral telemanipulation between dissimilar kinematic structures," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, Nov 2013.

[19] D. Prattichizzo and A. Bicchi, "Consistent task specification for manipulation systems with general kinematics," *ASME Journal on Dynamic Systems, Measurement, and Control*, vol. 119, pp. 760–767, 1997.

[20] C. Borst, M. Fischer, and G. Hirzinger, "A fast and robust grasp planner for arbitrary 3D objects," in *1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, MI, May 1999, pp. 1890–1896.

[21] T. Lee, M. Leokyand, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *49th IEEE Conf. on Decision and Control*, Atlanta, GA, Dec. 2010, pp. 5420–5425.

[22] D. J. Lee and C. Ha, "Mechanics and control of quadrotors for tool operation," in *2012 ASME Dynamic Systems and Control Conference*, Fort Lauderdale, FL, Oct. 2012.

[23] 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*, vol. 18, no. 4, pp. 1334–1345, 2013.