Shared Control: Balancing Autonomy and Human Assistance with a Group of Quadrotor UAVs

Antonio Franchi, Cristian Secchi, Markus Ryll, Heinrich H. Bülthoff, and Paolo Robuffo Giordano

***Abstract*—In this paper, we present a complete control framework and associated experimental testbed for the *bilateral shared control* of a group of quadrotor UAVs. This control architecture is applicable to any task and allows to integrate: *i)* a decentralized topological motion control (responsible for the mutual interactions in the UAV formation), *ii)* a human assistance module (allowing human intervention, whenever needed, on some aspects of the UAV group behavior), and *iii)* a force-feedback possibility (increasing the telepresence of the human assistants by providing suitable haptic cues informative of the UAV behavior). We will show, as worked-out case studies, how to specialize the topological motion controller to the relevant cases of *constant*, *unconstrained* and *connected* group topologies, and how to let a human operator intervening at the level of single UAVs or of the whole group dynamics. A detailed description of the experimental testbed is also provided with emphasis on the quadrotor UAV hardware and software architectures. Finally, the versatility of the proposed framework is demonstrated by means of experiments with real UAVs. Although quadrotors are used as actual robotic platforms, the proposed framework can be straightforwardly extended to many different kinds of UAVs with similar motion characteristics.**

I. Introduction

As widely shown in the recent literature, robustness and flexibility constitute the main advantages of multiple-robot systems w.r.t. single-robot ones. The use of multiple Unmanned Aerial Vehicles (UAVs) combines these benefits with the agility and pervasiveness of aerial platforms [1], [2]. The degree of autonomy of the multi-UAV system should be tuned according to the specificities of the situation under considerations. For regular missions, fully-autonomous UAV systems are often appropriate, but in general the use of *semi-autonomous* groups of UAVs, *supervised* or *partially-controlled* by one or more human operators, is the only viable solution in order to deal with the complexity and unpredictability of real-world scenarios as in, e.g., the case of search and rescue missions or exploration of large/cluttered environments [3]. In addition to that, the human presence is also often mandatory for taking the responsibility of critical decisions in highly risky situations [4].

In this paper, we describe a unified framework that allows to (i) let the group of UAVs autonomously control its topology in a safe and stable manner, and to (ii) suitably incorporate some skilled human operators in the control loop. This way, the human's superior cognitive capabilities and precise manual skills can be exploited as a valid support of the typical autonomy of a group of UAVs. In fact, drawing from the well-established field of *bilateral teleoperation* [5], our framework includes the possibility of providing the operators with haptic cues informative of the UAV-group and environmental state, as in [6] for the single-UAV case. This will typically increase the humans' situational awareness, execution performance, and ability to take the correct decisions as often demonstrated in bilateral teleoperation systems, see, e.g., [7] and references therein.

As for the UAV group behavior, rather than focusing on a particular task to be executed (e.g., coverage, mapping, or surveillance), we address the general aspects typically shared in any operational scenario, that is, UAV sensing/planning/motion-control, collective behavior during navigation, and human interfacing. The term 'shared control' both refers to sharing the autonomy between humans and UAVs, and to sharing the competences between two or more human operators assisting the group of UAVs from different perspectives.

The feasibility of our framework is demonstrated by illustrating several experiments run in our testbed, which includes a group of quadrotors and two haptic interfaces integrated in a flexible software framework. The explicit integrated possibility of both autonomous quadrotor flight behavior and bilateral interaction between human operators and a UAV-group can be considered as a novel feature compared to existing (and successful) testbeds for multiple-quadrotors [8], [9], [10].

II. UAV Model and Overall Architecture

We consider a group of N UAVs modeled as rigid bodies in $\mathbb{R}^3$. The *configuration* of the i-th UAV is represented by its *position* $\boldsymbol{p}_{\mathcal{B}_i} \in \mathbb{R}^3$ and *rotation matrix* $R_{\mathcal{B}_i} \in SO(3)$ w.r.t. a common inertial frame. The rotation may be also described by the usual *yaw* $\psi_{\mathcal{B}_i}$, *pitch* $\theta_{\mathcal{B}_i}$ and *roll* $\phi_{\mathcal{B}_i}$ angles.

With a special focus on the quadrotor platform, we assume that the UAV is only able to track a smooth reference trajectory $(\boldsymbol{p}_i(t),\, \psi_i(t))$ in the 4-dimensional space $\mathbb{R}^3 \times \mathbb{S}^1$. This is the case, for example, for helicopter and quadrotor UAVs [11], as well as for any other UAV whose position and yaw angle $\left(\boldsymbol{p}_{\mathcal{B}_i},\, \psi_{\mathcal{B}_i}\right)$ are flat outputs [12], i.e, algebraically defining, with their derivatives, the state and control inputs of the UAV.

A. Architecture of the Bilateral Shared Control

Figure 1 illustrates the proposed system architecture with some details on the single-UAV module. One or more human assistants are in charge of defining the current task for the group of UAVs by means of suitable *task interfaces* (e.g., a touch user interface) [13]. A task is a long/medium term activity that the group of UAVs is asked to carry on autonomously, e.g., covering or exploring an area, navigating between via points, surveilling a perimeter, or cooperatively transporting a

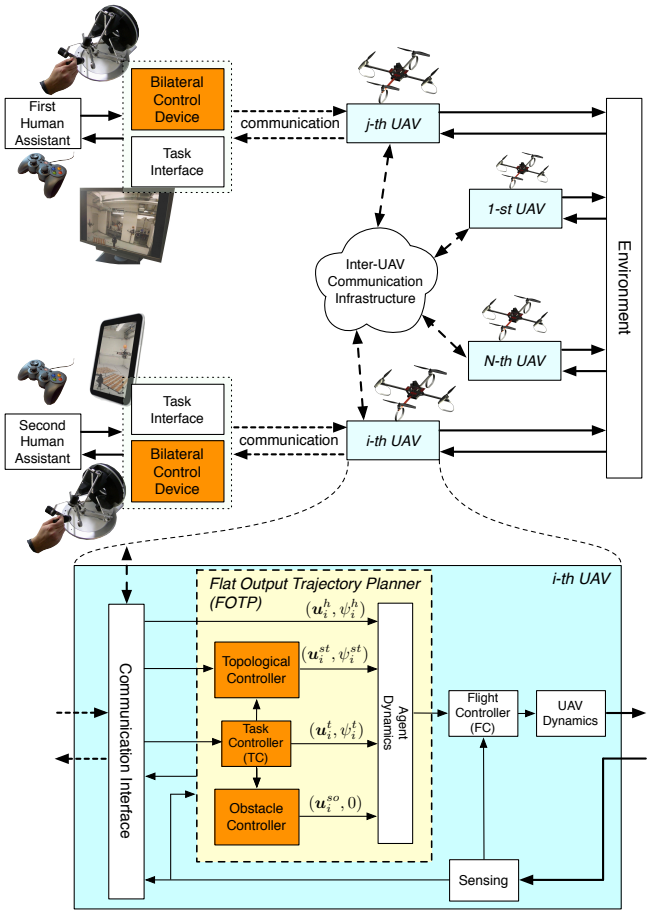

Fig. 1: Overall system architecture as seen from the point of view of the generic i-th UAV. The blocks in charge of the supportive features are: the bilateral control device, the topological controller, the obstacle controller, and the agent dynamics.

load. Our goal is to propose a framework that complements a particular task algorithm with some general supportive features of common utility in any supervised/shared operation, namely: *obstacle avoidance*, *inter-agent behavior*, *human-assistance*, and *human telepresence*.

A core component is the *flat-output trajectory planner* (FOTP) providing the reference flat outputs $(\boldsymbol{p}_i, \psi_i)$, and their derivatives, to the *flight controller* (FC). The flight controller of each UAV acts on the UAV physical control inputs (e.g., the propeller speeds) in order to let the UAV outputs $(\boldsymbol{p}_{\mathcal{B}_i}, \psi_{\mathcal{B}_i})$ track the desired ones $(\boldsymbol{p}_i, \psi_i)$. The FOTP of the i-th UAV coordinates with the FOTPs of the other UAVs by using a communication interface. This can either be the same used to communicate with the human assistants or a dedicated (local) one. The FOTP is designed so as to generate the quantities $(\boldsymbol{p}_i(t), \psi_i(t))$ as the time evolution of two virtual systems (henceforth unified under the name "*agent*"): one system for the desired yaw ψ_i (the *yaw-agent*), and one system for the desired position $\boldsymbol{p}_i$ (the *position-agent*).

In this paper we only consider kinematic yaw-agents, as this is in practice an acceptable assumption for many quadrotor-like UAVs. This then results in

$$\dot{\psi}_i = w_i, \tag{1}$$

where $w_i \in \mathbb{R}$ is the yaw-rate input. On the other hand, we consider to steer the position-agent either at the kinematic (first-order) level, i.e., by commanding a linear velocity $\boldsymbol{u}_i \in \mathbb{R}^3$:

$$\dot{\boldsymbol{p}}_i = \boldsymbol{u}_i, \tag{2}$$

or at the dynamic (second-order) level, i.e., interpreting the command $\boldsymbol{u}_i \in \mathbb{R}^3$ as a force:

$$\dot{\boldsymbol{p}}_i = \boldsymbol{v}_i \tag{3}$$

$$M_i \dot{\boldsymbol{v}}_i = \boldsymbol{u}_i - B_i \boldsymbol{v}_i. \tag{4}$$

Here, $\boldsymbol{v}_i \in \mathbb{R}^3$ and $M_i \in \mathbb{R}^{3\times 3}$ are the velocity and the (symmetric positive definite) inertia matrix of agent i, respectively, and $B_i \in \mathbb{R}^{3\times 3}$ is a positive definite matrix representing an artificial damping added to asymptotically stabilize the behavior of the agent and also take into account typical physical phenomena such as wind/atmosphere drag. The meaning of $\boldsymbol{u}_i$ (either velocity or force input) will be clear from the context.

Modeling UAVs as kinematic agents is a common assumption in the multi-robot literature (e.g., similar assumptions have been made in [1], [2]). Due to their higher complexity, dynamic agents are less commonly adopted. Nevertheless dynamic agents provide a better approximation of the actual UAV dynamics, and therefore are more appropriate whenever the dynamical properties of the UAVs are more 'stressed', as in, e.g., the case of intermittent interactions (see, e.g., the variable topology cases in this paper). The interested reader can also find in [14] a comparison of stability and performance issues for a network of kinematic and dynamic agents.

The inputs $(\boldsymbol{u}_i, \psi_i)$ will depend on the contribution of 4 terms: $(\boldsymbol{u}_i^t, w_i^t)$, $(\boldsymbol{u}_i^{so}, 0)$, $(\boldsymbol{u}_i^{st}, w_i^{st})$, and $(\boldsymbol{u}_i^h, w_i^h)$. These terms are generated by 4 individual subsystems whose meaning and purpose is explained in the following sections, see also Fig. 1 for a visual reference.

1) Task controller: By means of the *task interface*, a human assistant is given the possibility to select an algorithm, generically referred to as *task controller* (TC), for executing a particular task. The internal description of specific TCs is out of the scope of this paper. Any TC, however, will eventually generate an input $(\boldsymbol{u}_i^t, w_i^t)$ for the agent based on the sensorial feedback and inter-UAV communication. Apart from being responsible for the flat-output trajectory generation, the TC may also be in control of additional specific task requirements such as controlling the end-effector of a robotic arm attached to the UAV, or regulating the tension of a rope used for transportation. Finally, the TC is in charge of discriminating which part of the environment should be considered as obstacle by the obstacle controller (e.g., walls), and which part should be considered as safe (e.g., an object to be grasped or pushed). At the same time, the TC can also communicate the desired inter-UAV behavior (e.g., desired formation) to the topological controller, see the following points.

2) Obstacle controller: The obstacle controller generates an input $(\boldsymbol{u}_i^{so}, 0)$ as the gradient of an obstacle avoidance artificial potential to ensure a collision-free navigation w.r.t. the obstacles identified by the task controller. We do not consider an

obstacle avoidance action on the yaw-rate input, as we assume that it is always acceptable to virtually augment the UAV size up to a yaw-symmetric shape (e.g., an ellipsoid).

3) Topological controller: Formation control, requiring that the UAVs maintain some preferred inter-distances or mutual-poses, is a basic feature needed in almost all the multi-aerial tasks. The formation constraints can be more strict or more relaxed depending on the task. For example, in case of a simple coverage, the constraints can be quite relaxed, while in the case of precise sensor fusion (e.g., aerial measurements, interferometry), a particular fixed formation must be kept with high accuracy. The *topological controller* is meant to implement the desired task formation specifications by generating the inputs $(\boldsymbol{u}_i^{st}, w_i^{st})$. Section III provides a description of this controller for some relevant scenarios.

4) Human assistance: In any complex situation, human assistance cannot be limited to a generic supervision as in the case of task assignment. In the real world, complex tasks are made of a collection of activities, and some of them, usually the more challenging ones, need a direct human control. To this aim, the command $(\boldsymbol{u}_i^h, w_i^h)$, generated by the *bilateral control device*, allows the human to finely act on partial aspects of the task, e.g., by steering the centroid of the formation (global intervention) or by precisely maneuvering only one particular UAV (local intervention). This aspect is detailed in Sec. IV.

B. Human telepresence

The nature of the task interface and of the bilateral control device is meant to provide a suitable feedback to the human operator(s) to increase his(their) telepresence. In particular, the task interface can also be used to interactively monitor the state of the task, e.g., by providing augmented video streams from the remote site. In fact, in order to obtain an acceptable performance in the direct assistance case, the human operator needs to be constantly informed about the UAV/environment state with a good perceptive quality. Because of this, in our framework we decided to adopt the typical paradigm of visual/force feedback for human-robot shared control tasks, a paradigm known in the literature also as bilateral control. Section IV-B is devoted to the description of this component.

III. Topological Controller

The topological controller generates an input $(\boldsymbol{u}_i^{st}, w_i^{st})$ in order to implement a desired mutual interaction behavior, e.g., to reach a prescribed set of precise mutual configurations or to approximately keep a given separation distance among the UAVs. The collection of all the mutual interactions can be described by an interaction-topology graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertexes $\mathcal{V}$ represent the UAVs and the weighted edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ represent the intensity of the interaction between two UAVs. A non-zero weight models the presence of interaction while a zero weight means no interaction and is equivalent to the absence of that edge. We consider in detail three possible classes of interaction graphs (see Fig. 2): *constant topology*, when the task requires that the interaction pairs are always the same; *unconstrained topology*, when the topology can freely change over time, allowing the group to even disconnect into several sub-groups; *connected topology*, when the interaction graph can still freely change but with the constraint of remaining connected at all times, i.e., to ensure the group cohesion.

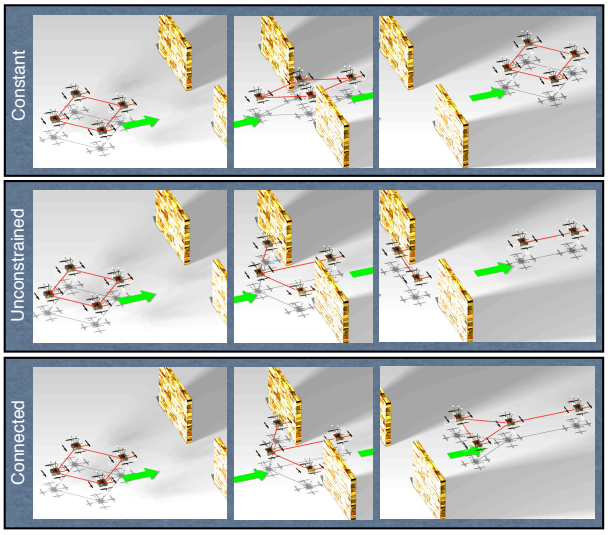

Fig. 2: The three topological behaviors: in the first sequence, the interaction graph remains always constant regardless of the interaction with the environment; in the second sequence, the graph is unconstrained, thereby changing due to the environmental interaction and eventually becoming disconnected; in the third and last sequence, the graph still changes but under the constraint of remaining connected.

A. Constant Topology

The need for constant-topology (i.e., fixed) UAV-formations naturally arises from the specifications of many UAV applications, e.g., for interferometry, transportation [2], for guaranteeing inter-UAV visibility or environmental coverage [1], [3]. The desired formation may be an output of the task controller, or directly specified by the human operators.

Assuming an interaction-topology graph is chosen, a formation is then commonly described by assigning the desired relative behavior between every UAV pair $(i, j) \in \mathcal{E}$. In these notes we consider two typical cases: assigning either the relative bearings $\bar{\boldsymbol{\beta}}_{ij}$ or the inter-distances $\bar{\delta}_{ij}$ $\forall (i, j) \in \mathcal{E}$.

A formation controller only constraining the relative bearings can be implemented by relying on sole relative bearings as measurements, see [15]. This is an interesting feature since relative bearings can be obtained using onboard monocular cameras, that is, light-weight, low-energy, and cheap sensors providing spatial information on the neighbors. Furthermore, when constraining the relative bearings, the UAV formation still possesses 5 DOFs, namely, translation of the centroid, synchronized rotation about the vertical axis, and expansion [15]. These DOFs can then be used to give motion commands to the whole formation.

On the contrary, the regulation of the relative inter-distances cannot be achieved using only inter-distances, but one also needs knowledge of the relative bearings, see [16]. In this case, the formation can be shown to still possess 6 DOFs: translation of the centroid and rotation about any axis in space.

As for formation-control, the typical approaches are based on artificial potentials that generate $(\boldsymbol{u}_i^{st}, w_i^{st})$ as the gradients of energy-like functions having a minimum at the desired inter-agent value (see [16] for one of these cases). However, the drawback of artificial potential approaches is the presence of local minima. In this regard, the work in [15] presents an interesting different strategy not based on artificial potentials and almost globally convergent.

B. Unconstrained Topology

Some tasks require loose inter-UAV couplings, for example the possibility of creating and destroying pairwise interactions (splits and joins) at any time. Typical applications in this context are: (1) navigation in cluttered environments, where an unconstrained interaction topology is much more adaptable than a fixed one; (2) multi-robot exploration, where UAVs frequently need to divide or gather themselves into smaller or larger groups.

In order to model this flexibility, [17], [18] consider the presence of a switching signal for every pair of UAVs $\sigma_{ij}(t) \in \{0, 1\}$ meant to represent the status of the interaction among agents i and j (with $\sigma_{ij} = 1$ indicating presence of the interaction, and $\sigma_{ij} = 0$ otherwise). The time behavior of $\sigma_{ij}(t)$ can model the effect of limited sensing capabilities of the UAVs (e.g., maximum sensing/communication range, occlusions of the line-of-sight), but can also be triggered by the task controller to account for any additional task (e.g., in order to split or join different subgroups).

In the unconstrained topology case, the topological controller should only ensure some form of loose aggregation. Therefore the control term $\boldsymbol{u}_i^{st}$ is designed as the sum of the local interactions over all the *neighbor* UAVs, i.e., only those j-th UAVs s.t. $\sigma_{ij}(t) = 1$. Each interaction force mimics the nonlinear spring behavior depicted in the potential plot of Fig. 3, i.e., a repulsive action if $d_{ij} < d_0$, an attractive action if $d_0 < d_{ij} \leq D$, and a null action if $d_{ij} > D$, where $d_0 < D$ is a neutral inter-distance between the agents.

C. Connected Topology

As a variation of the previous case, we also considered the possibility of allowing for time-varying topologies but under the constraint of maintaining *connectivity* of the graph $\mathcal{G}$ despite the creation or disconnection of individual links. In fact, while flexibility of the formation topology is a desirable feature, connectedness of the underlying graph is often a prerequisite for implementing distributed control/sensing algorithms.

The work in [19] shows a topological controller able to account for connectivity maintenance by exploiting a decentralized estimation $\hat{\lambda}_2$ of λ_2, the second smallest eigenvalue of the Laplacian matrix L associated to the graph $\mathcal{G}$. In fact, it is well-known that a graph $\mathcal{G}$ results connected if and only if $\lambda_2 > 0$. Exploiting the decentralized estimate $\hat{\lambda}_2$, in [19] it is shown how to implement a decentralized gradient-based controller that allows for a time-varying topology due to loss/gain of visibility or maximum range exceedance between agents while ensuring connectivity maintenance at all times.

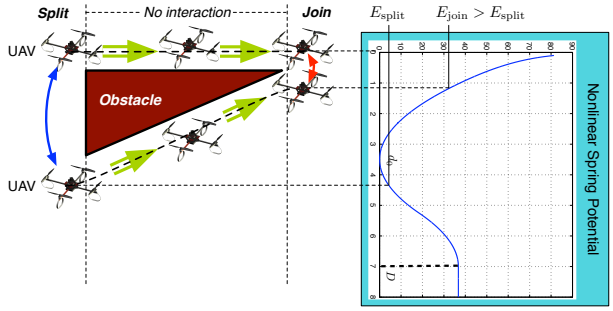

Fig. 3: Right: plot of the nonlinear spring potential modeling the agent interaction. Left: when the UAVs split, the energy E_{split} is stored in the spring, while when they join the energy $E_{\text{join}} > E_{\text{split}}$ is needed to implement the new desired coupling. In this case, without proper strategies, an amount $E_{\text{join}} - E_{\text{split}} > 0$ of energy would be introduced into the system, thus violating passivity.

As an additional feature, this same controller is also proven to implicitly guarantee inter-robot and obstacle collision avoidance: this is achieved by suitably shaping the weights on the edges of neighboring robots as functions of the inter-robot and robot-obstacle relative distances.

D. Ensuring a Stable Behavior of the Agent Group

Guaranteeing *passivity* of the agent group is a sufficient condition for guaranteeing a stable behavior in free motion and when interacting with unknown (passive) environments. In both the cases of unconstrained and connected topologies, the controlled dynamic agents in (3–4) are equivalent to floating masses with damping, so that their interconnection can be considered as a mechanical elastic element. Thus, the agents intuitively behave in a stable way, being essentially a network of springs and (damped) masses. This can be formally proven by showing that the system is passive, namely that the energy exchanged with the external world (i.e., humans and environment) is either stored in the form of (virtual) kinetic or potential energy or dissipated, and that no destabilizing regenerative effects are present.

However, the need for flexibility and connectivity maintenance of the approaches reported in Sec. III-B and in Sec III-C can possibly threaten passivity. In fact, both split and join events, as illustrated in Fig. 3, and the use of an estimation of the connectivity eigenvalue $\hat{\lambda}_2$ in place of the real value λ_2, can create extra energy in the agent group, see [18], [19]. On the other hand, passivity is preserved if and only if the amount of energy dissipated by the agent group is higher than the amount of energy produced. Thus, the amount of dissipated energy is a good measure of the current passivity margin.

In order to understand if an energy producing action can be implemented without violating the overall passivity, i.e., whether it is within the margin, the dynamics of each agent is augmented as follows:

$$\begin{cases} \dot{\boldsymbol{p}}_i = \boldsymbol{v}_i \\ M_i \dot{\boldsymbol{v}}_i = \boldsymbol{u}_i - B_i \boldsymbol{v}_i \\ \dot{t}_i = \frac{1}{t_i}(\boldsymbol{v}_i^T B_i \boldsymbol{v}_i) + \tau_i \end{cases} \tag{5}$$

where $t_i \in \mathbb{R}$ is the state of an energy storing element, called *tank*, characterized by the energy function $T_i(t_i) = \frac{1}{2} t_i^2$. It is

easy to see that $\dot{T}_i = (\boldsymbol{v}_i^T B_i \boldsymbol{v}_i) + \tau_i^T t_i$. Thus all the energy dissipated by the UAV is stored in the tank, and one can still inject/extract energy from the tank using the control input τ_i. For each passivity threatening action, every UAV computes the amount of energy E_i that would be produced by its actual implementation. If $E_i < 0$, the action is dissipative and the agent refills its tank by suitably acting on τ_i in order to inject back the amount $-E_i$. On the other hand, if $E_i > 0$, the agent can implement the action only if $T_i(t_i) > E_i$. If this is the case, the action is implemented and τ_i is exploited to extract E_i from the tank. If $T_i(t_i) < E_i$, the agent can still increase the amount of energy in its tank (e.g., by artificially increasing its damping) until the action can be implemented. Using this strategy, flexibility, connectivity and passivity can easily and elegantly coexists. A more formal and detailed illustration can be found in [18] and [19].

IV. Human Assistance and Telepresence

We consider M (usually $\ll N$) bilateral devices (i.e., with force-feedback capabilities) as human-robot interfaces for allowing a human operator to intervene on some aspects of the UAV motion while receiving suitable haptic cues. The devices are modeled as generic mechanical systems

$$\mathcal{M}_i(\boldsymbol{x}_i)\ddot{\boldsymbol{x}}_i + \mathcal{C}_i(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i)\dot{\boldsymbol{x}}_i = \boldsymbol{\tau}_i^c + \boldsymbol{\tau}_i^h \quad i = 1 \dots M \tag{6}$$

where $\boldsymbol{x}_i \in \mathbb{R}^{d_i}$ is the configuration vector of the i-th-interface, $\mathcal{M}_i(\boldsymbol{x}_i) \in \mathbb{R}^{d_i \times d_i}$ its positive-definite and symmetric inertia matrix, $\mathcal{C}_i(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i) \in \mathbb{R}^{d_i \times d_i}$ represents Coriolis and centrifugal terms, and $(\boldsymbol{\tau}_i^c, \boldsymbol{\tau}_i^h) \in \mathbb{R}^{d_i} \times \mathbb{R}^{d_i}$ are the control/human forces acting on the device, respectively. As usually done, we also assume that gravity effects are locally compensated. In the following we describe two different ways for interfacing a human operator with the UAV group dynamics, namely, a *local* and a *global* intervention modality.

A. Operator Intervention

1) Local intervention: In the local intervention case, each haptic interface influences the motion of a single UAV. A straightforward application of this case is the leader-follower modality: a group of follower UAVs can be guided through an environment by a leader UAV which, in turn, is controlled by a human operator. Another possibility is to exploit this modality to let a skilled operator temporarily help a UAV to solve conflicting situations. A final possibility (multi-leader/multi-follower) is to cooperatively transport an object with a team of UAVs of which only a small subset is driven by human operators.

We assume that $d_i = 3$ for every $i = 1 \dots M$ and consider an injective map $a : \{1, \dots, M\} \to \{1, \dots, N\}$ from the set of force-feedback devices to the set of UAVs. The position of the i-th haptic device will be treated as a (scaled) velocity reference for the $a(i)$-th position-agent. In the case of a kinematic position-agent, this results in either $\boldsymbol{u}_{a(i)}^h = \lambda_i \boldsymbol{x}_i$, or $\boldsymbol{u}_j^h = \boldsymbol{0}$ for all the j-th position-agents not in $Im(a)$. In the case of a dynamic position-agent, the local intervention results in the following proportional controller

$$\boldsymbol{u}_{a(i)}^h = B_h(\lambda_i \boldsymbol{x}_i - \boldsymbol{v}_{a(i)}) \tag{7}$$

where $B_h \in \mathbb{R}^{3\times3}$ is a positive definite damping matrix. Similarly to before, we also have $\boldsymbol{u}_j^h = \boldsymbol{0}\ \forall j \notin Im(a)$.

In the case, not considered here, of $d_i = 4$, one could extend the very same concept to also control the yaw of the UAVs with the additional DOF of the interface.

2) Global intervention: Global intervention allows the operator to control some generalized velocities of the whole group of UAVs by acting on the configuration of the available haptic interfaces. Among the many possible choices we consider here the following cases: cohesive motion of the whole group, synchronized rotation about any axis passing through the k-th UAV, and the expansion/contraction of the whole formation. Denote with $\boldsymbol{x}_t, \boldsymbol{x}_\omega \in \mathbb{R}^3$, and $x_s \in \mathbb{R}$, the configurations of the devices in control of the desired group velocity, angular velocity, and expansion rate, respectively. In case of kinematic position-agents, the desired global intervention action is implemented by setting

$$\boldsymbol{u}_i^h = \lambda_t J_{ti} \boldsymbol{x}_t + \lambda_\omega J_{\omega i} \boldsymbol{x}_\omega + \lambda_s J_{si} x_s \quad \forall i = 1 \dots N \tag{8}$$

where J_{ti}, $J_{\omega i}$, J_{si} are the i-th component of the map between the desired global velocities (generating translation, rotation, and dilation, respectively) and the velocity of the i-th agent, and $\lambda_t > 0$, $\lambda_\omega > 0$, and $\lambda_s > 0$ are suitable scaling factors from the haptic device configurations $(\boldsymbol{x}_t, \boldsymbol{x}_\omega, x_s)$ to the desired global velocities.

If a constant-topology controller is also acting on the UAVs, the global velocities should be orthogonal to the inter-robot constraints enforced by the topological controller in order to preserve the current inter-distances and/or relative bearings. The relative bearings are preserved when all the UAVs translate, dilate, or cohesively rotate around the vertical axis, i.e., when $\boldsymbol{x}_\omega = \begin{pmatrix} 0 & 0 & z_\omega \end{pmatrix}$ and $w_i^h = z_\omega$, with $z_\omega \in \mathbb{R}$. The relative inter-distances are preserved if the group does not dilate, i.e., when $x_s = 0$.

Global intervention in the dynamic case is implemented by using an approach similar to (7) in order to track the desired agent velocity given by (8).

B. Telepresence

The force-feedback on the bilateral devices is designed in order to provide haptic cues informative of how well the *real* UAVs are executing the desired human commands, and to feel the disturbances to which the UAVs are subject to, like turbulences and wind gusts. Recalling Sec. II, we let $\dot{\boldsymbol{p}}_{\mathcal{B}_i} \in \mathbb{R}^3$ be the *body-frame* velocity vector of the i-th UAV, and $\dot{\psi}_{\mathcal{B}_i} \in \mathbb{R}$ its yaw rate. We stress, again, that these represent real (measured) UAV quantities and not the reference (virtual) velocities of the agent tracked by the UAVs.

In case of local intervention, we consider the mismatch between the velocity commanded through the i-th interface and its actual execution by the $a(i)$-th UAV. Therefore we set $\boldsymbol{\tau}_i^c = -B_h(\lambda \boldsymbol{x}_i - \dot{\boldsymbol{p}}_{\mathcal{B}_{a(i)}})$. In case of global intervention, we consider the mismatch between the commanded generalized velocities and their average execution by all the UAVs.

Depending of the actual situation, the resulting forces can provide different cues to the human operator. In case of UAVs possessing large inertia (e.g., when the UAVs are

cooperatively transporting a load), the operator would feel high counteracting forces when asking for rapid changes of the group velocity. In case of significant air viscosity, the force would be proportional to the UAV velocity thus giving the impression of pulling a load with some friction. Finally, if the operator tries to push the UAVs against an obstacle as, e.g., a wall, the UAVs would remain still without following the operator commands. Therefore, the force felt would be similar to a spring-like obstacle repulsion. Along these lines, we also refer the interested reader to some preliminary psychophysical studies addressing the problem of human perception and telepresence in such novel bilateral control scenarios [20], [21].

C. Stabilization of the teleoperation system

Both in the cases of local and of global intervention, it is necessary to consider the destabilizing effects (e.g., communication delay, packet loss, sample & hold) arising from the interconnection between the bilateral control devices and the agent group.

The PSPM framework [22] can be used as a general tool for guaranteeing master passivity and, therefore, stability of the closed-loop teleoperation system when dealing with the aforementioned destabilizing effects, by acting on the signals exchanged over the communication channel. We omit further details here and refer the interested reader to [16], [22] and references therein for a complete treatment and formal proofs of these statements.

Another possibility is to use the two-layer approach [23] that, enforcing passivity, guarantees a stable behavior. Loosely speaking, this approach separates the exchange of information between bilateral control devices and agents into two layers. The passivity layer is controlled in such a way that a passive energetic interconnection is established between the bilateral control device and the agents. In this way, if the bilateral control device and the agents are passive, a stable behavior of the teleoperation system is guaranteed independently of sampling, variable delays and loss of communication. The transparency layer determines the inputs to be implemented on the bilateral control device and on the agents in order to obtain the desired performance (e.g., (7)). These desired inputs are then elaborated by the passivity layer that, exploiting the energy exchanged between local and remote sites, is able to implement them in a passivity-preserving way. For further details, the interested reader can consult [23], [24].

V. Quadrotor-based Experimental Testbed

A. Hardware Setup

1) UAV setup: We used quadrotors as UAVs because of their versatility, robustness, and construction simplicity. Furthermore, the quadrotor model perfectly matches our assumptions of Sec. II, i.e., ability to track the reference quantities $(\boldsymbol{p}_i, \psi_i)$ generated by the FOTP owing to the flatness of the quadrotor outputs $(\boldsymbol{p}_{\mathcal{B}_i}, \psi_{\mathcal{B}_i})$.

Our quadrotor setup (QR) is a customized version of the *MK-Quadro*[1] open-source platform (see Fig. 4a). The

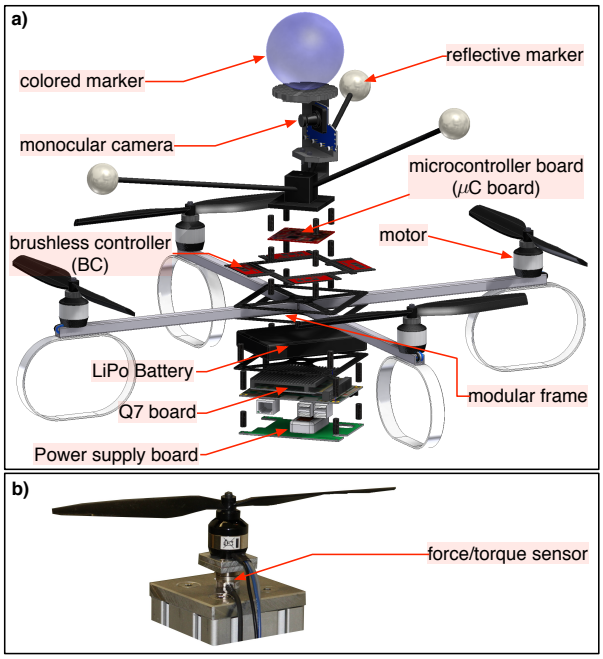

Fig. 4: (a): The quadrotor setup (QR) with its parts; (b): The testbed used for the identification of the motor/propeller dynamics.

QR frame spans $0.5\,\text{m}$, weights $0.12\,\text{kg}$, and is made by 4 aluminum rods joined together in a cross-shape by 2 plastic center-plates. Four brushless Roxxy 2827-35 motors are mounted at the end of each rod. Each motor is driven via a PWM signal by a BL-Ctrl V2.0 brushless controller (BC) which can withstand an average power consumption of $160\,\text{W}$ and peak current of $40\,\text{A}$. A propeller of diameter $0.254\,\text{m}$ is attached to each motor. By means of a custom-made measure testbed using a Nano17 force/torque sensor[2] (see Fig. 4b), we found that the maximum force and torque are $9.0\,\text{N}$ and $0.141\,\text{N}\cdot\text{m}$ respectively, and that the motor dynamics can be approximated by a first order linear system with a time constant of $0.047\,\text{s}$. For control design purposes, this can be in first approximation neglected w.r.t. the mechanical dynamics of the QR.

A $50\,\text{mm}$ square electronic board is mounted in the middle of the frame and hosts a 8-bit Atmega1284p microcontroller (μC) clocked by a $20\,\text{MHz}$ quartz. The μC can send the desired motor speeds to the BCs by means of an I^2C bus. The small board also hosts i) 3D LIS344alh accelerometer with a resolution of $0.0039 g_0\,\text{m/s}^2$ and range of $\pm 2 g_0\,\text{m/s}^2$, and ii) 3 ADXRS610 gyros with a resolution of $0.586\,\text{deg/s}$ and range of $\pm 300\,\text{deg/s}$. These are all accessible by the μC.

A $72 \times 100\,\text{mm}$ Q7 electronic board[3] is mounted underneath the frame. The board hosts a Z530 Intel Atom processor, $1\,\text{GB}$ DDR2 $533\,\text{MHz}$ RAM, an $8\,\text{GB}$ onboard Flash Disk and a WiFi card. The power consumption of this board is $10\,\text{W}$. The Q7 board communicates with the μC through a serial (RS232) cable with a baud-rate up to $115200\,\text{bits/s}$. During the debugging phase the Q7 board can also be dismounted from the QR and operated as desktop computer. In this case,

[1] http://www.mikrokopter.de

[2] http://www.ati-ia.com/

[3] http://www.seco.it/en/, http://www.qseven-standard.org/

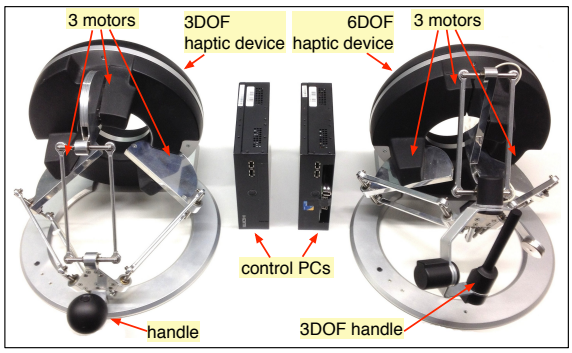

Fig. 5: Haptic interfaces with the corresponding control PCs used in the experimental testbed.

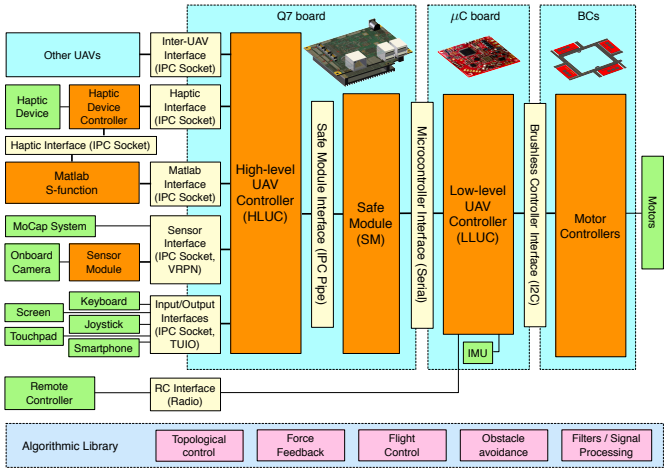

Fig. 6: The software setup.

the cable is replaced by a wireless serial connection XBee-PRO 802.15.4.

An adapted low-cost monocular camera is mounted on top of the μC board. This is connected to the Q7 board through USB, and has a horizontal/vertical field-of-view of about $88/60$ deg and a weight of less than 50 g. A set of reflective markers is used by an external tracking system in order to retrieve the position and orientation of the QR. A single colored ball is instead tracked by the cameras of the QRs in order to measure their relative bearing $\boldsymbol{\beta}_{ij}$.

The QR carries a 4 cell 2600 mAh LiPo battery underneath the Q7 board which powers the whole system by means of a custom power-supply board, allowing to handle the diversity of supplied components. The autonomy provided by the battery in full-configuration is about 10 minutes. All the electronic can also be supplied by an AC/DC socket adapter, e.g., while the battery is replaced.

2) Haptic Interfaces: The bilateral devices used in our testbed are an Omega.3 and an Omega.6[4], as shown in Fig. 5. The Omega.3 has 3 fully actuated DOFs, while the Omega.6 is a 6DOF device with 3 actuated translational and 3 passive rotational DOFs. Each device is connected to a mini PC by means of USB connection and can be controlled at 2.5 kHz. The workspace of the devices is approximately a cube with edge of 0.12 m, and the maximum provided force is about 10 N.

3) Additional components: Our hardware setup includes also additional components that, however, are not described in detail here: these consist of the network infrastructure, the motion capture (MoCap) system[5], and other human interfaces (e.g., joypads, screens, etc.).

B. Software Setup

The distributed software implementation of the whole system involves several processes interconnected through custom-developed interfaces, see Fig. 6. A custom C++ algorithmic library provides the control and signal processing functionalities needed by each process, such as force-feedback algorithms, topological controllers, obstacle avoidance techniques, flight controllers, signal processors and filters.

1) UAV Software: The Q7 board runs a GNU-Linux OS and hosts the *High-level UAV Controller* process (HLUC) that implements the FOTP and part of the FC. The μC board runs a single process, the Low-level UAV Controller (LLUC), which implements the remaining parts of the FC and interfaces directly with the IMU and the 4 motor controllers through the I^{2}C bus. The FC is a standard cascaded controller similar to the one used in [9].

The HLUC can use WiFi to communicate via Socket IPC with several other processes hosted by different machines, like the HLUC of other UAVs, haptic-device controllers, a Matlab instance, a MoCap system (using the VRPN protocol), sensor modules (which may be also hosted on the Q7 board), and input-output interfaces (e.g., using the TUIO[6] protocol, a touchpad or a smartphone). A similar interface allows also direct communication between haptic-device controllers and a Matlab instance, if needed. The communication frequency changes for each interface (e.g., it is $120 \div 240$ Hz for the MoCap System, and 25 Hz for the camera sensor module).

The communication between the HLUC and the LLUC is mediated by the Safe Module (SM) process, a very compact and well tested program whose role is to check the inputs generated by the HLUC and take full control of the UAV in case of detection of some inconsistencies in the input signals (e.g., erroneous frequencies, excessive jittering, etc.). In addition to the SM, in emergency cases a human operator may also bypass the Q7 board and manually control the UAV using a Radio Remote Controller.

The LLUC provides an estimate of roll and pitch angles $(\phi_{\mathcal{B}_i}, \theta_{\mathcal{B}_i})$ by fusing the IMU readings with a complementary filter. The yaw $\psi_{\mathcal{B}_i}$ can be either measured with an onboard compass, or retrieved by the MoCap system. A sensor module processing the onboard camera signal is used to obtain the relative bearings by detecting the spheres placed on top of the quadrotors. Finally, every BC runs a P-controller in order to regulate the motor speeds.

2) Other Software Components: The haptic device controller implements the local control loop computing the forces

[4]http://www.forcedimension.com
[5]http://www.vicon.com/
[6]http://www.tuio.org/

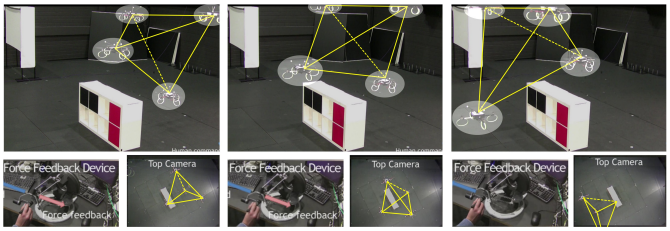

Fig. 7: An experiment employing constant topology controller and global intervention.

$\boldsymbol{\tau}^c$ at a frequency of 2.5 kHz, sends the needed quantities (e.g., $\boldsymbol{x}$) to the HLUC, and receives the UAV measurements needed to implement $\boldsymbol{\tau}^c$. An instance of Matlab can also be interfaced with HLUC for debugging and testing purposes (e.g., offline post-processing), or for fast-prototyping of complex control algorithms.

C. Illustrative Experiments

Figures 7 and 8a show two experimental sequences where our bilateral shared control framework is exploited in order to assist the flight of 4 quadrotors using one of the 3DOF haptic interfaces described before.

In the sequence of Fig. 7, we used the constant topology controller based on artificial potentials described in [16]. This controller keeps the desired inter-distances constant and leaves free 6DOF of the UAV group (3 for rotations and 3 for translation). Note how the interaction links (in yellow) are not changing over time because of the chosen fixed topology. The resulting desired formation is a tetrahedron. The human global intervention is applied only to the translational DOFs, and the human telepresence is realized by applying a force to the haptic device proportional to the mismatch between the desired and current centroid velocity. From the reported sequence, one can appreciate how the actions of the human command and of the obstacle avoidance produce a rotation of the whole formation, thus allowing to overcome the obstacles.

In the sequence of Fig. 8a, we used the unconstrained topology controller described in [18], [25]. The controller in this case does not overly constrain the topology: this is clearly visible in the sequence since the interaction links are changing over time. The human intervention is local and is limited to the UAV highlighted with the red ball. The motion of the UAVs leads to several split and rejoins, triggering in some cases the tank/spring energy exchange needed to preserve passivity of the slave side described in Sec.III-D. The force cue $\boldsymbol{\tau}_i^c$ displayed to the human operator during the experiment is shown in Fig. 8b: the peaks of $\boldsymbol{\tau}_i^c$ occur during the transient discrepancies between $\boldsymbol{u}_i^h$ and $\dot{\boldsymbol{p}}_{\mathcal{B}_{a(i)}}$. These inform the human operator about the 'lag' between the connected UAV and its velocity command. Figure 8c shows the evolution of the 6 inter-agent potentials (links) over time. At the beginning of the motion, 3 links start not connected with their potential at the 'infinity' value 0.5 [J] while, as time goes on, new links are created/destroyed as can be seen from the various jumps in the inter-agent potentials. Figure 8d shows the superimposition of the external energy supplied to the slave system (blue solid line) and the variation of the internal UAV-group energy (red dashed line). From this plot, it is possible to verify that the passivity condition for the group of UAVs is always met.

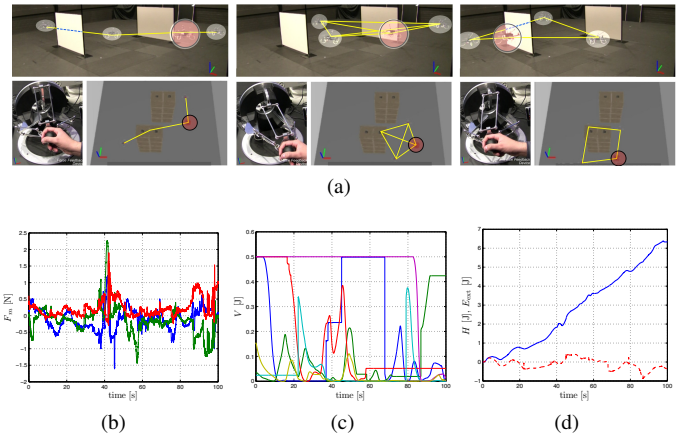

Fig. 8: An experiment with unconstrained topology controller and local intervention. (b): force displayed to the human operator on the bilateral control device; (c): behavior of the 6 link potentials over time; (d): Behavior of the external energy supplied to the UAV-group system (solid blue line), and the internal UAV-group energy (dashed red line).

We finally encourage the interested reader to watch the videos of these and other experiments based on the presented framework at http://www.youtube.com/user/MPIRobotics/videos.

VI. Conclusions and Future Work

In this paper we presented a control framework and its associated experimental testbed for the *bilateral shared control* of a group of quadrotor UAVs. The control architecture allows to integrate a topological motion controller, a human assistance module, and a force-feedback possibility to increase the telepresence of the human assistants. The versatility of the proposed framework has been demonstrated by means of experiments using a hardware and software architecture based on quadrotor UAVs.

In the future, we aim at extending our testbed also to outdoor scenarios, thus replacing the MoCap system with GPS and/or other localization algorithms. We are also considering the possibility of combining bilateral shared control with UAVs autonomously performing complex tasks, such as cooperative transportation or exploration. Finally, we are also running extensive user studies in order to better assess the immersiveness and quality of feedback provided to the human assistants during the operation of the UAVs.

VII. Acknowledgments

This research was partly supported by WCU (World Class University) program funded by the Ministry of Education, Science and Technology through the National Research Foundation of Korea (R31-10008). The authors also wish to thank Dr. Hyoung Il Son, Carlo Masone and Volker Grabe for their useful help in the development and implementation of some of the presented controllers and of the quadrotor hardware setup.

REFERENCES

[1] M. Schwager, B. Julian, M. Angermann, and D. Rus, "Eyes in the sky: Decentralized control for the deployment of robotic camera networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1541–1561, 2011.

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

[3] X. C. Ding, M. Powers, M. Egerstedt, R. Young, and T. Balch, "Executive decision support: Single-agent control of multiple UAVs," *IEEE Robotics & Automation Magazine*, vol. 16, no. 2, pp. 73–81, 2009.

[4] R. Murphy, J. Kravitz, S. Stover, and R. Shoureshi, "Mobile robots in mine rescue and recovery," *IEEE Robotics & Automation Magazine*, vol. 16, no. 2, pp. 91–103, 2009.

[5] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[6] 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*, Anchorage, AK, May 2010, pp. 5302–5308.

[7] T. M. Lam, H. W. Boschloo, M. Mulder, and M. M. V. Paassen, "Artificial force field for haptic feedback in UAV teleoperation," *IEEE Trans. on Systems, Man, & Cybernetics. Part A: Systems & Humans*, vol. 39, no. 6, pp. 1316–1330, 2009.

[8] M. Valenti, B. Bethke, D. Dale, A. Frank, J. McGrew, S. Ahrens, J. P. How, and J. Vian, "The MIT indoor multi-vehicle flight testbed," in *2007 IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, Apr. 2007, pp. 2758–2759.

[9] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP multiple micro-UAV testbed," *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.

[10] S. Lupashin, A. Schöllig, M. Hehn, and R. D'Andrea, "The flying machine arena as of 2010," in *2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010, pp. 2970–2971.

[11] V. Mistler, A. Benallegue, and N. K. M'Sirdi, "Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback," in *10th IEEE Int. Symp. on Robots and Human Interactive Communications*, Bordeaux, Paris, France, Sep. 2001, pp. 586–593.

[12] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995.

[13] B. Bethke, M. Valenti, and J. P. How, "UAV task assignment," *IEEE Robotics & Automation Magazine*, vol. 15, no. 1, pp. 39–44, 2008.

[14] M. Schwager, N. Michael, V. Kumar, and D. Rus, "Time scales and stability in networked multi-robot systems," in *2011 IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, May 2011, pp. 3855–3862.

[15] A. Franchi, C. Masone, H. H. Bülthoff, and P. Robuffo Giordano, "Bilateral teleoperation of multiple UAVs with decentralized bearing-only formation control," in *2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Francisco, CA, Sep. 2011, pp. 2215–2222.

[16] D. Lee, A. Franchi, P. Robuffo Giordano, H. I. Son, and H. H. Bülthoff, "Haptic teleoperation of multiple unmanned aerial vehicles over the internet," in *2011 IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, May 2011, pp. 1341–1347.

[17] A. Franchi, P. Robuffo Giordano, C. Secchi, H. I. Son, and H. H. Bülthoff, "A passivity-based decentralized approach for the bilateral teleoperation of a group of UAVs with switching topology," in *2011 IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, May 2011, pp. 898–905.

[18] 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, in Press*, 2012.

[19] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "Passivity-based decentralized connectivity maintenance in the bilateral teleoperation of multiple UAVs," in *2011 Robotics: Science and Systems*, Los Angeles, CA, Jun. 2011.

[20] H. I. Son, J. Kim, L. Chuang, A. Franchi, P. Robuffo Giordano, D. Lee, and H. H. Bülthoff, "An evaluation of haptic cues on the tele-operator's perceptual awareness of multiple UAVs' environments," in *IEEE World Haptics Conference*, Istanbul, Turkey, Jun. 2011, pp. 149–154.

[21] H. I. Son, L. L. Chuang, A. Franchi, J. Kim, D. J. Lee, S. W. Lee, H. H. Bülthoff, and P. Robuffo Giordano, "Measuring an operator's maneuverability performance in the haptic teleoperation of multiple robots," in *2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Francisco, CA, Sep. 2011, pp. 3039–3046.

[22] D. J. Lee and K. Huang, "Passive-set-position-modulation framework for interactive robotic systems," *IEEE Trans. on Robotics*, vol. 26, no. 2, pp. 354–369, 2010.

[23] M. Franken, S.Stramigioli, S. Misra, C. Secchi, and A. Macchelli, "Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency," *IEEE Trans. on Robotics*, vol. 27, no. 4, pp. 741–756, 2011.

[24] C. Secchi, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "Bilateral teleoperation of a group of UAVs with communication delays and switching topology," in *2012 IEEE Int. Conf. on Robotics and Automation*, St. Paul, MN, May 2012, pp. 4307–4314.

[25] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "Experiments of passivity-based bilateral aerial teleoperation of a group of UAVs with decentralized velocity synchronization," in *2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Francisco, CA, Sep. 2011, pp. 163–170.

VIII. AFFILIATIONS

Antonio Franchi, Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany, antonio.franchi@tuebingen.mpg.de.

Markus Ryll, Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany, markus.ryll@tuebingen.mpg.de.

Crisitan Secchi, Department of Science and Methods of Engineering, University of Modena and Reggio Emilia, via G. Amendola 2, Morselli Building, 42122 Reggio Emilia, Italy cristian.secchi@unimore.it.

Heinrich H. Bülthoff, Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany, and Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul, 136-713 Korea, hhb@tuebingen.mpg.de.

Paolo Robuffo Giordano, Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany, prg@tuebingen.mpg.de.