2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

Bilateral Teleoperation of a Group of UAVs with Communication Delays and Switching Topology

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

Abstract— **In this paper, we present a passivity-based decentralized approach for bilaterally teleoperating a group of UAVs composing the slave side of the teleoperation system. In particular, we explicitly consider the presence of time delays, both among the master and slave, and within UAVs composing the group. Our focus is on analyzing suitable (passive) strategies that allow a stable teloperation of the group despite presence of delays, while still ensuring high flexibility to the group topology (e.g., possibility to autonomously split or join during the motion). The performance and soundness of the approach is validated by means of human/hardware-in-the-loop simulations (HHIL).**

I. INTRODUCTION

Groups of mobile robots have proven to be very effective in solving complex tasks like surveillance, exploration and search and rescue, and the problem of implementing a desired emerging behavior of a group has received a lot of attention in the last years (see, e.g., [1], [2]). Nevertheless, when the task is very complex, autonomy of the fleet is far from being reached and the human intervention is still necessary. Therefore, bilateral teleoperation of a semi-autonomous group seems a very promising solution. In fact, in this way the user can guide the robots by a master interface and receive an informative feedback on the status of the group. In particular, a force feedback should be included since it improves the skills of the human during teleoperation as shown in [3].

The bilateral teleoperation of a group of robots is an emerging research area in the telerobotics community and several works addressed this problem [4], [5], [6]. Because of their extreme flexibility and their high mobility, teleoperation of groups of Unmanned Aerial Vehicles (UAVs) (the so-called *Aerial Teleoperation*) has been addressed in [7], [8], [9], [10], [11]. In particular, in [9] a highly flexible decentralized control strategy was proposed in which the UAVs were allowed to change the shape of the group by means of split and join maneuvers, while guaranteeing both inter-agent and obstacle collision avoidance. In this way, the user could simply focus on the motion of the group and feel the state of the fleet and of its interaction with the remote environment through a visual and force feedback.

C. Secchi is with the 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

P. Robuffo Giordano and A. Franchi are with the Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany {prg,antonio.franchi}@tuebingen.mpg.de.

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

However, as in any teleoperation setting, non-negligible time delays can be present both in the communication channel between master and slave side, as well as within the members of the group. Although both kinds of delays have destabilizing effects if not properly handled, this problem is rarely addressed in the multi-slave teleoperation literature. A notable exception are [4], [7] where the passifying PD action proposed in [12] is used for handling the delayed communication between master and slave.

In this context, the goal of this work is to extend the (passivity-based) framework proposed in [9] so as to explicitly consider communication delays between master and slave, and within the (multi-UAV) slave system. In order to preserve the passivity of the teleoperation system and to deteriorate the quality of the force feedback as less as possible, the interconnection between master and slave is implemented by extending the two layer approach proposed in [13] to the multi-slave teleoperation case. This strategy allows to obtain the highest possible transparency allowed by the passivity constraint. Furthermore, in order to preserve the flexibility of the group of UAVs, we also detail two new procedures for passively implementing split and join maneuvers in case of delayed communication among the agents at the slave side.

The paper is organized as follows: in Sec. II a brief summary of the distributed control strategy proposed in [9] is summarized for the reader's convenience. In Sec. III, an extension of the two layer framework for the multi-slave teleoperation case is developed, and Sec. IV presents two procedures for implementing split and join maneuvers in case of communication delays. After discussing human/hardware-in-the-loop simulations in Sec. V aimed at to validating the proposed approach, Sec. VI draws the conclusions and addresses future perspectives.

II. BACKGROUND MATERIAL

In this Section, the passive and decentralized control strategy for the teleoperation of a group of UAVs proposed in [9] is briefly summarized. Each UAV is modeled as a floating mass in $\mathbb{R}^3$ and the slave side consists of a group of N agents among which a leader is chosen. The motion of each UAV depends on the motion of the surrounding vehicles, while the motion of the leader depends also on its coupling with the master. In order to consider the difference between the bounded workspace of the master and the unbounded workspace of a UAV, the position of the master is treated as a velocity setpoint for the leader at the slave side and the difference between the position of the master and the

velocity of the leader generates a force at the master side for feeding back to the user an information about the motion status of the fleet.

A. The Master Side

The master can be a generic fully actuated mechanical system described by the following Euler-Lagrange equations:

$$M_M(x_M)\ddot{x}_M + C_M(x_M, \dot{x}_M)\dot{x}_M + B_M\dot{x}_M = F_M \quad (1)$$

where $M_M(x_M)$ represents the inertia matrix, $C(x_M, \dot{x}_M)\dot{x}_M$ is a term representing the centrifugal and Coriolis effects, and B_M is a matrix representing both the viscous friction present in the system and any additional damping injection via local control actions. We also assume that gravity is locally compensated. The variables x_M and $v_M := \dot{x}_M$ represent the position and the velocity of the end-effector. In order to passively couple the position of the master to the velocity of the leader, in [9] we proposed a local control loop that renders the master passive with respect to the pair (F_M, r_M) at the expense of possible increases in the master damping B_M, where

$$r_M = \rho v_M + \rho\lambda x_M, \qquad \rho > 0,\ \lambda > 0. \quad (2)$$

By properly choosing the design parameters ρ and λ , it is possible to reduce the contribution of v_M and to make the second term equal to Kx_M, where K is a desired scaling factor. Thus, the r_M variable approximates the (scaled) position of the master but, during fast motions, the velocity term $\rho\lambda v_M$ may act as a disturbing effect.

B. The Slave Side

1) Model of the agents: The UAVs are assumed to be endowed with a Cartesian trajectory tracking controller (as, for instance, the one proposed in [14]) ensuring a closed loop behavior close enough to that of a fully actuated floating mass in $\mathbb{R}^3$. We then model each agent, and its local control structure, as:

$$\begin{cases} \dot{p}_i = F_i^{\mathrm{a}} + F_i^{\mathrm{e}} - B_i M_i^{-1} p_i \\ \dot{t}_i = (\alpha_i \frac{1}{t_i} D_i + w_i) \\ y_i = \begin{pmatrix} M_i^{-1} p_i \\ t_i \end{pmatrix} \end{cases} \qquad i = 1, \ldots, N. \quad (3)$$

Here, $p_i \in \mathbb{R}^3$ and $M_i \in \mathbb{R}^{3\times 3}$ represent the momentum and the inertia matrix of agent i, respectively, $\mathcal{K}_i = \frac{1}{2} p_i^T M_i^{-1} p_i$ is the kinetic energy stored by the agent during its motion, and $B_i \in \mathbb{R}^{3\times 3}$ is a positive semidefinite matrix representing an artificial damping added for asymptotically stabilizing the behavior of the agent. Forces $F_i^{\mathrm{a}} \in \mathbb{R}^3$ and $F_i^{\mathrm{e}} \in \mathbb{R}^3$ represent the interaction of agent i with other agents and with the external world (i.e., the obstacles or the master side), respectively.

The power dissipated by the agents because of their local damping B_i, i.e.,

$$D_i = p_i^T {M_i^{-1}}^T B_i M_i^{-1} p_i, \quad (4)$$

is monitored and stored back into the local energy variable $T_i = \frac{1}{2} t_i^2 \in \mathbb{R}$ called *tank*, whose state $t_i \in \mathbb{R}$ augments the agent dynamics in (3). The quantity $\alpha_i \in \{0, 1\}$ in (3) is a control parameter used to disable/enable the storage of D_i into the tank. In fact, because of the reasons reported in [15], the energy stored in the tanks needs be bounded by above in order to avoid the implementation of practically unstable behaviors. Thus, α_i is set to 0 if the energy stored in the tank reaches an upper bound $\bar{T}_i$ to be selected depending on the particular application. We also set a small threshold $\epsilon > 0$ below which energy extraction from the tank is prevented, in order to avoid singularities in (3).

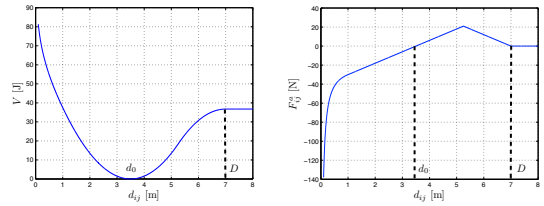


Fig. 1: The shape of the interagent potential as a function of the distance (left), and its corresponding coupling force (right).

The quantity $w_i \in \mathbb{R}$ in (3) is an additional input used to exchange energy with the tank through the port (w_i, t_i). Finally, the outputs of the overall system (3) are the velocity of the agent $v_i = M_i^{-1} p_i$ and the tank state t_i. It is possible to prove that system (3) is passive w.r.t. its input/output ports, see [9].

2) Decentralized inter-agent interactions: Two agents are assumed to sense each other and to communicate (i.e., they are considered as *neighbors*) if their relative distance d_{ij} is less than $D \in \mathbb{R}^+$. Furthermore, agents can measure the distance from any obstacle located within the range D. The agents are coupled so as to achieve a flexible, cohesive and collision free behavior by means of a set of interaction forces

$$F_i^{\mathrm{a}} = \sum_{j \neq i} F_{ij}^{\mathrm{a}}. \quad (5)$$

For each neighboring agent j, agent i computes an *inter-agent* interaction force F_{ij}^{a} whose magnitude and direction depends on the relative distance and bearing respectively. This force is designed so as to regulate the distance between the agents to a desired value d_0, to prevent collisions between the agents, and to vanish as the distance among the agents becomes larger than D. Figure 1 depicts the shape of a possible inter-agent force and associated scalar potential as a function of the inter-robot distance d_{ij}.

As explained in [9], such interaction force can be expressed as a nonlinear spring whose lower bounded potential energy function V depends on the relative position among the agents $x_{ij} := x_i - x_j \in \mathbb{R}^3$. Formally,

$$\begin{cases} \dot{x}_{ij} = v_{ij} \\ F_{ij}^{\mathrm{a}} = \frac{\partial V(x_{ij})}{\partial x_{ij}} \end{cases} \quad (6)$$

where $v_{ij} = v_i - v_j$ is the relative velocity among the agents. We note that the overall interaction force (5) can be computed in a decentralized way since, if agent j is not detected, i.e., it is not a neighbor, it is considered as being farther than D and a null force is implemented.

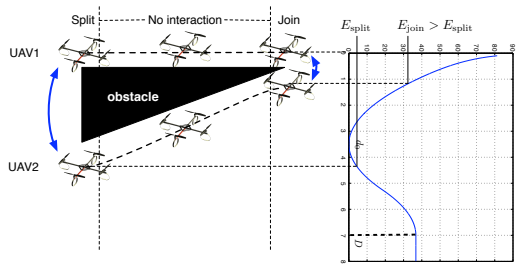


Fig. 2: When two agents split, the energy E_{split} is stored in the spring, while when they join the energy $E_{\text{join}} > E_{\text{split}}$ is needed to implement the new desired coupling. In this case, without proper strategies, an amount $E_{\text{join}} - E_{\text{split}} > 0$ of energy would be introduced into the system, thus violating passivity.

3) Split and join decisions: In order to enable the fleet to reshape its formation and/or to vary its topology in a flexible way, the agents are allowed to autonomously implement split and join decisions based on suitable strategies depending on the particular situation. We then refer to a *split* as the cancelation of the coupling force F^{a}_{ij} between a pair of agents i and j even though $d_{ij} \leq D$. A *join* is the (re-)establishment of the coupling, e.g., after a split. Clearly, a join can happen only if $d_{ij} \leq D$. Intuitively, a split between two agents mimics the disconnection from the virtual elastic element V_{ij} that represents their coupling. The spring becomes isolated and keeps on storing the same energy that was storing before the split decision, while the agents keep on interacting with the rest of the system. Thus, a split is a passivity preserving decision.

A join decision, on the other hand, can lead to a violation of passivity: when two agents i and j join, they instantaneously switch from a state characterized by no interaction, to the inter-agent interaction represented by (6). Some extra energy can be produced during the join procedure, possibly threatening the passivity of the system. In fact, in the general case, the relative distance of two agents at the join decision can be different from their relative distance at the split decision, and this can result in a non passive behavior: the illustrative example of Fig. 2 shows this situation.

To remedy this problem, we exploit the energy in the tanks T_i in order to passively implement join decisions that would otherwise violate the passivity constraint. In short, in the critical case of Fig. 2, agents i and j implement a join decision only if the sum of the energy stored in their tanks is greater than the energy produced by the (re-)establishment of the coupling. When this is not the case, tanks are suitably recharged by increasing the local damping B_i (or by more sophisticated techniques) until there is enough energy for implementing the join. This procedure can be modeled as an exchange between the tanks and the nonlinear springs that couple the agents through the inputs w_i[1].

4) Passivity of the overall slave side: Since the forces F^{a}_{ij} are symmetric, the interactions among the agents can be modeled as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the vertices represent the agents and an edge (i, j) represents the presence of a spring coupling agent i with agent j.

[1]Full details of this strategy, as well as of additional techniques to suitably redistribute the tank energies among the agents, can be found in [9].

Defining $p = (p_1^T, \ldots, p_N^T)^T \in \mathbb{R}^{3N}$, $B = diag(B_i)$, $x = (x_{12}^T, \ldots, x_{1N}^T, x_{23}^T, \ldots, x_{2N}^T, \ldots, x_{N-1N}^T)^T \in \mathbb{R}^{3\frac{N(N-1)}{2}}$, $v = (v_1^T, \ldots, v_N^T)^T \in \mathbb{R}^{3N}$ and $F^e = (F_1^{eT}, \ldots, F_N^{eT})^T \in \mathbb{R}^{3N}$, the overall slave side (agents-tanks-springs) can be shown to take the compact form:

$$\begin{cases} \begin{pmatrix} \dot{p} \\ \dot{x} \\ \dot{t} \end{pmatrix} = \left[\begin{pmatrix} 0 & \mathcal{I} & 0 \\ -\mathcal{I}^T & 0 & \mathcal{I}_\gamma^T \\ 0 & -\mathcal{I}_\gamma & 0 \end{pmatrix} - \begin{pmatrix} B & 0 & 0 \\ 0 & 0 & 0 \\ -\alpha P B & 0 & 0 \end{pmatrix} \right] \nabla H + G F^{\text{e}} \\ v = G^T \nabla H \end{cases} \tag{7}$$

where $\nabla H = \left(\frac{\partial^T H}{\partial p} \; \frac{\partial^T H}{\partial x} \; \frac{\partial^T H}{\partial t} \right)^T$ and

$$H = \sum_{i=1}^{N} \mathcal{K}_i + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} V(x_{ij}) + \sum_{i=1}^{N} T_i \tag{8}$$

represents the total energy of the overall slave side. Moreover, $\mathcal{I} = \mathcal{I}_\mathcal{G} \otimes I_3$, with $\mathcal{I}_\mathcal{G}$ being the incidence matrix of the graph $\mathcal{G}$ whose edge numbering is induced by the entries of the vector x. Matrix $G = \begin{pmatrix} (I_N \otimes I_3)^T & 0^T \end{pmatrix}^T$, with I_3 and I_N being the identity matrices of order 3 and N respectively, 0 represents a null matrix of proper dimensions, and $\otimes$ denotes the Krönecker product. The matrix $\mathcal{I}_\gamma = \Gamma \circ (\mathbf{1} \otimes \mathcal{I}_\mathcal{G})$, where $\circ$ is the element-wise product, $\mathbf{1} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^T$, and Γ is a matrix of proper dimensions whose elements represent an energetic interconnection between tanks and springs mediated by the inputs w_i in (3). Matrix P describes the storage of energy into the tanks and takes the expression

$$P = diag(\frac{1}{t_i} p_i^T M_i^{-T}) \quad i = 1, \ldots, N \tag{9}$$

Finally, $\alpha = diag(\alpha_i)$is the matrix containing the switching parameters, $t = (t_1, \ldots, t_N)^T$, and $c = (c_1, \ldots, c_N)^T$. It is possible to show that that the system represented in (7) is passive with respect to the pair (F^e, v) using H as a storage function.

C. The Teleoperation System

Let the agent L be the leader. It is possible to decompose $F_L^{\text{e}} = F_s + F_L^{\text{env}}$, where F_L^{env} is the component of the force due to the interaction with the external environment (obstacles) and F_s is the component due to the interaction with the master side. Similarly, we can decompose F_M in (1) as $F_M = F_m + F_h$, where F_h is the component due to the interaction with the user and F_m is the force acting on the master because of the interaction with the slave.

For achieving the desired teleoperation behavior, master and slave sides are joined using the following interconnection:

$$\begin{cases} F_s = -b_T(v_L - r_M) \\ F_m = b_T(v_L - r_M) \end{cases} \tag{10}$$

where $b_T > 0$ is a design parameter. This is equivalent to joining the master and the leader using a damper which generates a force proportional to the difference of the two velocity-like variables of the master and the leader. Since r_M is "almost" the position of the master, see Sect. II-A, the force fed back to the master and the control action sent to the leader are the desired ones. The complete teleoperation

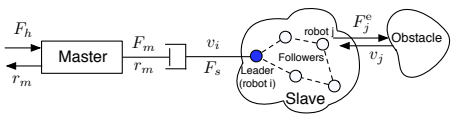


Fig. 3: The overall teleoperation system.

system, represented in Fig. 3, consists then of the interconnection of a passive master side, a passive interconnection and a passive slave side. Recalling that the interconnection of passive systems is again passive, we can conclude that the teleoperation system is passive w.r.t. external actions (human force F_h and environment).

III. A TWO LAYER APPROACH FOR DELAYED MASTER-SLAVE COUPLING

In this Section, we will extend the bilateral control strategy summarized in Sec. II in order to guarantee a passive behavior in case of non negligible communication delays between master and slave sides. To this end, we will adapt the two layer approach proposed in [13] for the following reasons: first, thanks to its flexibility, this approach can be easily adapted to our framework, and, second, it allows to passively implement any desired coupling force in presence of master/slave communication delays. Therefore, besides providing robustness against delays, it will also be possible to avoid the use of the r_M variable in (10) and to implement the master/slave coupling by directly using the master position x_M.

Consider the Lagrangian model of the master in (1) which can be rewritten in a port-Hamiltonian form [16] as:

$$\begin{cases} \begin{pmatrix} \dot{p}_M \\ \dot{x}_M \end{pmatrix} = \left[\begin{pmatrix} 0 & -I \\ I & 0 \end{pmatrix} - \begin{pmatrix} B_M & 0 \\ 0 & 0 \end{pmatrix} \right] \begin{pmatrix} \frac{\partial H_M}{\partial p_M} \\ \frac{\partial H_M}{\partial x_M} \end{pmatrix} + (F_h + F_M) \\ v_M = \frac{\partial H_M}{\partial p_M} \end{cases} \tag{11}$$

where p_M is the momentum and $H_M = \frac{1}{2} p_M^T M_M^{-1}(x_M) p_M$ is the kinetic energy.

Similarly to what done for the agents within the slave side, we propose to keep track of the energy dissipated by the master using a local tank variable $t_M \in \mathbb{R}$. Thus, the augmented dynamics of the master becomes:

$$\begin{cases} \begin{pmatrix} \dot{p}_M \\ \dot{x}_M \\ \dot{t}_M \end{pmatrix} = \left[\begin{pmatrix} 0 & -I & -w_M \\ I & 0 & 0 \\ w_M^T & 0 & 0 \end{pmatrix} - \right. \\ \left. \begin{pmatrix} B_M & 0 & 0 \\ 0 & 0 & 0 \\ -\alpha_M P_M B_M & 0 & 0 \end{pmatrix} \right] \begin{pmatrix} \frac{\partial H_M}{\partial p_M} \\ \frac{\partial H_M}{\partial x_M} \\ \frac{\partial T_M}{\partial t_M} \end{pmatrix} + \begin{pmatrix} F_h \\ 0 \\ \frac{1}{t_M}(\alpha_M {}^M P_{in} - {}^M P_{out}) \end{pmatrix} \\ v = \frac{\partial H_M}{\partial p_M} \end{cases} \tag{12}$$

where $T_M = \frac{1}{2} t_M^2$, $P_M = \frac{1}{t_M} p_M^T M_M^{-T}(x_M)$, and the quantities ${}^M P_{in}$ and ${}^M P_{out}$ represent power flows that allow the master tank to exchange energy with the external world. The energy dissipated by the master is given by:

$$D_M = p_M^T M_M^{-T} B_M M_M^{-1} p_M \tag{13}$$

Using (12) it can be seen that

$$\dot{T}_M = \alpha_M D_M + w_M^T \frac{\partial H_M}{\partial p_M} + (\alpha_M {}^M P_{in} - {}^M P_{out}) \tag{14}$$

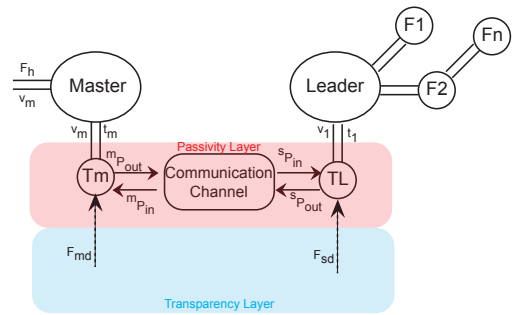


Fig. 4: The two layer architecture for the multi-slave teleoperation.

As before, the parameter $\alpha_M \in \{0, 1\}$ disables the storage of energy in the tank when $T_M(t_M)$ is greater than a selected upper bound $\bar{T}_M$. The variable $w_M \in \mathbb{R}^N$ is a control parameter that allows to exploit (i.e., to extract or inject) the energy of the tank for reproducing the desired control action F_m from (10) by setting $w_M = -\frac{F_m}{t_M}$. In order to avoid singularities ($t_M = 0$), it is necessary to set a lower threshold $\varepsilon_M > 0$ under which energy cannot be extracted from the tank (either using w_M or ${}^M P_{out}$).

Now consider the leader: this is subject to three forces, i.e., the master/leader coupling force F_s, the inter-agent force F_L^a and the external (obstacle) force F_L^{ext}. In order to couple the leader with the master in presence of delays, we extend again the model reported in (3) for allowing the leader to implement F_s using the energy stored in its tank. Thus, the leader model becomes:

$$\begin{cases} \dot{p}_L = -B_L M_L^{-1} p_L - w_s t_L + F^* \\ \dot{t}_L = \frac{\alpha_L}{t_L} D_L + \frac{1}{t_L}(\alpha_L {}^s P_{in} - {}^s P_{out}) + w_s^T v_L + w_L \\ y_L = \begin{pmatrix} v_L \\ t_L \end{pmatrix} \end{cases} \tag{15}$$

where $F^* = F_L^{env} + F_L^a$ and the term $w_s \in \mathbb{R}^3$ is used for implementing the desired coupling force F_s by setting $w_s = -\frac{F_s}{t_L}$. As for the input w_L, this is still used to couple the tank with the rest of the fleet as explained in the previous section. Finally, the power flows ${}^s P_{in}$ and ${}^s P_{out}$ are used to model an exchange of energy with external entities (the master in our case).

Let T be the constant delay characterizing the communication channel along which the master and leader exchange information. Following the idea discussed in [13], we propose a teleoperation architecture, illustrated in Fig. 4, in which the exchange of information between master and slave side takes place over two distinct layers. In the passivity layer, energy is exchanged between the tanks using the signals ${}^\star P_{in}$ and ${}^\star P_{out}$, where $\star = M, s$. Ensuring a proper exchange of energy at this layer is sufficient for guaranteeing the passivity of the teleoperation system. In particular, we propose to interconnect the tanks over the communication channel by:

$$\begin{cases} {}^s P_{in}(t) = {}^M P_{out}(t - T) \\ {}^M P_{in}(t) = {}^s P_{out}(t - T) \end{cases} \tag{16}$$

The desired coupling forces are then generated locally using only the energy available in the tanks: therefore, one can use the exact position of the master x_M instead of the r_M variable in (10). Over the transparency layer, the leader sends its velocity v_L to the master which computes $F_M(t) = b_T(v_L(t-T) - x_M(t))$, the desired force to feedback to the robot. Similarly, the master sends its position to the slave which computes the desired coupling force $F_s = -b_T(v_L(t) - x_M(t-T))$. These forces are implemented using the energy stored in the tank as shown in (12) and (15).

The tanks of the leader and of the master need to be always "full enough" for allowing to passively reproduce the desired control action. Therefore, for each tank, we define the quantities $T^e_\bullet$ where $\bullet = M, L$ and $\varepsilon < T^e_\bullet < \bar{T}_\bullet$, to represent a predefined emergency threshold below which the tank needs to be refilled. In order to control the exchange of energy along the communication channel, the master sends to the leader an energy request signal ${}^M E_{req}(t)$ defined as:

$$ {}^M E_{req}(t) = \begin{cases} 1 & if \quad T_M(t) < T^e_M \\ 0 & if \quad T_M(t) \geq T^e_M \end{cases} \tag{17} $$

In a similar way, the leader sends an energy request signal ${}^s E_{req}(t)$ to the master. The overall power extracted from the master tank and sent to the leader is then given by:

$$ {}^M P_{out}(t) = (1-\alpha_M) D_M)t) +^s E_{req}(t-T)\tau(T_M(t),^e T_M)\bar{P} \tag{18} $$

where $\tau(T_M(t),^e T_M)$ is a threshold function which is 0 if $T_M(t) <^e T_M$ and 1 if $T_M(t) \geq^e T_M$. When the tank has reached its maximum level, $\alpha_M = 0$ and all the energy dissipated is sent to the tank of the leader. If the master receives a request of energy of from the tank (${}^{req}E_s(t-T) = 1$) and if the tank is above the emergency threshold ($\tau(T_M(t),^e T_M) = 1$) a constant flow of power $\bar{P} > 0$ is extracted from the tank of the master and sent to the slave. Similarly, the overall power sent from the leader to the master is

$$ {}^s P_{out}(t) = (1-\alpha_L) D_L +^M E_{req}(t-T)\tau(T_L(t),^e T_L)\bar{P} \tag{19} $$

Remark 1: It can happen that the energy stored in the tank approaches to ε either because the power flow from the other side has not arrived yet or because there is not enough energy in the system. In this case, the only option for refilling the tank is to augment the local damping. The price to pay is a spurious dissipative force temporarily acting on the system.

Proposition 1: The system consisting of the master represented in (12) and of the the leader represented in (15) that exchange energy as indicated in (16) is passive.

Proof: The power flow along the communication channel is given by

$$ \dot{H}_c(t) =^M P_{out}(t) -^M P_{in}(t) +^s P_{out}(t) -^s P_{in}(t) \tag{20} $$

Using (16) in (20) we have that

$$ \dot{H}_c(t) = \frac{d}{dt}\int_{t-T}^{t} ({}^M P_{out}(\tau) +^s P_{out}(\tau))d\tau \tag{21} $$

Thus, the energy exchanged by the tanks is stored in the communication channel and is described by the following lower bounded energy function

$$ H_c(t) = \int_{t-T}^{t} ({}^M P_{out}(\tau) +^s P_{out}(\tau))d\tau > 0. $$

Consider now the combined master/leader/channel energy function:

$$ \begin{aligned} \mathcal{H}(t) &= H_M(t) + T_M(t) + H_L(t) + T_L(t)(\tau) \\ &+ \int_{t-T}^{t} ({}^M P_{out}(\tau) +^s P_{out}(\tau))d\tau \end{aligned} \tag{22} $$

Deriving (22) and using (12), (15) we obtain:

$$ \begin{aligned} \dot{\mathcal{H}} &= -(1-\alpha_M)D_M - (1-\alpha_L)D_L + \alpha^M P_{in}(t) - {}^s P_{out}(t) + \\ &+ \alpha_L {}^s P_{in}(t) - {}^s P_{out}(t) + F_h^T v_M + F^* v_L + w_s^T t_L + \dot{H}_c(t) \end{aligned} \tag{23} $$

Since $\alpha_M, \alpha_L \in \{0, 1\}$ the first and the second term are negative. Using (20), we have that

$$ \alpha_M^M P_{in}(t) -^s P_{out}(t) + \alpha_L {}^s P_{in}(t) - {}^s P_{out}(t) \leq -\dot{H}_c(t) \tag{24} $$

Thus, (23) can be rewritten as

$$ \dot{\mathcal{H}} \leq F_h^T v_M + F^* v_L + w_s^T t_L \tag{25} $$

which proves the passivity of the system. ■

Remark 2: In case of unreliable channels, some information can be lost during the communication or the delay can be variable. The passivity of the system can be preserved using the null packet strategy proposed in [17] and replacing with 0 any unreceived information. It is straightforward to adapt the proof proposed in [17] to our master-leader case.

The system composed by the master, the leader and the communication channel is passive independently of the delay. The system has three ports it can interact with the rest of the world. The port (F_h, v_M) allows to interact with the user, the port (v_L, F^*) allows to interconnect the leader with the rest of the fleet and with the environment and the port (w_L, t_L) allows to interconnect the leader to the interagent potential elements for passively implementing split and join maneuvers.

Thus, since the group of followers is a passive system and since the interconnection of passive systems is still passive, the overall teleoperation system, obtained by interconnecting the leader with the rest of the fleet, is passive independently of the communication delay.

IV. INTER-AGENT DECISIONS WITH DELAYS

Every agent is assumed to measure its relative position w.r.t. neighboring agents by means of a position sensor, but inter-agent communication is still needed in order to agree on the split and join decisions. Mobile wireless networks are the most suitable way for implementing a distributed communication among the UAVs but at the cost of some delay, usually variable, in the exchange of information. We will indicate with $\Delta > 0$ an upper bound on the maximum communication delay between any two agents.

The implementation of the split and join maneuvers strongly relies on the exchange of information between the agents. Therefore, in this Section we will analyze the effect of the delay during split/join decisions, and we will detail two procedures for still (passively) executing them in presence of communication delay.

We will assume that the agents share a common clock (this can be done using any standard time synchronization algorithm, see e.g. [18]). Furthermore since the communication happens only among neighboring agents, congestion problems are very limited and we can assume that data is always delivered.

Notice that, since the delay associated to the sensing is assumed negligible, the inter-agent behavior described in (5) can be safely implemented since it only depends on relative position measurements.

A. Delayed Split

In the non-delayed case described in [9], when agent i decides to split from agent j, it sends its decision to agent j so that both are able to implement the decision synchronously since the information is delivered instantaneously. The synchronicity guarantees the passivity of the split whose energetic meaning is that the agents get disconnected from the virtual spring representing their coupling.

If the same strategy was implemented in the delayed case, passivity would not be preserved anymore. In fact, when agent i splits, agent j keeps on generating a force F^a_{ji} until it receives the split decision. During this period the coupling cannot be modeled as a spring anymore because $|F^a_{ij}| \neq |F^a_{ji}|$ and, therefore, the passivity of the fleet is threatened.

In order to ensure the passivity preservation during split decisions, we force agent i and agent j to plan and execute a split at the same time instant in the future. In particular, if agent i decides to split from j, then it sends, at time τ^i_s, a *split notification* to agent j with attached the time $\tau^i_s + \Delta$. This represent the future time at which they are supposed to split. Since the delay is bounded by Δ and the packet is always delivered, agent j will be certainly able to perform the split at the required time. Nevertheless the split may be anticipated in two particular situations. First, in the case that agent j decided also to split from i and sent a previous split notification at time $\tau^j_s < \tau^i_s$ In this case, the split is performed at $\tau^j_s + \Delta$. Second, whenever the two agents are not able to perceive their relative position anymore, because they moved out of the range of their sensors. In this last case the split is synchronous since we assumed the sensing delay to be negligible.

Both agents store in the local variable E_{ij} the energy stored in the virtual spring at the moment of the split. This is done by measuring the relative distance using the onboard sensing.

B. Delayed Join

In the non-delayed case, when agent i decides to join with agent j, it sends a request which contains the amount of energy T_i, stored in its tank, to the agent j, which in turn replies communicating to i the amount of energy stored in the tank T_j. Since there is no delay, they can synchronously compute the amount of energy available, the amount of energy necessary for implementing the join, and decide whether further refilling the tanks or to conclude the join procedure. This synchronous procedure is crucial for the passive implementation of the join.

If the delay in the communication is not negligible, each agent would receive an outdated information about the stored energy. In fact, during the time necessary for delivering the information, the agent keeps on interacting with the rest of the fleet and the energy in the tank can change, e.g., can decrease. This makes the information not suitable for taking a decision about a passive implementation of the join. A redesign of the join procedure is then necessary in order to ensure the passivity of the decision.

In the delayed case the join procedure cannot be instantaneous and must be implemented with an handshaking mechanism. If at time τ^s_i an agent i decides to join with agent j then it sends to agent j a *join request* with attached the amount of energy stored in its tank at time τ^s_i, namely $T_i(\tau^s_i)$. Then agent i waits until the response of j arrives at time $\tau^r_i \leq \tau^s_i + 2\Delta$. If the response is positive, then it contains also the amount of energy stored in the tank of j at the time τ^r_j when the response has been sent, namely $T_i(\tau^r_j)$. The two agents then will try to join exactly at time $\tau^s_i + 2\Delta$. The amount of energy needed for implementing the join at time $\tau^s_i + 2\Delta$ is computed simultaneously by i and j using the sensed inter-distance. If this energy is less than $T_i(\tau^s_i) + T_i(\tau^r_j)$, then both the agents decide simultaneously to join since they have all the needed information to update their tanks. Otherwise the join procedure is aborted and the handshaking has to be started again if agent i is still willing to join with agent j. This last case (the abortion) applies also to the situation when the response from j is negative.

A few clarifications are needed for the proposed algorithm:

1) : A locking mechanism ensures that, in case of join, $T_i(\tau^s_i + 2\Delta) + T_j(\tau^s_i + 2\Delta) \geq T_i(\tau^s_i) + T_i(\tau^r_j)$, i.e., the amount of energy considered available for the join is actually present in the tanks. In fact, we assume that the agent i and j will reply negatively to any other join request from another agent k while they are waiting for the join response (in the case of i) or they are about to try to join (for both agents). This is clearly a conservative strategy which could be in principle improved at the cost of an increased complexity (and fragility) of the join algorithm.

2) : If the join failed due to the absence of enough energy in the tanks, then the two agents may also temporary increase the damping in order to refill their tanks before attempting the join another time.

3) : If agent j sent a join request before receiving the one of agent i, then agent j has the precedence and the roles in the description are exchanged.

4) : During the join negotiation, and in case the decision is refused or aborted, no coupling force is implemented between pairs of agents and, therefore, there is the risk of a collision. This can be avoided by augmenting the local damping for "braking" an agent when it is too close and

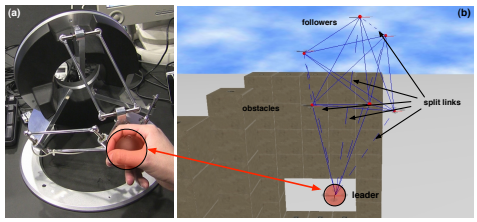
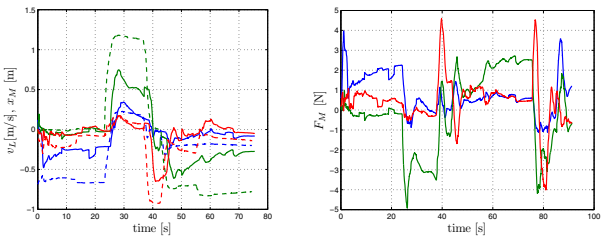


Fig. 5: Human-Hardware-in-the-loop setup



(a) (b)

Fig. 6: Results of the HHIL simulation. Left: superimposition of the leader velocity v_L (solid lines) and the master position x_M (dashed lines). Right: force cue F_m applied on the master and displayed to the human operator

decoupled from another agent.

5) : The tank of the leader can decrease even during a join negotiation because of the interaction with the master. This problem can be solved dividing the tank into two parts: one reserved to the interaction with the fleet and the other to the coupling with the master.

Using this strategy, a synchronous knowledge of the energy available for implementing the join is reproduced. The price to pay to meet the passivity constraint is a more conservative behavior with respect to the non delayed case.

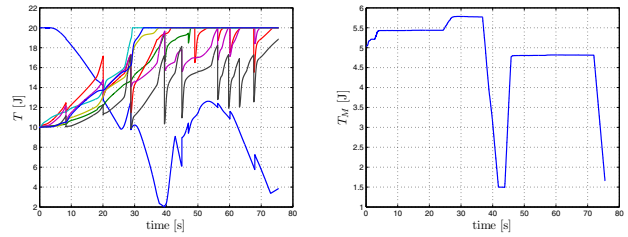
V. HARDWARE-IN-THE-LOOP SIMULATIONS

In this section we report the results of a human/hardware-in-the-loop simulation (HHIL) aimed at validating the passive teleoperation framework developed so far. Figure 5 shows a snapshot of our setup which consists of a 3-DOF force-feedback device, the Omega.3[2]. As for the UAVs, they were simulated in a physically realistic 3D environment based on the Ogre3D engine and the PhysX libraries for simulating the interaction between the UAVs and the environment.

We considered $N = 8$ agents and simulated a master/slave delay $T = 1$ $[s]$ and a maximum inter-agent communication delay $\Delta = 1$ $[s]$. At every message sent by an agent, the actual communication delay δ was randomly drawn from a uniform distribution taking values in $(0, \Delta]$. Furthermore, in order to show the generality of our approach, we did not consider a particular strategy for taking split and join decisions, but implemented them as completely random events with independent probabilities of 1% of being triggered at every simulation step.

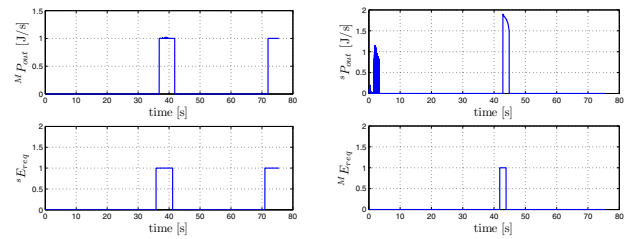
Figure 6(a) shows the behavior over time of the three components of the leader velocity $v_L(t)$ (solid lines), and the corresponding (delayed) master command $x_M(t - T)$

[2]http://www.forcedimension.com



(a) (b)

Fig. 7: Results of the HHIL simulation. Left: evolution of the tanks energies T_i for the $N = 8$ agents. Right: evolution of the master tank energy T_M



(a) (b)

Fig. 8: Results of the HHIL simulation. Top left: energy flow sent from the master to the leader. Bottom left: energy request signal from the leader to the master. Top and bottom right: symmetric signals from the leader to the master and viceversa

(dashed lines) during the HHIL simulation. We can note the following: the leader is in general tracking the master velocity commands with a persistent steady-state error which is especially visible when $x_M \simeq const$. This is an expected and desired behavior of our teleoperation framework due to the 'drag' force exerted by the agents on the leader because of their local damping B_i, see [9], [11] for more details. In fact, it can be proven that this mismatch results into a force cue informing the human operator about the absolute velocity of the agents in the group. Still because of the interaction of the leader with the other agents, one can note the occasional small jerky behaviors in v_L: these are mainly due to the random split/join decisions over time taken by the leader and by its neighbors, and are also reflected as small spikes in the force cue displayed to the human operator, as depicted in Fig. 6(b). Here, one can also note how F_m is correctly representing the 'steady-state' mismatch between v_L and $x_M(t-T)$. As a final remark, note how all quantities keep bounded, proving the stability of the system despite the communication delays.

Figure 7(a) reports the behavior of the $N = 8$ tank energies T_i over time, while Fig. 7(b) the behavior of T_M, the master tank energy. Note the frequent negative jumps into the the tanks T_i: these are the energy amounts extracted from the tanks in order to passify unsafe join decisions, as explained in Sect. II-B. Note also how the tanks get refilled over time thanks to the energy dissipated by every agent during its motion. The same also holds for the master tank T_M, although, since the master was kept almost fixed for large portions of the simulation, its tank replenishment proceeded at a slower rate compared to the agents. It is

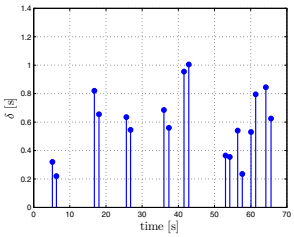


Fig. 9: Results of the HHIL simulation. Random values of δ, the communication delay applied to the messages sent by agent 7 at every split/join request

interesting to note that at about $t \approx 36$ $[s]$ one of the agent tanks (the one of the leader) dropped below the emergency value $T_L^e = 5$ $[J]$, therefore triggering the energy request signal ${}^sE_{req}$ of Sect. III. To illustrate this point, Figs. 8(a–b) show the time behavior of ${}^MP_{out}$ and ${}^sE_{req}$ (left column), and ${}^sP_{out}$ and ${}^ME_{req}$ (right column). Note how at $t \approx 36$ $[s]$, after an energy request is sent from the leader to the master (${}^sE_{req} = 1$), a corresponding energy flow is sent back from the master to the leader (${}^MP_{out} = \bar{P}$). This is also reflected in the sudden increase of the leader tank (Fig. 7(a)) and corresponding decrease in the master tank (Fig. 7(b)). Because of this energy exchange, also the master tank drops at its emergency value $T_M^e = 1.5$ $[J]$ at about $t \approx 41$ $[s]$, triggering the energy request signal ${}^ME_{req} = 1$. This is followed by an energy flow from the slave to the master (Fig. 8(b)) which allows to exit from this undesirable situation.

As a final plot, we show in Fig. 9(a) the values of the random communication delay δ at every split/join message sent by agent 7. Note how (i) the split/join decision takes place in a random way, and how (ii) the values of δ are randomly distributed over $(0, , \Delta]$ as expected.

We finally encourage the reader to watch the video attached to the paper where a complete HHIL simulation is shown.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, a passive teleoperation strategy for a group of UAVs with switching topology was presented. Building upon previous results in this area, we focused our attention on an often overlooked but actual issue in many teloperation systems, that is, presence of time delays. Specifically, we considered presence of time delays among the master and slave system and within the agents composing a group. The potentially destabilizing action of such delays was dealt with by suitably monitoring the energy flows within the system so that no excess of energy is created over time. This was done by means of passive and decentralized procedures and at the minimum possible expense for the transparency of the teleoperation system.

Although an experimental evaluation of the general teleoperation approach shared by this paper has already been given in [11] in the case of negligible delays, in the future we aim at experimentally testing the procedures discussed in this paper in more realistic settings involving significant communication delays as those discussed in the reported simulations.

ACKNOWLEDGEMENTS

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).

REFERENCES

[1] R. M. Murray, "Recent research in cooperative control of multi-vehicle systems," *ASME Journal on Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 571–583, 2007.

[2] S. Martinez, J. Cortes, and F. Bullo, "Motion coordination with distributed information," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 75 –88, aug. 2007.

[3] B. Hannaford, L. Wood, D. McAffee, and H. Zak, "Performance evaluation of a six-axis generalized force-reflecting teleoperator," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 21, no. 3, pp. 620 –633, may/jun 1991.

[4] N. Do, Y. Yamashina, and T. Namerikawa, "Bilateral teleoperation of multiple cooperative robots with time-varying delay," in *IEEE International Conference on Control Applications*, sept. 2010, pp. 2053 –2058.

[5] S. Sirouspour, "Modeling and control of cooperative teleoperation systems," *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1220–1225, 2005.

[6] Y. Cheung and J. S. Chung, "Cooperative control of a multi-arm system using semi-autonomous telemanipulation and adaptive impedance," in *14th Int. Conf. on Advanced Robotics*, Munich, Germany, Jun. 2009, pp. 1–7.

[7] E. J. Rodríguez-Seda, J. J. Troy, C. A. Erignac, P. Murray, D. M. Stipanović, and M. W. Spong, "Bilateral teleoperation of multiple mobile agents: Coordinated motion and collision avoidance," *IEEE Trans. on Control Systems Technology*, vol. 18, no. 4, pp. 984–992, 2010.

[8] D. J. Lee, A. Franchi, P. Robuffo Giordano, H. I. Son, , and H. H. Bülthoff, "Semi-Autonomous Haptic Teleoperation of Multiple Unmanned Aerial Vehicles over the Internet," in *2011 IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, May 2011.

[9] A. Franchi, P. Robuffo Giordano, C. Secchi, H. Son, and H. Buelthoff, "A passivity-based decentralized approach for the bilateral teleoperation of a group of UAVs with switching topology," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.

[10] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "Bilateral teleoperation of groups of UAVs with decentralized connectivity maintenance," in *Proceedings of the 2011 Robotics: Science and Systems Conference*, Los Angeles, USA, 2011.

[11] ——, "Experiments of passivity-based bilateral aerial teleoperation of a group of uavs with decentralized velocity synchronization," in *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[12] D. J. Lee, O. Martinez-Palafox, and M. W. Spong, "Bilateral teleoperation of a wheeled mobile robot over delayed communication network," in *2006 IEEE Int. Conf. on Robotics and Automation*, Orlando, FL, May 2006, pp. 3298 –3303.

[13] 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 Transactions on Robotics*, vol. 27, no. 4, pp. 741 –756, August 2011.

[14] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, "A Control Approach for Thrust-Propelled Underactuated Vehicles and its Application to VTOL Drones," *IEEE Trans. on Automatic Control*, vol. 54, no. 8, pp. 1837–1853, 2009.

[15] 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.

[16] C. Secchi, S. Stramigioli, and C. Fantuzzi, *Control of Interactive Robotic Interfaces: a port-Hamiltonian Approach*, ser. Springer Tracts in Advanced Robotics. Springer, 2007.

[17] S. Stramigioli, C. Secchi, A. van der Schaft, and C. Fantuzzi, "Sampled data systems passivity and discrete port-hamiltonian systems," *IEEE Trans. on Robotics*, vol. 21, no. 4, pp. 574–587, 2005.

[18] D. Mills, "Internet time synchronization: the network time protocol," *IEEE Trans. on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.