Preprint version 50th IEEE Conference on Decision and Control, December 12-15, 2011, Orlando, FL, USA

Distributed Online Leader Selection in the Bilateral Teleoperation of Multiple UAVs

Antonio Franchi, Heinrich H. Bülthoff and Paolo Robuffo Giordano

***Abstract*— For several applications like data collection, surveillance, search and rescue and exploration of wide areas, the use of a group of simple robots rather than a single complex robot has proven to be very effective and promising, and the problem of coordinating a group of agents has received a lot of attention over the last years. In this paper, we consider the challenge of establishing a bilateral force-feedback teleoperation channel between a human operator (the master side) and a remote multi-robot system (the slave side) where a special agent, the leader, is selected and directly controlled by the master. In particular, we study the problem of distributed online optimal leader selection, i.e., how to choose, and possibly change, the leader online in order to maximize some suitable criteria related to the tracking performance of the whole group w.r.t. the master commands. Human/hardware-in-the-loop simulation results with a group of UAVs support the theoretical claims of the paper.**

I. INTRODUCTION

Teleoperation of a group of mobile robots in a bilateral way (i.e., providing a force feedback to the human operator) is an emerging topic which combines the field of autonomous multi-robot systems and the studies on human-robot interaction. This topic received a rapidly-increasing attention in recent years, starting from [1], [2], [3] up to [4], [5], [6], [7], [8], whose theoretical setting is also shared by this work.

The use of a system with multiple deployable agents instead of a bulky robot presents undoubtable advantages, since a decentralized solution increases the robustness and flexibility of the robotic system. These advantages constitute the main motivation between the high number of recent publications in this field, see [9], [10], [11] just to cite a few. On the other hand, in real scenarios, the complete autonomy of multiagent systems is still far from being a reality because of the complexity and unpredictability of the environment as in, e.g., the case of search and rescue missions. Therefore, in many practical cases, the use of a semi-autonomous group of robots partially guided by a human operator still represents the only viable solution. As an alternative to unilateral teleoperation, the use of suitable sensorial feedback has also been proven to improve the (tele-)presence of the human operator, in particular by exploiting haptic cues via force-feedback [12]. It is then interesting to study the possibility of establishing a bilateral teleoperation channel interfacing a human operator with a remote group of agents possessing some local autonomy, but still bound to follow the high-level human motion commands. In this respect, we focus our attention to the particular case of a swarm of UAVs because of their adaptability and potential pervasiveness to many different scenarios. Nevertheless our results may be seamlessly extended to ground, marine and submarine robots.

In our previous works, we considered the situation where a static communication link was established between the master and either one [4], [6], [8] or a subset [5], [7] of the total number of UAVs. The UAVs connected to the master were denoted as *leaders* of the group, and their selection was made at the beginning of the task without an explicit criterion. This choice was also kept unchanged during the whole task execution. Starting from this scenario, in this work we consider the possibility of letting the agents to autonomously choose online the leader that will be connected to the master in order to maximize some suitable criteria related to the tracking performance of the master motion commands. In this sense we refer here to *online leader selection*, in opposition to the *static leader election* which has been deeply investigated for autonomous multi-agent systems. In the static leader election case, the problem is to find a distributed control protocol such that eventually exactly one agent takes the decision that it is the leader [13]. In [14], the leader election problem is solved by the `FLOODMAX` distributed algorithm using explicit message passing among the formation. In [15], the leader election problem is solved using fault detection techniques and without explicit communication, like in some species of the animal world. However, in all these works the leader election is assumed to be performed only once and the only objective is to select *any one out of many*, i.e., without considering some suitable criteria to be optimized.

The main contributions of these notes are the following: (i) we introduce a practically relevant variant of the static leader election problem, namely the *online leader selection problem*, where the leader can be constantly selected as a function of the current state of the system in order to optimize some suitable criterion; (ii) we propose a distributed algorithm able to solve this problem by considering the tracking performance of a reference trajectory as an optimization criterion; (iii) we implement a bilateral force-feedback teleoperation channel to let a human operator to control the multi-agent slave, and (iv) we prove the stability of the overall system in presence of switching leader, discrete and/or delayed master/slave communication, and bilateral force feedback on the human/master-device.

The paper is organized as follows: in Sec. II we illustrate

A. Franchi and P. Robuffo Giordano are with the Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany {antonio.franchi,prg}@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, Seoul, 136-713 Korea. E-mail: hhb@tuebingen.mpg.de.

the overall teleoperation control architecture. In Sec. III the main results about the slave dynamics and leader selection algorithm are presented. In Sec. IV we show how to implement in a stable (passive) way the force-feedback on the master side. In Sec. V we present some simulation results obtained by using a real haptic device and a human in the loop, and we conclude the paper with Sec.VI.

II. TELEOPERATION CONTROL ARCHITECTURE

The *slave side* of our teleoperation system consists of a group of N Unmanned Aerial Vehicles (UAV) tracking a set of N kinematic agents in $\mathbb{R}^3$. The configuration and dynamics of the agents are described by $\boldsymbol{p} = (\boldsymbol{p}_1, \dots, \boldsymbol{p}_N) \in \mathbb{R}^{3N}$, with $\dot{\boldsymbol{p}}_i = \boldsymbol{u}_i$, where $\boldsymbol{p}_i, \boldsymbol{u}_i \in \mathbb{R}^3$ are respectively the position and the control command of the i-th agent, $i = 1, \dots, N$. In practice, the configuration and velocity of the actual UAVs, denoted in the following as $\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i \in \mathbb{R}^3$, will in general not track exactly the configuration/velocity $\boldsymbol{p}_i, \dot{\boldsymbol{p}}_i$ of the corresponding kinematic agent. This is mainly because of the UAV inertia, actuator limits, sensors noise, air drag, etc. For the sake of this presentation, we assume that the UAVs are embedded with a trajectory tracking controller capable of following the corresponding kinematic agent with good approximation, i.e., by keeping the tracking position and velocity errors small enough. Many UAV tracking controllers proposed in the past literature, such as [16], [17], could be used in this sense. We also note that equivalent assumptions have been made in previous related works: in [10] the authors consider the UAV a simple kinematic integrator, in [11] the UAVs are abstracted again as kinematic integrators with some additional details on low-level PID controls.

The *master device*, with which a human operator interacts, is a mechanical system described by the following Euler-Lagrange equations:

$$M(\boldsymbol{x})\ddot{\boldsymbol{x}} + C(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} = \boldsymbol{\tau} + \boldsymbol{f} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^3$ is the configuration, $M(\boldsymbol{x}) \in \mathbb{R}^{3\times3}$ the positive-definite/symmetric inertia matrix, $C(\boldsymbol{x}, \dot{\boldsymbol{x}}) \in \mathbb{R}^{3\times3}$ the Coriolis matrix, and $\boldsymbol{\tau}, \boldsymbol{f} \in \mathbb{R}^3$ the control and human forces, respectively. Furthermore, teleoperation is performed in the following sense: the position of the master $\boldsymbol{x}$ is transformed into a velocity reference to be followed by the N agents, namely

$$\boldsymbol{u}^\star = \rho\boldsymbol{x},$$

with $\rho \in \mathbb{R}^+$ being a scaling factor. Since in practice the workspace of a master device is limited, we have that $\|\boldsymbol{x}\|_\infty \leq X \in \mathbb{R}^+$ implying that $\|\boldsymbol{u}^\star\|_\infty \leq U^\star = \rho X$.

To enforce the teleoperation scalability w.r.t. the number of agents, we assume a limited bandwidth (i.e., constant w.r.t. N) for the communication channel between master and agents. Therefore we assume that the master can only multicast the desired velocity $\boldsymbol{u}^\star$ to a limited number M of agents, called *leaders*, at some discrete time instant $t_k = kT$, where $k = 0, 1, \dots$, and $T > 0$ is taken as the master-leader communication sampling time. In these notes, we consider the case $M = 1$ (only *one* leader), and we also denote with square brackets $[k]$ the discrete sequence obtained sampling a signal at t_k, e.g., $\boldsymbol{u}^\star[k] := \boldsymbol{u}^\star(t_k)$. In accordance with Def. 1 of Sec. III-C, the N UAVs composing the slave side will then run a distributed algorithm in order to track the discrete signal $\boldsymbol{u}^\star[k]$ in an optimal way by autonomously selecting the leader, i.e., the recipient of the master command $\boldsymbol{u}^\star$. The following Sec. III will illustrate the details of this strategy.

Finally, the bilateral nature of our teleoperation scheme will provide the human user with a force cue $\boldsymbol{\tau}$ informing him/her about the slave side motion status, e.g., the discrepancy between the commanded velocity $\boldsymbol{u}^\star$ and the actual velocities of the agents and UAVs. The description and analysis of the force-feedback master control will be done in Sec. IV.

III. THE SLAVE SIDE

Hereafter, for the sake of simplicity, we consider one-dimensional agents composing the slave-side. We then denote with $p = (p_1 \ \dots \ p_N)^T \in \mathbb{R}^N$ the N positions of the 1-dimensional agents, and with $u^\star[k] \in \mathbb{R}$ the desired velocity sent by the master, whose norm is bounded by $U^\star$. Since the slave dynamics on the 3 axes are decoupled, the results can be straightforwardly reformulated in the 3-dimensional case by means of the Krönecker product. The first-order kinematics of each agent is then rewritten as $\dot{p}_i = u_i$ with $i = 1, \dots, N$, where $u_i \in \mathbb{R}$ is the 1-dimensional control input of the i-th agent.

A. Classical Leader-Follower Dynamics

The primary goal of the slave side is to cohesively follow the desired velocity $u^\star[k]$ received from the master. However, since because of communication constraints the master can unicast only to one agent, we follow the framework proposed in [18], [19], [20] (among the others) in order to propagate the master velocity command to all the agents in a distributed way.

We consider homogeneous leader-follower networks with a fixed, connected, and undirected communication graph $\mathcal{G}$. We denote with $\mathcal{N}_i$ the neighbors of every agent i, i.e., the agents with which i can communicate, and with $a_{ij} \in \{0, 1\}$, $i, j = 1, \dots, N$ the entries of the adjacency matrix of $\mathcal{G}$. The control strategy analyzed in [18], [19] derives from the consensus protocol, with the difference that an agent, called *leader*, is fully controlled by an exogenous signal u, while the others, the *followers*, obey to the agreement protocol:

$$u_i = -\sum_{j\in\mathcal{N}_i\setminus l} a_{ij}(p_i - p_j) - a_{il}(p_i - p_l) \quad \forall i \neq l \tag{2}$$

$$u_l = u, \tag{3}$$

where l is the leader index, and $u \in \mathbb{R}$ represents an exogenous velocity signal for the multi-agent system.

Henceforth we will use the left superscript to denote the lack of a certain index from the superscripted quantity. As an immediate application of this notation, we denote with ${}^l p \in \mathbb{R}^{N-1}$ the vector of follower positions obtained by removing the l-th entry from p. The Laplacian matrix of the sub-graph

obtained by $\mathcal{G}$ removing the leader and its adjacent edges is denoted with ${}^lL \in \mathbb{R}^{N-1\times N-1}$. Lastly we denote with ${}^lA \in \{0,1\}^{N-1\times N-1}$ the diagonal matrix which has ones only on the diagonal entries corresponding to the neighbors of the leader, i.e, ${}^lA = \mathrm{diag}({}^la_{1l}, \dots, {}^la_{(N-1)l})$, with

$$\begin{aligned} {}^la_{il} &= a_{il} && i = 1, \dots, l-1 \\ {}^la_{il} &= a_{(i+1)l} && i = l, \dots, N-1 \end{aligned}$$

With these definitions, (2) can be rewritten in matrix form as

$$ {}^l\dot{p} = -{}^lL\,{}^lp - {}^lA({}^lp - \mathbf{1}p_l), \tag{4}$$

where $\mathbf{1}$ is the column vector of all ones of proper dimensions. Using the property that ${}^lL\mathbf{1} = \mathbf{1}^T{}^lL = \mathbf{0}$, it is straightforward to show that the position error ${}^l\tilde{p} = {}^lp - \mathbf{1}p_l$ and the velocity error ${}^l\tilde{v} = {}^l\dot{p} - \mathbf{1}\dot{p}_l$ have the following dynamics:

$$ {}^l\dot{\tilde{p}} = -{}^lM\,{}^l\tilde{p} - \mathbf{1}u_l \tag{5}$$
$$ {}^l\dot{\tilde{v}} = -{}^lM\,{}^l\tilde{v} - \mathbf{1}\dot{u}_l \tag{6}$$

where ${}^lM = {}^lL + {}^lA$ is a symmetric positive definite matrix for any connected graph $\mathcal{G}$ and leader l (since ${}^lA$ has always a non-negative entry on the main diagonal, see [21])[1]. In conclusion, as well know in the literature, the velocities of the followers converge to the velocity of the leader if $\dot{u}_l \equiv 0$, while, if the stronger condition $u_l \equiv 0$ holds, convergence of the follower-leader positions is also obtained.

B. Leader-Follower Dynamics with Time-varying Leader

In order to let the human operator to control the velocity of the slave side, we set $u_l(t) = u^\star[k]$ in (3) for any $t \in [t_k, t_{k+1})$. Therefore, u_l becomes a piecewise constant signal and $\dot{u}_l$ in (6) cannot be evaluated $t = t_k$. However, in any interval $[t_k, t_{k+1})$, Eq. (6) still holds and can can be rewritten as

$$ {}^l\tilde{v}(t_k) = -{}^lM\,{}^l\tilde{p}(t_k) - \mathbf{1}u^\star[k] \tag{7}$$
$$ {}^l\dot{\tilde{v}} = -{}^lM\,{}^l\tilde{v} \qquad t \in [t_k, t_{k+1}). \tag{8}$$

Departing from [18], [19], in these notes we consider the case in which the leader is time varying, i.e., the signal $l = l(t)$ is switching. The adopted switching policy will be illustrated in Sec. III-C and III-D, while here we focus on the stability of the slave side under a switching $l(t)$.

Proposition 1: The slave side velocity system is asymptotically stable for every switching signal $l(t)$.

Proof: If the commanded velocity $u^\star$ is constant we can rewrite the error dynamics of the whole slave side (leader plus followers) as $\dot{\tilde{v}} = (\Omega_{l(t)} - I)L\tilde{v}$, where $\tilde{v} = \dot{p} - \mathbf{1}u^\star \in \mathbb{R}^N$, $\Omega_l \in \mathbb{R}^{N\times N}$ is a matrix of all zeros but the entry (l,l) set to 1, $I \in \mathbb{R}^{N\times N}$ is the identity matrix, and $L \in \mathbb{R}^{N\times N}$ the Laplacian matrix of $\mathcal{G}$. Considering the dynamics of $\frac{1}{2}\|\tilde{v}\|^2$, we have $\tilde{v}^T\dot{\tilde{v}} = \tilde{v}^T(\Omega_{l(t)} - I)L\tilde{v} = -\tilde{v}^TL\tilde{v}$. In fact, since the l-th entry of $\tilde{v}$ is zero, it follows $\tilde{v}^T(\Omega_{l(t)} - I) = \tilde{v}^T$, implying that $\tilde{v}$ converges to the subspace generated by $\mathbf{1}$, in this case to $0\mathbf{1} = \mathbf{0}$, since one entry of $\tilde{v}$ is constantly zero. Therefore, we can conclude that the slave side switching system is stable with $\frac{1}{2}\|\tilde{v}\|^2$ taken as a common Lyapunov function. The stability for any piecewise constant input $u^\star$ is additionally guaranteed by the fact that $u^\star$ is bounded by $U^\star$ and the slave system is ISS (being a linear system). ■

[1]The correspondent notation in [19] is ${}^lM = L_f$ and ${}^lA = \mathrm{diag}(l)$. Here we used a different notation mainly for two reasons: (1) in our case the leader will change over time, hence L_f needs a reference to the current leader, and (2) the presence of L in L_f can generate confusion with the Laplacian of $\mathcal{G}_l$.

C. Online Leader Selection Problem

We considered a switching policy $l(t)$ aimed at improving the convergence rate of the velocities of the agents to the desired velocity $u^\star[k]$ commanded by the master. From (8) it follows that

$$ \|{}^l\tilde{v}(t)\|^2 \le e^{-2\,{}^l\lambda^{\min}(t-t_k)}\|{}^l\tilde{v}(t_k)\|^2 \quad \forall t \in [t_k, t_{k+1}), \tag{9}$$

where ${}^l\lambda^{\min}$ is the smallest eigenvalue of ${}^lM$, and ${}^l\tilde{v}(t_k)$ is defined in (7). Therefore, since at time t^-_{k+1}

$$ \|{}^l\tilde{v}(t^-_{k+1})\|^2 \le \quad e^{-2\,{}^l\lambda^{\min}T}\|{}^lM\,{}^l\tilde{p}(t_k) + \mathbf{1}u^\star[k]\|^2, \tag{10}$$

it is natural to choose $l(t)$ in order to minimize the right side cost function in (10).

Problem 1 (Optimal Leader Selection): Denote by ${}^lM \in \mathbb{R}^{N-1\times N-1}$ the matrix obtained removing the l-th column and row from the Laplacian of a given graph $\mathcal{G}$, and by ${}^l\lambda^{\min}$ its smallest eigenvalue. Assume a discrete-time desired velocity signal $u^\star[k]$ and the states of the agents $p[k] = (p_1[k] \dots p_N[k])^T$ are given. Find the sequence of leaders $l[k]$ which solves, at each step k, the minimization

$$ \min_{l\in\{1,\dots,N\}} \; e^{-2\,{}^l\lambda^{\min}T}\|{}^lM\,{}^l\tilde{p}[k] + \mathbf{1}u^\star[k]\|^2, \tag{11}$$

where T is the given sampling period and ${}^l\tilde{p}[k] \in \mathbb{R}^{N-1}$ denotes the error vector obtained from $p[k] - \mathbf{1}u^\star[k]$ by removing the l-th entry.

Notice that Prob. 1 assumes a leader signal $l(t)$ switching only at times t_k. However, in virtue of Prop. 1, all the results of this paper still hold if the frequency of switching of $l(t)$ is higher than $1/T$. We did not consider this possibility for the sake of presentation and also because, since the exogenous signal u_l is constant in $[t_k, t_{k-1})$, the evolution of the system is completely described by the initial conditions at t_k.

Notice also that Prob. 1 considers an upper-bound of the system time evolution obtained by taking into account the slowest dynamics of ${}^lM$ related to its smallest eigenvalue ${}^l\lambda^{\min}$. In case of complete knowledge of L, one could perform a more accurate minimization by considering the true evolution of the system, i.e., by knowing all the N eigenvalues of ${}^lM$. This approach, however, would require to run a local algorithm with input size not constant with N, the number of agent, i.e., a not decentralized solution which would violate one of the assumptions of our work.

Remark 1: For $T \to \infty$, the minimization (11) tends to be equivalent to solving $\max_{l\in\{1,\dots,N\}} {}^l\lambda^{\min}$: the current relative positions tend to lose relevance w.r.t. the

graph topology with an infinite horizon. In fact, for any constant c there exists a large enough $\bar{t} > 0$ such that $ce^{-2t \max_{l\in\{1,\dots,N\}} {}^l\lambda^{\min}}$ dominates any other exponential function with a faster decreasing rate, $\forall t > \bar{t}$.

Problem 1 can be extended in several ways depending on the used metric in evaluating the norm in (11). When considering the multi-dimensional case, as opposite to the scalar case we are considering now, we suggest the following two possibilities:

1) use of the 2-norm, i.e., minimizing, over $i = 1, \dots, N$, the sum of the three components of the cost function in (11):

$$\min_{l\in\{1,\dots,N\}} \left(e^{-2\,{}^l\lambda^{\min} T} \sum_{j\in\{x,y,z\}} \|{}^l M^l \tilde{p}_j[k] + \mathbf{1} u_j^\star[k]\|^2 \right), \tag{12}$$

2) use of the infinity norm, i.e., minimizing, over $i = 1, \dots, N$, the maximum of the square root of the three components of the cost function in (11)

$$\min_{l\in\{1,\dots,N\}} \left(e^{-{}^l\lambda^{\min} T} \max_{j\in\{x,y,z\}} \|{}^l M^l \tilde{p}_j[k] + \mathbf{1} u_j^\star[k]\| \right). \tag{13}$$

D. Distributed Algorithm for Online Leader Selection

At each time t_k, $2N$ global quantities are needed in order to solve Prob. 1: namely ${}^1\lambda^{\min}, \dots, {}^N\lambda^{\min}$, and $\|{}^1\tilde{v}(t_k)\|^2, \dots, \|{}^N\tilde{v}(t_k)\|^2$. A single central unit could: (i) compute ${}^1\lambda^{\min}, \dots, {}^N\lambda^{\min}$ at the beginning of the task (since the topology is time-invariant), (ii) keep track of $\|{}^1\tilde{v}(t_k)\|^2, \dots, \|{}^N\tilde{v}(t_k)\|^2$, (iii) solve Prob. 1 and communicate the computed leader to the master. Unfortunately this solution is not scalable with the number of agents.

The objective of these notes, in accordance with the large majority of works on multi-agent systems, is to provide an algorithm which is distributed, i.e., which satisfies the following definition:

Definition 1: An algorithm is distributed if it satisfies at least the following requirements:

1) the input-size of any algorithm run by an agent i is proportional to the number of its neighbors $|\mathcal{N}_i|$ and is not proportional to the total number of agents N,
2) the velocity command from the master $u^\star[k]$ is sent only to $l[k-1]$ (the leader at time t_{k-1}) and can be used only by the new selected leader $l[k]$ which is locally chosen in $\mathcal{N}_{[k-1]} = l[k-1] \cup \mathcal{N}_{l[k-1]}$.

In compliance with Def. 1 we state a local version of the leader selection problem.

Problem 2 (Locally-optimal Leader Selection): Given the definitions of Prob. 1, denote with $c_l[k]$ the cost to be minimized in (11). Find a distributed algorithm which generates a sequence of leaders $l[1], \dots, l[k], \dots$ such that $c_{l[k]}[k] \leq c_i[k]\ \forall i \in \mathcal{N}_{l[k-1]}$.

The solution to Prob. 2 is represented by Algorithm 1, which runs on every agent $i \in \mathcal{N}_{[k-1]}$ and takes as inputs ${}^i\hat{\lambda}$ and $\hat{v}_i(t)$, i.e., the distributed estimates of ${}^i\lambda^{\min}$ and $\|{}^i\tilde{v}(t)\|^2$ respectively. In the following we describe how the distributed estimations can be performed.

Algorithm 1: Distributed Leader Selection

Each agent i has an estimate ${}^i\hat{\lambda}^{\min}$ of ${}^i\lambda^{\min}$ and ${}^i\hat{v}(t)$ of $\|{}^i\tilde{v}(t)\|^2$.

At t_k the previous leader $l[k-1]$ receives $u^\star[k]$ from the master.

Agent $l[k-1]$ sends $u^\star[k]$ to every neighbor $i \in \mathcal{N}_{l[k-1]}$;
Every agent $i \in \mathcal{N}_{[k-1]}$ computes $\hat{c}_i[k]$ using its own estimates ${}^i\hat{\lambda}^{\min}$ and ${}^i\hat{v}(t)$ and sends it back to $l[k-1]$ (if $i \neq l[k-1]$);
Agent $l[k-1]$ computes the set $C = \operatorname{argmin}_{i\in\mathcal{N}_{[k-1]}} \hat{c}_i[k]$;
if $l[k-1] \in C$ **then**
    $l[k] = l[k-1]$;
else
    Agent $l[k-1]$ nominates $l[k]$ in C, e.g., randomly;
Agent $l[k]$ informs the master that it is the current leader;

1) Estimation of ${}^i\lambda^{\min}$: Since the graph $\mathcal{G}$ is time-invariant ${}^i\lambda^{\min}$ is constant. We can then assume that before the actual beginning of the task a preliminary phase is implemented by the robots during which the ${}^1\lambda^{\min}, \dots, {}^N\lambda^{\min}$ are sequentially estimated in a distributed way. A possible sketch of this procedure is: first the agents establish an order in the network. This can be done generating a spanning tree and exploring the three in a depth first order. After, starting from the first in the list, the agents take sequentially the role of the leader in a round-robin fashion. An agent i informs its neighbors as soon as it becomes a leader and starts the estimation of ${}^i\lambda^{\min}$. When the estimate has converged, the agent informs its neighbors that it is not a leader anymore and the next agent on the list becomes the leader. This initialization procedure is done until all the agents have estimated their ${}^i\lambda^{\min}$.

The estimations of the individual ${}^i\lambda^{\min}$ can be obtained by exploiting the distributed version of the continuous-time power iteration method [22]. This method allows to estimate the smallest eigenvalue ${}^l\lambda^{\min}$ of ${}^lM$ by implementing the following estimation dynamics:

$$\dot{\xi} = -k_1\,{}^lM\xi - k_2(\theta_i - 1)\xi \tag{14}$$

where ξ is the estimate of the eigenvector associated to ${}^l\lambda^{\min}$, and θ_i is the distributed estimate of $\mathrm{Ave}\,(\xi^i)^2$ obtained through a PI average consensus estimator [23]. The estimator (14) can be implemented in a distributed way since ${}^lM\xi$ is a distributed operation.

2) Estimation of $\|{}^i\tilde{v}(t)\|^2$: The term $\|{}^i\tilde{v}(t)\|^2$ represents the rightmost part of the cost function in (11). Expand $\|{}^iM^i\tilde{p}(t) + \mathbf{1}u^\star(t)\|^2$ as the three terms:

$$\|{}^i\tilde{v}\|^2 = ({}^iM^i\tilde{p})^T({}^iM^i\tilde{p}) + 2u^\star\mathbf{1}^T({}^iM^i\tilde{p}) + Nu^{\star 2}, \tag{15}$$

where we omitted the time dependency for conciseness. The third term can be computed by the l-th agent (the leader) knowing the number of agents and the master command $u^\star$, and the computed value is then communicated to its neighbors $i \in \mathcal{N}_l$ before Algorithm 1 is run. The second

term can be rewritten as

$$2u^\star \mathbf{1}^T {}^iM^i\tilde{p} = 2u^\star \sum_{j\in\mathcal{N}_i} a_{ij}(p_j - p_i) \tag{16}$$

which can also be computed online by every agent $i \in \{l\} \cup \mathcal{N}_l$, i.e., the current leader l and its neighbors, by relying only the the neighbor information. The first term can be rewritten as

$$({}^iM^i\tilde{p})^T({}^iM^i\tilde{p}) = \zeta - \eta_i^2, \tag{17}$$

where $\eta_i = \sum_{j\in\mathcal{N}_i} a_{ij}(p_i - p_j)$ and $\zeta = (Lp)^T(Lp) = \sum_{i=1}^N \eta_i^2$. The *unique* quantity ζ does not depend on the current leader and can be continuously estimated by every agent by only using local information, for instance via the PI average consensus estimator (PI-ace) [23]. The PI-ace allows N agents, each of which measures the time-varying scalar $\eta_i^2(t)$, to compute an approximation of $\frac{1}{N}\sum_{i=1}^N \eta_i^2(t)$ using only local communication. The estimate of ζ runs in parallel to the motion of the agents and the leader selection algorithm, which use the estimated ζ whenever is needed.

Remark 2: In practice, in order to have a good estimate, a certain bound on the maximum norm of $\dot{\zeta}$ will be required. However this condition will be typically satisfied in our scenario for two main reasons: (i) the agent dynamics (and therefore the dynamics of ζ) cannot be too fast, since it is assumed to be feasible for a real UAV which possess a certain inertia, and (ii) the estimator of ζ runs only on the local network of the slave-side which has normally much more bandwidth compared with the master-slave link.

IV. THE MASTER SIDE

In Sec. II we explained how the multi-agent system is driven by commanding the velocity of the current leader with $\boldsymbol{u}^\star = \rho\boldsymbol{x}$, where $\boldsymbol{x}$ is the position of a 3DOF force feedback device whose dynamic model is given by (1). In this section we describe how the control force $\boldsymbol{\tau}$ in (1) is implemented, in order to provide a suitable haptic feedback to the user while ensuring the stability of the overall teleoperation system during the interaction with the slave side the the human operator.

Henceforth we call *leader UAV* the UAV which is tracking the (kinematic) leader agent $\boldsymbol{p}_l$. In a similar fashion we will refer to the *neighbor UAVs*. At each time t_k, the leader UAV sends to the master controller a signal $\boldsymbol{z}(t) = \boldsymbol{z}_1(t) + \boldsymbol{z}_2(t)$, where

$$\boldsymbol{z}_1(t) = \frac{1}{\rho}\dot{\boldsymbol{q}}_l \tag{18}$$

$$\boldsymbol{z}_2(t) = \frac{1}{\rho}\left(\boldsymbol{p}_l - \frac{1}{\|\mathcal{N}_l\|}\sum_{i\in\mathcal{N}_l}\boldsymbol{q}_i\right). \tag{19}$$

In (18) $\dot{\boldsymbol{q}}_l$ is the actual velocity of the leader UAV and (19) implements the difference between the position of the leader and the positions of its neighbor UAVs. The master device is then controlled by means of:

$$\boldsymbol{\tau} = -B\dot{\boldsymbol{x}} - K_1\boldsymbol{x} - K_2(\boldsymbol{x} - \bar{\boldsymbol{z}}(k)), \tag{20}$$

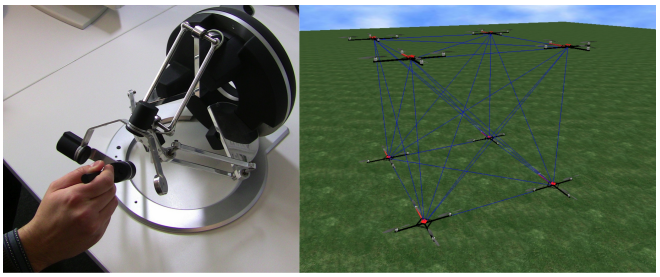


Fig. 1: Human/Hardware in-the-loop simulation setup: human driving a 3DOF haptic device (left) + physical simulation of 8 quadrotors (right).

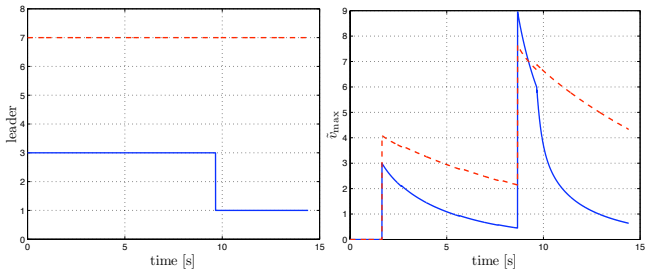
where B is a positive definite damping matrix whose role is to stabilize the master device, K is a diagonal matrix with non-negative entries (possibly all zeros) whose role is to give to the user the perception of the distance from the zero-commanded velocity, and $\bar{\boldsymbol{z}}(k)$ is the passive set-position modulation (PSPM) of $\boldsymbol{z}(t)$. By following the framework proposed in [5], we exploit here the passive set-position modulation (PSPM) algorithm [24] to ensure master passivity [25] w.r.t. the pair (power port) $(\boldsymbol{\tau}, \boldsymbol{z}(t))$ with the control (20). Indeed, the PSPM action can enforce a passive behavior on the master also in presence of delays and packet losses in the communication channel (see [24] for details). This is enough to guarantee a stable interaction with a passive environment such as the human side [26] and our kinematic system, and thus an overall stable teleoperation.

The meaning of the terms $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ is explained as follows. Even though we assumed that a UAV is able to track the velocity of its own kinematic agent with a negligible error, during the transients the UAV velocity will generally not match exactly with the agent velocity because of the reasons reported in Sec. II. The term $\boldsymbol{z}_1$ can then provide the operator with a force cue related to the velocity tracking error of the leader, namely $\boldsymbol{x} - \frac{1}{\rho}\dot{\boldsymbol{q}}_l$. However the contribution of $\boldsymbol{x} - \frac{1}{\rho}\dot{\boldsymbol{q}}_l$ can be appreciated only during the transients and is not informative of the overall multi-agent dynamics. On the other hand the term $\boldsymbol{z}_2$ stabilizes to a constant nonzero value when the commanded velocity is kept constant because of the consensus-like dynamics (5). In particular, this term is related to the position difference between the leader and its neighbors.

V. HUMAN/HARDWARE IN THE LOOP SIMULATIONS WITH A GROUP OF UAVS

We tested the proposed approach using an Omega.6[2] force-feedback device as 3-DOF master robot (see Fig. 1, left). The master controller runs at about 2.5 kHz on a dedicated GNU-Linux machine and communicates via ethernet with one of the UAV flight controllers (FCs) – the leader, which runs on separate GNU-Linux machine. Each FC controls a simulated quadcopter within a custom developed virtual 3D environment (VE) based on the Ogre3D engine[3] for the

[2] http://www.forcedimension.com
[3] http://www.ogre3d.org/



(a) (b)

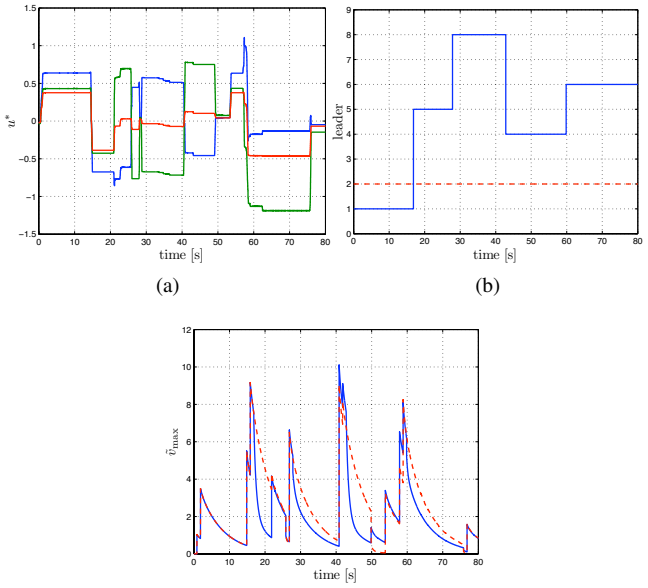
Fig. 2: Result of the first human/hardware in-the-loop simulation, response to full-left/full-right command: (a) Leader selected online (solid line) vs. leader selected a priori (dashed line); (b) Corresponding costs $\tilde{v}_{\max}$ representing the velocity tracking error of the whole team in the two cases, online (solid line) vs. a priori (dashed line).

graphics side and PhysX[4] for the physical simulation of the UAVs (see Fig. 1, right). The FC is able to track a 3D reference trajectory similar to those generated by the multi-agent system on the slave side. This VE simulation runs at 60 Hz on a Windows Machine.

We present two comparative simulations in which we considered a group of 8 UAVs and a sampling time $T = 1$ second for the master-leader communication. A different offset has been added to every agent in order to distribute the UAVs in a collision free formation, following an approach similar to the use of biases in the consensus protocol [27].

In the first simulation (Fig. 2), we compare the online leader selection algorithm to an a-priori leader selection when receiving the same (simple) command sequence: a full-speed command to the left followed by a full-speed command to the right. The online leader selection chooses the 3-rd agent during the first phase and the 1-st agent in the last phase. On the contrary, the a priori selection keeps the 7-th agent as leader during the whole operation (Fig. 2a). In Fig. 2b the costs defined in (13) (denoted here $\tilde{\boldsymbol{v}}_{\max}$ for the sake of brevity) represents the velocity tracking error of the whole team in the two cases. It is evident that the online leader selection outperforms the a priori leader selection. Only for a brief moment during the commutation of the velocity command from left to right, the tracking error of the online leader selection is worse than the a priori one. This behavior is due to the better tracking performance of the online leader selection case. Indeed, these two cases obviously follow two different time histories because of the different choice made for the leader: (i) in the a priori case (dashed line) the slave system reacts more slowly and, when the command switch takes place at time $t \simeq 8$ seconds, the group is still far from having reached the desired velocity, while (ii) during the online case (solid line) the slave system possesses a faster response behavior and, when the command switch takes place, the group is much closer to having reached the desired velocity. Therefore, the change of commanded velocity induces a larger transient in this case,

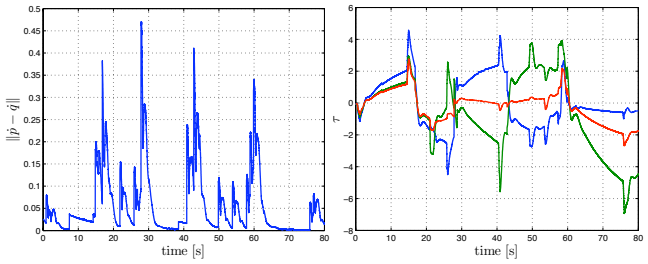
[4] http://www.nvidia.com/object/physx_new.html



(a) (b)

(c)

Fig. 3: Result of the second human/hardware in-the-loop simulation, response to a complex velocity command: (a) The 3 components of the velocity command $\boldsymbol{u}^\star$; (b) Leader selected online (solid line) vs. leader selected a priori (dashed line); (b) Corresponding costs $\tilde{v}_{\max}$ representing the velocity tracking error of the whole team in the two cases, online (solid line) vs. a priori (dashed line).



(a) (b)

Fig. 4: Result of the second human/hardware in-the-loop simulation, response to a complex velocity command: (a) UAV velocity tracking error w.r.t. the agent $\|\dot{\boldsymbol{p}} - \dot{\boldsymbol{q}}\|$; (b) Components of the control force $\boldsymbol{\tau}$ given to the operator.

resulting in the (transitory) higher peak in the $\tilde{v}_{\max}$ index. However, this effect is quickly recovered as it is clear from Fig. 2b.

In the second pair of simulations (Fig. 3 and Fig. 4) we compare the online leader selection algorithm with the a priori selection in the response to a more complex 3D command $\boldsymbol{u}^\star$, whose components are depicted in Fig.3a. Figure 3b shows the online (solid line) and a priori (dashed line) leader selection, and Fig. 3c represent the correspondent costs $\tilde{\boldsymbol{v}}_{max}$ defined in (13). The behavior is similar to the previous experiment. In order to show the tracking capabilities of the UAV flight controller, in Fig. 4a we show the tracking velocity error averaged among the 8 UAVs, and in Fig. 4b we show the 3 components of the force $\boldsymbol{\tau}$ reflected to

the human operator. In this plot we can appreciate two basic features of the haptic cue. First, the presence of peaks due to the velocity tracking error in response to quick changes of the commanded velocity. This cue informs the operator about the inertia of the leader UAV. Second, the fact that the force tends to converge to a steady state value in response to a constant commanded velocity. The absolute amount of this cue informs the operator about the total number of UAVs in the group, in fact the more the UAVs, the more the elongation (for a constant number of leader neighbors). The rate of change of the same cue informs the operator about the agreement of the whole group to the desired velocity. In fact, the closer the rate to zero, the more the group has reached the velocity agreement.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper we have shown how to design a leader-follower control strategy to allow the bilateral teleoperation of a fleet of UAVs by means of a single remote human operator while optimizing online the leader selection in a decentralized way. To this end, we proposed a distributed algorithm which works in conjunction with two distributed estimators. The stability of the overall scheme is preserved ensuring the passivity of the master control which provides a force feedback to the human operator. The presence of this haptic cue informs the user about the salve-side motion status. The approach is validated by means of human/hardware in the loop simulation.

We are currently investigating the possibility of dealing with more leaders at the slave side in order to have a better control of the motion of the UAVs.

REFERENCES

[1] D. Lee and M. W. Spong, "Bilateral teleoperation of multiple cooperative robots over delayed communication network: theory," in *2005 IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain, Apr. 2005, pp. 360–365.

[2] D. Lee, "Semi-autonomous teleoperation of multiple wheeled mobile robots over the internet," in *2008 ASME Dynamic Systems and Control Conference*, Ann Arbor, MI, Oct. 2008.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] A. Howard, L. E. Parker, and G. S. Sukhatme, "Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection," *International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 431–447, 2006.

[10] M. Schwager, B. J. Julian, and D. Rus, "Optimal coverage for multiple hovering robots with downward facing cameras," in *2009 IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, May 2009, pp. 3515–3522.

[11] 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, 2010.

[12] B. Hannaford, "Stability and performance tradeoffs in bi-lateral telemanipulation," in *1989 IEEE Int. Conf. on Robotics and Automation*, Scottsdale, AZ, May 1989, pp. 1764–1767.

[13] N. A. Lynch, *Distributed Algorithms*. Morgan Kaufmann, 1997.

[14] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*, ser. Applied Mathematics Series. Princeton University Press, 2009.

[15] I. Shames, A. M. H. Teixeira, H. Sandberg, and K. H. Johansson, "Distributed leader selection without direct inter-agent communication," in *2nd IFAC Work. on Estimation and Control of Networked Systems*, Annecy, France, Sep. 2010, pp. 221–226.

[16] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro," in *2005 IEEE Int. Conf. on Robotics and Automation*, May 2005, pp. 2247–2252.

[17] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Trans. on Robotics*, vol. 24, no. 2, pp. 331–340, 2008.

[18] A.Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 162–186, 2009.

[19] P. Twu, M. Egerstedt, and S. Martini, "Controllability of homogeneous single-leader networks," in *49th IEEE Conf. on Decision and Control*, Atlanta, GA, Dec. 2010, pp. 5869–5874.

[20] W. Ren and Y. Cao, *Distributed Coordination of Multi-agent Networks: Emergent Problems, Models, and Issues*. Springer, 2010.

[21] Y. Cao and W. Ren, "Distributed coordinated tracking with reduced interaction via a variable structure approach," *to appear in IEEE Trans. on Automatic Control*, 2011.

[22] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks," *Automatica*, vol. 46, no. 2, pp. 390–396, 2010.

[23] R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *45th IEEE Conf. on Decision and Control*, San Diego, CA, Jan. 2006, pp. 338–343.

[24] 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.

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

[26] N. Hogan, "Controlling impedance at the man/machine," in *1989 IEEE Int. Conf. on Robotics and Automation*, Scottsdale, AZ, May 1989, pp. 1626–1631.

[27] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215 –233, 2007.