Decentralized Bilateral Aerial Teleoperation of Multiple UAVs - Part II: a Top-down Perspective

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

***Abstract*—In this notes, we concisely review some recent advancements in the field of Aerial Teleoperation, i.e., how to bilaterally couple a single human operator with a remote fleet of semi-autonomous UAVs which 1) must keep a desired formation and avoid obstacle- collisions, and 2) must collectively follow the human commands. While Part I presented the bottom-up approach, in which the group behaves as a fluid formation, in this Part II we briefly summarize the top-down approach, where the group is abstracted as a semi-rigid body, and is subject of reversible deformation when interacting with the environment.**

I. Introduction

Compared to a unique and complex robot, a group of simple mobile robots is more suitable and robust in addressing task requiring pervasiveness, ubiquity and flexibility like exploration [1], coverage [2], cooperative transportation [3], formation control [4], distributed estimation [5] and sensing [6, 7]. In almost all those applications, man's intelligence may be profitably used online when high-level cognitive-based decisions are needed (e.g., during navigation in dynamic and crowded environments or whenever critical decision has to be taken based on a few noisy visual elements). Our belief is that *bilateral* teleoperation, where the human receives also a force feedback, is one of the most valid tools to "plug" the human in the loop of the multi-robot system. In fact, it implements a human-multi-robot synergy where the operator is able to partially control some aspects of the group motion while experiencing an improved perception of the remote environment through the haptic feedback.

Apart from the pure robotic perspective the study of this kind of systems constitutes also a novel and challenging topic in the broad field of human–machine interfaces and telepresence applications. Indeed a remote group of mobile robots may be used, in the near future, in order to enhance the human perceptions and actions allowing rapid and precise operations at the macroscopic, microscopic, and planetary scales.

In these notes we presents, in an organic and unifying perspective, some recent bilateral teleoperation systems establishing the remote control of groups of mobile robots for *navigation* purposes [8, 9, 10]. Multi-robot navigation, defined here as the safe and cohesive motion in a obstacle-populated environments, is the indispensable premise for any other specific objective in mobile robots, such as exploration, localization, transport, or manipulation. The remote mobile robots (the *slave-side* from now on) possess some level of local autonomy and act as a *group* by achieving some desired shape and avoiding collisions by means of decentralized controllers. At the same time, the human operator, acting on the *master device*, is in control of the overall group motion and receives, through haptic feedback, suitable cues informative enough of the remote robot/environment state. On top of the remote navigation layer, the group is still allowed to perform additional local tasks by exploiting the internal slave side redundancy w.r.t. the master device commands. The proposed approaches are suitable for an heterogeneous group of robots, comprising aerial, ground, space, naval, or underwater vehicles.

Bilateral teleoperation of (multiple) mobile robots presents several differences w.r.t. conventional teleoperation systems: first, there exists a structural *kinematic dissimilarity* between master and slave sides, i.e., the master possesses a limited workspace while the slave an unbounded one.

Second, in the navigation scenario, physical contact with the environment must be avoided by the robots. Therefore, some virtual forces must be fabricated for the operator in place of the of real interaction forces. To some extent, this implies a redefinition of standard concepts related to telepresence in order to properly assess the human operator immersiveness. We refer the reader to [11, 12] for some preliminary studies which are exploring the human perception point of view in these uncommon teleoperation scenarios.

Lastly, the slave-side possesses large motion redundancy w.r.t. the master-side because of the mismatch between the degrees of freedom (DOFs) of the master (usually in the range of 3–6), and the DOFs of the slave (in the range of $6N$ for N robots, when considered as rigid bodies).

This Part II briefly reviews the *top-down* approach which has been developed in [8, 9, 10]. While in bottom-up approach the group behaves as a fluid formation, in the top-down the group is abstracted as a semi-rigid body, and is subject of reversible deformation when interacting with the environment. The ideas behind them are briefly illustrated by Fig. 1(a): flexibility of the fleet is traded for a more rigid and predefined shape of the group, but still allowing local *reversible* deformations because of the interaction with the environment (obstacles). Imposing this kind of behavior to the slave-side finds applications in all those scenarios where maintaining a desired shape is mandatory, such as distributed monitoring, optimal placement for communication/exploration, and so on.

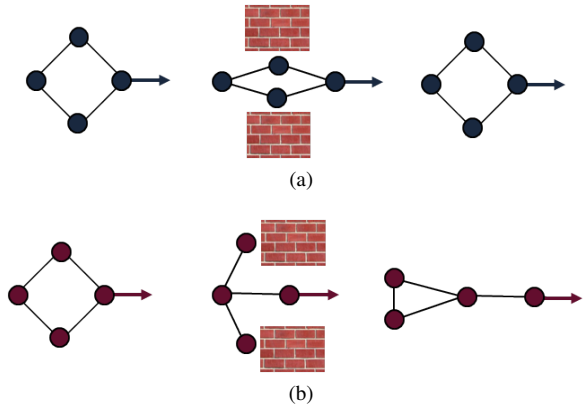

Fig. 1: Top: an illustrative picture of the *top-down* approach. Bottom: an illustrative picture of the *bottom-up* approach

Part I [13] deals with the *bottom-up* approach, which was developed in [14, 15, 16]. Figure 1(b) depicts an illustrative example of this scenario: a group of 4 agents splits some 'interaction links' in the proximity of an obstacle to actively reshape the formation and get over the narrow passage. Such flexible behavior was achieved by only requiring 1-hop information within the group, and a single communication link between the human operator and one of the robots (which was then referred to as the 'unique' leader)

These notes have only an introductory purpose and are written in a informal style. We refer the interested reader to [8, 9, 10] for a detailed description of the methods, the proofs, and for the experimental results.

II. Slave Side Model

The slave-side of the proposed teleoperation system is composed of N robots, modeled as rigid bodies in space. We assume that that the i-th robot is endowed with a proper controller which can track (with a small/negligble error) any smooth trajectory of the output $\boldsymbol{q}_i^\star = (\boldsymbol{p}_i^\star, \theta_i^\star) \in \mathbb{R}^3 \times \mathbb{S}^1$, where $\mathbb{S}^1$ is the unit circle, and $\boldsymbol{p}_i^\star$, $\theta_i^\star$ represent, respectively, the *position* of a particular point[1] of the robot in world frame $\mathcal{W}$ and the rotation around the vertical axis (z) of $\mathcal{W}$ (*yaw*).

Notice that this is the case, for example of all the robots possessing a flat output [17] of the aforementioned form. Several robots, which are commonly used in the field, possess a flat output, e.g., the position of the center of mass and the yaw angle of the quadrotor. Many UAV tracking controllers proposed in the literature (e.g., see [18, 19, 20] for the quadrotor) can be used for the tracking purpose.

In order to generate a reference trajectory for the output of the robot, namely $\boldsymbol{q}(t) = \boldsymbol{p}_i(t), \theta_i(t)$, we design a controller which selects online the linear velocity $\boldsymbol{u}_i$ and yaw-rate ω_i of the following kinematic *agent*:

$$\dot{\boldsymbol{q}} = \begin{pmatrix} \dot{\boldsymbol{p}}_i \\ \dot{\theta}_i \end{pmatrix} = \begin{pmatrix} R_i & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_i \\ \omega_i \end{pmatrix}, \tag{1}$$

[1]Typically the center of mass or the geometric center.

where $\mathbf{0} = (0\,0\,0)^T$, and matrix R_i represents a rotation of angle θ_i around the z axis. We denote the inverse/transpose of R_i with ${}^iR = R_i^T$, and the rotation between two body frames with ${}^iR_j = {}^iR R_j$.

A *bearing* is a unit vector in $\mathbb{R}^3$, i.e., a point in $\mathbb{S}^2$, the unit sphere. The *relative bearing* between the i-th and the j-th agent, as seen in the body frame of the i-th agent is defined as:

$${}^i\boldsymbol{\beta}_{ij} = {}^iR\,\boldsymbol{p}_{ij}/\delta_{ij}, \tag{2}$$

where $\boldsymbol{p}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i$, and

$$\delta_{ij} = \|\boldsymbol{p}_j - \boldsymbol{p}_i\| \tag{3}$$

is the *inter-distance* between agent i and j.

We assume the availability of relative bearings and inter-distances among the agents, when needed. In a real situation those measurements can be obtained by post-processing the measurement of a sensor attached to the robot, e.g., by adding a correction term related to the displacement between the real robot and the agent (i.e., the desired trajectory point). This represent a feasible assumption as long as the discrepancy between the desired and actual output of the robot is small, as requested in the previous assumption. In addition, in order to compute the inter-distances and relative bearings between the agents, some additional measures may be needed depending on the mechanics of the actual robots. For example, the availability of ${}^i\boldsymbol{\beta}_{ij}$ relies on assumption that the agents have a common knowledge of the vertical (z) direction, which stem from practical foundation. In fact in the real world, even if no global positioning system is given, measurement of the direction of gravity is quite reliable and available everywhere by means of an accelerometer, from which the UAV can obtain roll and pitch angles.

Finally, we assume that the robots are endowed with an obstacle detector which can measure the position of the obstacle points which are distant less than $D^o > 0$ from $\boldsymbol{p}_i^\star$, and therefore from the correspondent agent position $\boldsymbol{p}_i$.

III. Control of the Reference Trajectory on the Slave Side

The motion of the robots is controlled by assigning them (online) a certain reference trajectory generated by a proper design of the agent velocities $\boldsymbol{u}_i, \omega_i$ in (1). In the presented teleoperation system, the agent velocities are composed by 3 terms, which correspond to the main objective of the multi-robot tele-navigation task:

$$(\boldsymbol{u}_i, \omega_i) = (\boldsymbol{u}_i^f, \omega_i^f) + (\boldsymbol{u}_i^m, \omega_i^m) + (\boldsymbol{u}_i^o, 0). \tag{4}$$

The term $(\boldsymbol{u}_i^f, \omega_i^f)$ is used in order to stabilize the group on a desired formation, the term $(\boldsymbol{u}_i^m, \omega_i^m)$ allows the human operator to control the overall behavior of the formation, and $(\boldsymbol{u}_i^o, 0)$ is an action aimed at preventing collisions with the surrounding obstacles.

These 3 terms are designed in order to let the group of agents (i.e., the reference trajectories) behave as a reversible semi-rigid structure of points which can be moved as a single

body by the human operator, deforms during the interaction with the environment, and returns back to the original shape in free space. In the next Section we will detail each control objective and its related control term.

A. Formation Control Term

The first objective of the agents is to autonomously keep a desired *constrained formation*. A single formation is a vector of agent configurations $\boldsymbol{q} = (\boldsymbol{q}_1, \dots, \boldsymbol{q}_N) \in (\mathbb{R}^3 \times S^1)^N$. A *constrained-formation* is a set of formations which is specified by requiring that some mutual quantities between the agents maintain certain desired fixed values. In other words a constrained formation is the equivalence class of all the formations *realizing* the desired values for the constrained quantities.

In these notes we will consider two kind of constrained formations: *distance-* and *bearing-formations*. A distance-formation (resp. bearing-formation) is specified by constraining to some definite fixed values a certain set of inter-distances (resp. relative bearings) between the agents, i.e., a distance-formation is the equivalence class of all the formations *realizing* the same inter-distances (resp. relative bearings).

We denote with $\mathcal{N}$ the set $\{(i,j) \in \{1,\dots,N\}^2 | i \neq j\}$. A set of $N(N-1)$ inter-distances $\{{}^i\delta_{ij} \in \mathbb{R}^+\}_{(i,j)\in\mathcal{N}}$ (resp. relative bearings $\{{}^i\boldsymbol{\beta}_{ij} \in \mathbb{S}^2\}_{(i,j)\in\mathcal{N}}$) is *feasible* if there exists a formation realizing them (henceforth called a realization). Finally we denote with $\mathcal{V}$ the set $\{1,\dots,N\}$.

Remembering that a constrained-formation is an equivalence class of formations we say that a distance-formation (resp. bearing formation) is *rigid* if all its formations have the same inter-distances (resp. relative bearings) between all the agents (which means not only the ones specifying the formation). In other words, a distance-formation (resp. bearing-formation) is rigid if it specifies in a unique way all the other (non-determined) inter-distances (resp. relative bearings) of the formation. In these notes we will always assume that the desired constrained-formation are feasible and rigid.

We present now separately the formation controllers in the case of distance- and bearing-formation.

1) Distance-formation case: Assume that a feasible and rigid distance-formation is given in the form of a set of desired inter-distances d_{ij} where $(i,j) \in \mathcal{E}^d \in \mathcal{V} \times \mathcal{V}$ and let $\mathcal{N}_i^d = \{j \mid (j,i) \in \mathcal{E}^d\}$. The used formation controller is defined by:

$$\boldsymbol{u}_i^f := -\sum_{j\in\mathcal{N}_i^d} \frac{\partial \varphi^f(||\boldsymbol{p}_i - \boldsymbol{p}_j||^2, d_{ij})^T}{\partial \boldsymbol{p}_i}, \quad \omega_i^f = 0 \tag{5}$$

where φ^f is a certain artificial potential function which creates attractive action if $||\boldsymbol{p}_i - \boldsymbol{p}_j||$ is large, repulsive action if $||\boldsymbol{p}_i - \boldsymbol{p}_j||$ is small, and has a unique minimum in d_{ij}. Figure 2 depicts a possible design for φ^f.

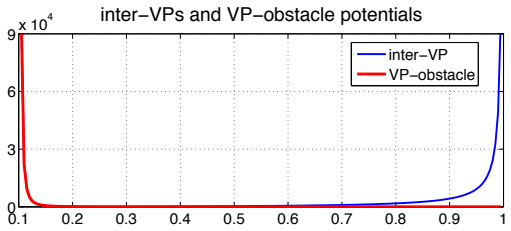

Fig. 2: Examples of φ^f.

2) Bearing-formation case: Assume that a feasible bearing-formation is given in the form of a set of desired relative bearings $\{{}^i\boldsymbol{b}_{ij} \equiv ({}^ia_{ij}, {}^ie_{ij})\}_{(i,j)\in\mathcal{N}}$. We use the polar parametrization of a relative bearing ${}^i\boldsymbol{\beta}_{ij}$ in terms of elevation ${}^i\eta_{ij} \in [-\pi/2\ \pi/2]$ and azimuthal angle ${}^i\alpha_{ij} \in (-\pi\ \pi]$ defined by

$${}^i\boldsymbol{\beta}_{ij} = \begin{pmatrix}\cos{}^i\eta_{ij}\cos{}^i\alpha_{ij} & \cos{}^i\eta_{ij}\sin{}^i\alpha_{ij} & \sin{}^i\eta_{ij}\end{pmatrix}^T, \tag{6}$$

denoting in brief ${}^i\boldsymbol{\beta}_{ij} \equiv ({}^i\alpha_{ij}, {}^i\eta_{ij})$. For any i,j,k we define $\hat{\boldsymbol{p}}_{ij} = \frac{\boldsymbol{p}_{ij}}{\delta_{ij}}$, $\gamma_{ijk} = \frac{\delta_{ik}}{\delta_{ij}}$, and $M = \begin{bmatrix} M' & \boldsymbol{0} \\ \boldsymbol{0}^T & 0 \end{bmatrix}$, with $M' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. The proposed controller is:

$$\boldsymbol{u}_1^f = \boldsymbol{0} \qquad \omega_1^f = 0 \tag{7}$$

$$\boldsymbol{u}_2^f = -\frac{K_p}{\cos{}^1\eta_{12}}\Big[\sin({}^1\alpha_{12} - {}^1a_{12}) M\, {}^1\boldsymbol{\beta}_{12} + \big(\big(\sec{}^1\eta_{12}\, {}^1\boldsymbol{\beta}_{12} - \sec{}^1e_{12}\, {}^1\boldsymbol{b}_{12}\big)\cdot\hat{\boldsymbol{z}}\big)\hat{\boldsymbol{z}}\Big] \tag{8}$$

$$\omega_2^f = K_\omega \sin({}^2\alpha_{21} - {}^2a_{21}) \tag{9}$$

$$\boldsymbol{u}_i^f = -K_p\, {}^iR_1\, \big(\bar{\delta}_{1i}\, {}^1\boldsymbol{\beta}_{1i} - \bar{d}_{1i}\, {}^1\boldsymbol{b}_{1i}\big) \tag{10}$$

$$\omega_i^f = \begin{cases} K_\omega \sin({}^i\alpha_{i1} - {}^ia_{i1}) & \text{if } \cos{}^ie_{i1} \neq 0 \\ K_\omega \sin({}^i\alpha_{i2} - {}^ia_{i2}) & \text{otherwise} \end{cases} \tag{11}$$

where $i = 3,\dots,N$, $\hat{\boldsymbol{z}} = (0\,0\,1)^T$, $\bar{\delta}_{1i} = \gamma_{12i}\sec({}^1\eta_{12})$, δ_{12}^0 is the initial inter-distance between agents 1 and 2, $\bar{d}_{1i} = \frac{\|{}^2\boldsymbol{b}_{21}\times{}^2\boldsymbol{b}_{2i}\|}{\|{}^i\boldsymbol{b}_{i1}\times{}^i\boldsymbol{b}_{i2}\|}\sec({}^1e_{12}) = \frac{d_{1i}}{d_{12}}\sec({}^1e_{12})$, ${}^i\boldsymbol{b}_{ij} \equiv ({}^ia_{ij}, {}^ie_{ij})$, K_p, K_ω are positive gains, and ${}^iR_1$ can be computed as $R^T(\alpha_{i1})R(\pi)R(\alpha_{1i})$, denoting by $R(*)$ the rotation matrix of a given angle around $\hat{\boldsymbol{z}}$.

B. Group Steering

The second objective of the agents is to achieve a coherent motion while following the motion commands of the human. The allowed human commands are chosen in order to be not in conflict with the formation control commands. In order to achieve this orthogonality property the allowed motions must belong to the tangent bundle of the manifold defined by the formation constraints. We refer to them as *constraint-invariant movements*. The constraint-invariant movements of a rigid distance-formation are:

1) cohesive translation of all the agents
2) cohesive rotation of all the agents around any axis
3) any different yaw rate for every agent.

On the other side, the constraint-invariant movements of a rigid bearing-formation are:

1) cohesive translation of all the agents
2) cohesive rotation of all the agents around the first agent
3) cohesive expansion/contraction of all the agents.

If the agents 1 and 2 are not aligned with any other agent in the bearing-formation the latter are given by the following distribution

$$(\dot{\boldsymbol{p}}_h, \dot{\theta}_h) = ((v_x\, v_y\, v_z)^T, 0) + \omega(-M\boldsymbol{p}_{1h}, -1) + \lambda(\gamma_{12h}\hat{\boldsymbol{p}}_{1h}, 0),$$

for any $h = 1, \ldots, N$, $v_x, v_y, v_z, \omega, \lambda \in \mathbb{R}$, where we assumed $\hat{\boldsymbol{p}}_{hh} = \mathbf{0}$ by convention.

The terms v_x, v_y, v_z represent a uniform translation in any direction, ω a synchronized rotation around the vertical axis passing through the agent 1, and λ an isotropic dilation/contraction centered on the agent 1. By properly combine rotation (dilation) with translation we can achieve a rotation around any vertical axis, and a dilation w.r.t any point. For instance, setting $\lambda = 0$ and $(v_x\, v_y\, v_z)^T = M\boldsymbol{p}_{13}$ generates a rotation around the 3-rd agent.

Note that knowledge of relative-bearings (and not of inter-distances) is sufficient to perform the synchronized dilation. In fact, $\gamma_{iji} = 0$, $\gamma_{ijj} = 1$, and, using cross-products, we have, $\forall k \neq i, j$ $\gamma_{ijk} = \frac{\|^j\boldsymbol{\beta}_{ji} \times {}^j\boldsymbol{\beta}_{jk}\|}{\|^k\boldsymbol{\beta}_{ki} \times {}^k\boldsymbol{\beta}_{kj}\|}$. On the other hand, knowledge of inter-distances is needed to perform the synchronized rotation.

In these notes we present the situation where the human is allowed to control only the translation and the expansion rate. The latter control will be present only for the bearing-formation case. In particular we use the following controller:

$$\boldsymbol{u}_i^m = {}^iR\boldsymbol{\nu}^t - r\gamma_{12i}\boldsymbol{\beta}_{i1}, \quad \omega_i^m = 0 \tag{12}$$

The human operator can control the translation and the rate of expansion of the whole formation by manipulating $\boldsymbol{\nu}^t$ and r respectively.

C. Obstacle Avoidance Action

The obstacle avoidance action as given by

$$\boldsymbol{u}_i^o := -\sum_{\boldsymbol{p}_r^o \in \mathcal{O}_i} \frac{\partial \varphi^o(\|\boldsymbol{p}_i - \boldsymbol{p}_r^o\|)^T}{\partial \boldsymbol{p}_i} \tag{13}$$

where $\mathcal{O}_i$ is the set of obstacles points measured by the obstacle detector, and φ^o is a certain artificial potential, which produces repulsive action if $\|\boldsymbol{p}_i - \boldsymbol{p}_r^o\|$ is small, smoothly converges to zero as $\|\boldsymbol{p}_i - \boldsymbol{p}_r^o\| \to D^o$, and stays zero with $\|\boldsymbol{p}_i - \boldsymbol{p}_r^o\| \geq D^o$, to make the effect of obstacles for each agent emerge/disappear when they move closer/farther from the agent.

The obstacle avoidance potential can be tuned in order to ensure that the repulsive action goes to infinity as $\|\boldsymbol{p}_i - \boldsymbol{p}_r^o\| \to D^r$ where D^r is the minimum radius such as the ball $B_{D^r}(\boldsymbol{p}_i^\star)$ contains the whole robot. The red plot in Fig. 2 depicts a possible design for φ^o.

IV. Master Side and Force Feedback

We use a 3DOF and a 1DOF force feedback devices in order to control translation and expansion rates of the agent formation. The 3DOF haptic device is modeled as

$$M(\boldsymbol{x}_t)\ddot{\boldsymbol{x}}_t + C(\boldsymbol{x}_t, \dot{\boldsymbol{x}}_t)\dot{\boldsymbol{x}}_t = \boldsymbol{\tau}_t + \boldsymbol{f}_t \tag{14}$$

where $\boldsymbol{x}_t \in \mathbb{R}^3$ is the configuration, $M(\boldsymbol{x}_t) \in \mathbb{R}^{3\times3}$ is the positive-definite/symmetric inertia matrix, $C(\boldsymbol{x}_t, \dot{\boldsymbol{x}}_t) \in \mathbb{R}^{3\times3}$ is the Coriolis matrix, and $\boldsymbol{\tau}_t, \boldsymbol{f}_t \in \mathbb{R}^3$ are the control and human forces, respectively. The 1DOF device is modeled as

$$m\ddot{x}_r = \tau_r + f_r \tag{15}$$

where $x_r \in \mathbb{R}$ is the position, $m \in \mathbb{R}^+$ is the mass, and $\tau_r, f_r \in \mathbb{R}^3$ are the control and human forces, respectively.

After having chosen the beacon agents 1 and 2, the tele-control is implemented by setting in (12)

$$\boldsymbol{\nu}^t = \lambda_t \boldsymbol{x}_t, \quad r = \lambda_r x_r, \tag{16}$$

where $\lambda_t > 0$ and $\lambda_r > 0$ are used suitable scaling factors from $(\boldsymbol{x}_t,\, x_r)$ to the desired agent velocities. Therefore the velocity commanded by the master to the i-th agent, in its local frame, results in:

$$\boldsymbol{v}_i^m = \lambda_t {}^iR\boldsymbol{x}_t - \lambda_r x_r \gamma_{12i}\boldsymbol{\beta}_{i1}. \tag{17}$$

which can be computed by the i-th agent using only local measurements and the measurements from agents 1 and 2 by means of local communication. Notice that in case of distance-formations only the 3DOF device is used and we can set $r = 0$.

The UAVs are assumed to track the i-th agent velocity with sufficient precision. However, during the transients, the UAV *actual velocity* $\dot{\boldsymbol{q}}_i^\star$ will not track exactly the agent velocity $\dot{\boldsymbol{q}}_i$. In order to implement the tele-sensing, we provide the operator with two haptic cues proportional to the translation-velocity and expansion-speed tracking errors respectively, defined as

$$\boldsymbol{e}_t = \boldsymbol{x}_t - \boldsymbol{z}_t(t) \qquad e_r = x_r - z_r(t) \tag{18}$$

$$\boldsymbol{z}_t = \frac{1}{\lambda_t N}\sum_{i=1}^{N}(\lambda_r x_r \gamma_{12i} R_i \boldsymbol{\beta}_{i1} + R_i {}^i\dot{\boldsymbol{q}}^\star) \tag{19}$$

$$z_r = \frac{1}{\gamma_{12i}\lambda_r N}\sum_{i=1}^{N} {}^i\dot{\boldsymbol{q}}^\star \cdot \boldsymbol{\beta}_{i1} \tag{20}$$

The i-th UAV sends to the master device its current velocity in body frame ${}^i\dot{\boldsymbol{q}}_i^\star[k] = {}^iR\dot{\boldsymbol{q}}_i^\star[k]$, where the symbol $[k]$ indicates that the signal is received, sampled and discretized over the master-slave communication channel. The master controller uses all the received velocities in order to compute $\boldsymbol{z}_t[k]$ and $z_r[k]$, and implements the teleoperation controls as

$$\boldsymbol{\tau}_t = -B_t\dot{\boldsymbol{x}}_t - K_t\boldsymbol{x}_t - K_t^\star(\boldsymbol{x}_t - \bar{\boldsymbol{z}}_t[k]) \tag{21}$$

$$\tau_r = -B_r\dot{x}_r - K_r x_r - K_r^\star(x_r - \bar{z}_r[k]) \tag{22}$$

where B_t, B_r are a positive definite damping matrix whose role is to stabilize the master devices, K_t, K_r are 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 $\bar{\boldsymbol{z}}_t(k)$, $\bar{z}_r(k)$ are the passive set-position modulation (PSPM) versions of $\boldsymbol{z}_t(t)$ and $z_r(t)$ respectively. By following the framework proposed in [8], we exploit here the PSPM algorithm [21] to ensure master passivity [22] w.r.t. the pairs (power ports) $(\boldsymbol{\tau}_t, \boldsymbol{z}_t)$ and (τ_r, z_r) with the control (21–22). 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 [21] for details). This is sufficient to guarantee a stable interaction with a passive environment such as the human side [23] and our kinematic system, and thus an overall stable teleoperation.

V. Human/Hardware in-the-Loop Simulations for the Distance-Formation Case

We used an Omega6® (Force Dimension) as a master derive with actuated linear 3-DOF and un-actuated rotational 3-DOF. Its local servo-loop runs at about 2.5 kHz on a Linux machine. A simulation environment was constructed to simulate UAVs dynamics and their control laws, by using Ogre3D engine (http://www.ogre3d.org/ for 3D rendering and computational geometry) and PhysX (http://www.nvidia.com/object/physx_new.html for simulating physical interaction). See Fig. 3. Our simulation runs at 60 Hz, creating asynchronous data update between the master device (i.e. 2.5kHz) and the slave UAVs (i.e. 60Hz), which produces some similarity with the Internet communication.

For the hardware/human in-the-loop simulation, we simulated $N = 8$ UAVs with all-to-all communication (complete graph G), and set the 28 inter-VP potentials φ_{ij}^c so as to realize a cubic shape with edge, edge diagonal, and cube diagonal measuring 3.3 [m], 4.67 [m], and 5.7 [m], as shown in Fig. 3.

During the simulation, the user flew the UAVs in the simulated environment (with two walls installed at the left and right sides) twice from the side to side with almost a constant velocity, thus alternating 4 steady-state collective motion conditions, with 4 hard contacts with the walls (treated as obstacles). Behavior of the inter-VP distances is shown in Fig. 4, where we can see that: the UAVs start from the undeformed cubic shape until about $t_1 = 1.9$ [s] when some of them get close to the first wall and their cubic shape deforms accordingly. Similar behaviors repeat after $t_2 = 5$ [s] when the UAVs move away from the wall and head towards the other wall (around $t_3 = 7.5$ [s]).

Fig. 5 contains the master position q (solid line) and the agents' centroid velocity $\frac{1}{\lambda N}\sum_{i=1}^{N}\dot{p}_i$ (dashed line), showing a good teleoperation tracking performance when away from the walls. Fig. 6, on the other hand, depicts the master control force τ (top plot) and the average obstacle avoidance actions $\frac{1}{\lambda N}\sum_{i=1}^{N}u_i^o$ (bottom plot), showing a good haptic perception of the walls (i.e., obstacles). Finally, Fig. 7 (top) shows the UAV-VP tracking error $\|p_1 - x_1\|$ for the 1th UAV (similar plots for other UAVs omitted), and Fig. 7 (bottom) the behavior of the PSPM virtual energy reservoir $E(k)$, which does not deplete (i.e., correct reproduction of the signal y) and drops

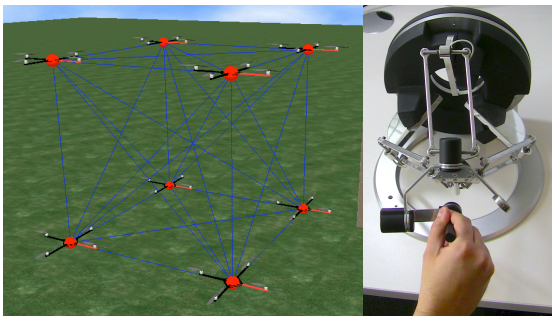

Fig. 3: UAVs in 3D simulation environment and master haptic device.

to produce the obstacle haptic perception when the UAVs get close to the walls.

From these plots, we can then observe that:

1) stability of the closed-loop teleoperation system even with the asynchronous master-slave data update;
2) good teleoperation performance, with haptic perception of obstacles (Fig. 6) and human velocity command and UAVs' centroid velocity coordination (Fig. 5); and
3) good UAV-VP tracking performance as shown by the small $||p_1 - x_1||$ in Fig. 7.

VI. Experiments for the Bearing-Formation Case

A. Bearing Formations with Limited FOV

Camera hardware suitable for the use on small flying robots usually suffers from a limited field of view (FOV). Furthermore, low on board processing power does not allow for a parallel use of multi cameras. Thus, in contrast to [9], we have to deal with a limited FOV for real robot experiments. In order to compensate for a vertical (resp. horizontal) limited FOV two strategies apply: (1) shifting vertically (resp. horizontally) the agent or (2) rotating the agent by changing its roll/pitch (resp. yaw) angles. We opt for the the rotation strategy since it is fastest and preserves mutual positions, i.e. the shape of the group.

In particular, in this notes we focus on limited horizontal FOV only, mainly due to two reasons. First, we will primarily test formations with a horizontal dominant dimension, which, in practice, are perhaps the most interesting ones. Lastly, since quadrotors are under-actuated, it is unfeasible to change their roll/pitch independently from the horizontal speed. However, the proposed approach can be easily extended to a limited vertical FOV provided that the camera is mounted on an additional tilting unit. Therefore, henceforth we assume the agent i to be measuring the azimuth angle ${}^i\alpha_{ij}$, if and only if ${}^i\alpha_{ij} \in [a_{\min}, a_{\max}]$.

We will first introduce a relaxed definition of a bearing formation and then propose a controller based on this formulation, which overcomes the limitation on the horizontal FOV.

We introduce the following relaxed bearing formation definition: two bearing formations $\{({}^i\alpha_{ij}, {}^i\eta_{ij})\}_{(i,j)\in\mathcal{N}}$ and $\{({}^i\alpha'_{ij}, {}^i\eta'_{ij})\}_{(i,j)\in\mathcal{N}}$ are equivalent if the shift ${}^i\alpha'_{ij} - {}^i\alpha_{ij}$ is

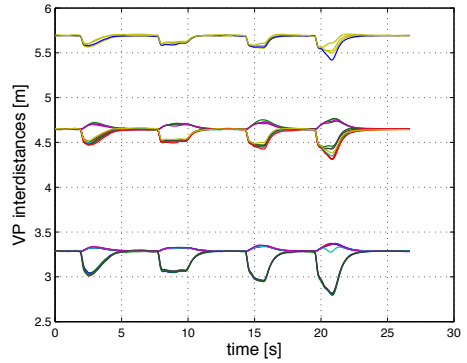

Fig. 4: Inter-VP distance: steady-state values corresponding to the un-deformed N-VPs flying object shape.

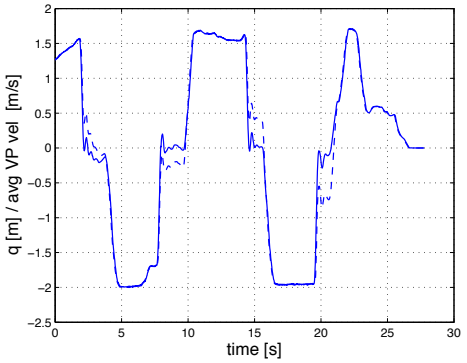

Fig. 5: Tele-coordination of $q(t)$ and $\frac{1}{\lambda N}\sum_{i=1}^{N}\dot{p}_i(t)$.

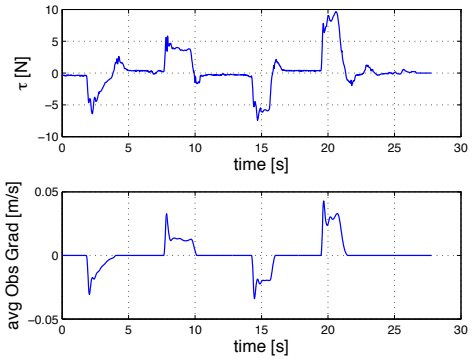

Fig. 6: Haptic perception of obstacles (i.e., $\frac{1}{\lambda N}\sum_{i=1}^{N}u_i^o(t)$) via $\tau(t)$.

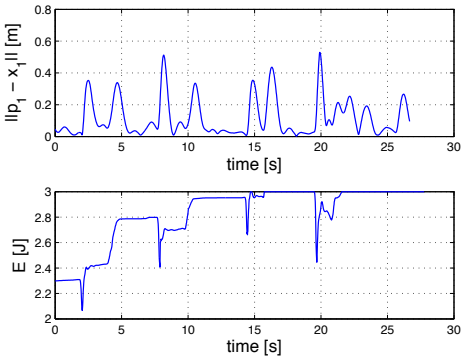

Fig. 7: UAV-VP tracking error $\|p_1(t)-x_1(t)\|$ and $E(k)$: note that small tracking error is maintained with $E(k)$ never depleting.

the same for every $(i,j)\in\mathcal{N}$, i.e., if the difference between any two azimuths is constant

$$ {}^i\alpha_{ij} - {}^i\alpha_{ik} = {}^i\alpha'_{ij} - {}^i\alpha'_{ik} \quad \forall (i,j),(i,k)\in\mathcal{N}. \tag{23}$$

The rotation speed ω_i of the agent i does not affect the difference in (23). In fact, from the dynamic equation of the azimuth we have:

$$ {}^i\dot{\alpha}_{ij} - {}^i\dot{\alpha}_{ik} = \frac{1}{\delta_{ij}}({}^j\boldsymbol{\alpha}_{ji}^T\boldsymbol{u}_j - {}^i\boldsymbol{\alpha}_{ij}^T\boldsymbol{u}_i) - \frac{1}{\delta_{ik}}({}^k\boldsymbol{\alpha}_{ki}^T\boldsymbol{u}_k - {}^i\boldsymbol{\alpha}_{ik}^T\boldsymbol{u}_i). \tag{24}$$

Therefore the rotation dynamic of an agent can be freely chosen without affecting the relaxed bearing formation.

B. Formation Control with Limited FOV

In order to deal with the limited FOV in the horizontal plane, we ask the agents to follow an opportunely shaped bearing trajectory, i.e., any agent $h\in(1,\dots,N)$ will be forced to rotate with a given yaw rate $\omega_{\mathrm{rot},h}(t)$ in order to execute a periodic motion (e.g., a sinusoidal or a constant-slope trajectory). By suitably choosing $\omega_{\mathrm{rot},h}(t)$ and if the bearing formation is maintained, agent h would be able to periodically measure the relative azimuth ${}^h\alpha_{hj}$ of another agent j for a fraction of the trajectory period. For the remaining fraction, a direct measure of the azimuth is not given. Hence it must be estimated on the basis of motion proprioception.

We define the modified version of the bearing formation control problem, which accounts for the horizontally limited FOV.

Problem 1 (Relaxed bearing-formation control): Given a set of feasible desired bearings $\{{}^i\boldsymbol{b}_{ij}=({}^ia_{ij},{}^ie_{ij})\}_{(i,j)\in\mathcal{N}}$, find a control law $(\boldsymbol{u}_i^f,\omega_i^f)$ depending on $\{{}^i\boldsymbol{\beta}_{ij}\}_{(i,j)\in\mathcal{N}}$ which steers ${}^i\boldsymbol{\beta}_{ij}$ to the trajectory ${}^i\tilde{\boldsymbol{b}}_{ij}(t)=({}^ia_{ij}+\theta_{\mathrm{rot},i}(t),{}^ie_{ij})$ with $\theta_{\mathrm{rot},i}(t)=\int_t\omega_{\mathrm{rot},i}(t)\mathrm{d}t$ and the distances $\{\delta_{ij}\}_{(i,j)\in\mathcal{N}}$ to a constant non-zero value, $\forall(i,j)\in\mathcal{N}$.

First we present the control law used to solve Prob. 1 as if the azimuth is always measured. After we will describe the estimate that we used in the real case. Let us consider the following control law:

$$\boldsymbol{u}_1^f = \boldsymbol{0} \tag{25}$$

$$\omega_1^f = -\omega_{\mathrm{rot},1} \tag{26}$$

$$\boldsymbol{u}_2^f = -\frac{K_p}{\cos{}^1\eta_{12}}\Big[\sin({}^1\alpha_{12}-\theta_{\mathrm{rot},1}-{}^1a_{12})M\,{}^1\boldsymbol{\beta}_{12} + \big(\sec{}^1\eta_{12}\,{}^1\boldsymbol{\beta}_{12}-\sec{}^1e_{12}\,{}^1\boldsymbol{b}_{12}\big)\cdot\hat{\boldsymbol{z}})\hat{\boldsymbol{z}}\Big] \tag{27}$$

$$\omega_2^f = -\omega_{\mathrm{rot},2}+K_\omega\sin({}^2\alpha_{21}-\theta_{\mathrm{rot},2}-{}^2a_{21}) \tag{28}$$

$$\boldsymbol{u}_i^f = -K_p\,{}^iR_1\,\big(\bar{\delta}_{1i}\,{}^1\boldsymbol{\beta}_{1i}-\bar{d}_{1i}R(\theta_{\mathrm{rot},1})\,{}^1\boldsymbol{b}_{1i}\big) \tag{29}$$

$$\omega_i^f = \begin{cases} -\omega_{\mathrm{rot},i}+K_\omega\sin({}^i\alpha_{i1}-\theta_{\mathrm{rot},i}-{}^ia_{i1}) & \text{if } \cos{}^ie_{i1}\neq 0 \\ -\omega_{\mathrm{rot},i}+K_\omega\sin({}^i\alpha_{i2}-\theta_{\mathrm{rot},i}-{}^ia_{i2}) & \text{otherwise}\end{cases} \tag{30}$$

where $i=3,\dots,N$, $\hat{\boldsymbol{z}}=(0\,0\,1)^T$, $\bar{\delta}_{1i}=\gamma_{12i}\sec({}^1\eta_{12})$, $\bar{d}_{1i}=\frac{\|{}^2\boldsymbol{b}_{21}\times{}^2\boldsymbol{b}_{2i}\|}{\|{}^i\boldsymbol{b}_{i1}\times{}^i\boldsymbol{b}_{i2}\|}\sec({}^1e_{12})$, ${}^i\boldsymbol{b}_{ij}\equiv({}^ia_{ij},{}^ie_{ij})$, ${}^iR_1$ can be

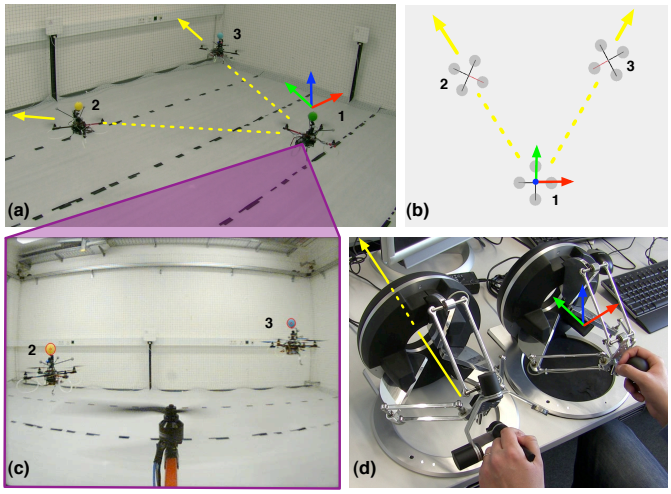

Fig. 8: Experimental setup: (a) Triangular formation during the experiments. The translation of agent 1 on the right is controlled only by the operator. Agent 2 and 3 move on the associated lines towards agent 1 according to the commanded expansion rate. (b) Formation as seen from above in the simulation. (c) Subjective view of agent 1 with agent 2 and 3 highlighted by the optical detection system. (d) Omega.6 (left) and Omega.3 (right) haptic-feedback devices used to control the expansion and translation respectively.

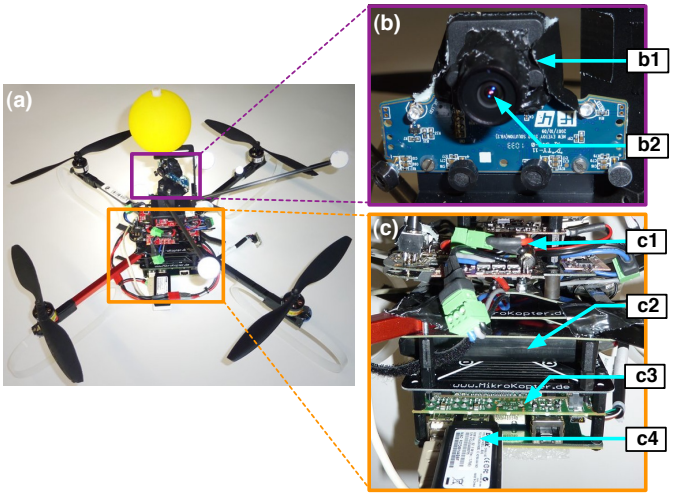

Fig. 9: Quadrotor setup. (a) Quadrotor in its flight configuration. (b) Camera setup, b1) Playstation Eye 3 camera with M12 lens holder attached, b2) 140° lens. (c) Computational setup c1) Microcontroller and IMU, c2) 2200 mAh Lithium Polymer Battery, c3) Mini PC with Intel Atom CPU and extension board holding the connectors, c4) WLAN adapter,

computed as $R^T({}^i\alpha_{i1})R(\pi)R({}^1\alpha_{1i})$, denoting by $R(*)$ the rotation matrix of a given angle around $\hat{\boldsymbol{z}}$ and $M = \begin{bmatrix} M' & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix}$ with $M' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$.

Proposition 1: Given a starting configuration described by the bearings $\{{}^i\boldsymbol{\beta}^0_{ij} \equiv ({}^i\alpha^0_{ij}, {}^i\eta^0_{ij})\}_{i,j=1,\dots,N}$ such that $\left\|{}^1\boldsymbol{\beta}^0_{1i} \times {}^2\boldsymbol{\beta}^0_{2i}\right\| \neq 0$ and $\cos{}^1\eta^0_{12} \neq 0$, and a set of feasible desired bearing trajectories $\{{}^i\tilde{\boldsymbol{b}}_{ij}(t) \equiv ({}^ia_{ij} + \theta_{\mathrm{rot},i}(t), {}^ie_{ij})\}_{i,j=1,\dots,N}$ such that $\left\|{}^1\tilde{\boldsymbol{b}}_{1i}(t) \times {}^2\tilde{\boldsymbol{b}}_{2i}(t)\right\| \neq 0$, $\cos{}^1e_{12} \neq 0$ for all $i = 3,\dots,N$, control (25-30) asymptotically, and almost globally, steers ${}^i\boldsymbol{\beta}_{ij} \to {}^i\tilde{\boldsymbol{b}}_{ij}(t)$ and $\delta_{ij} \to \bar{d}_{1i}\delta^0_{12}\cos{}^1\eta^0_{12}$, for any $(i,j) \in \mathcal{N}$.

As an estimate ξ of the azimuth ${}^h\alpha_{hj}$, we use the following dynamics:

$$ {}^h\dot{\xi}_{hj} = K_\xi({}^h\alpha_{hj} - {}^h\xi_{hj}) - \omega_h \tag{31} $$

where the constant $K_\xi \in \mathbb{R}_{\geq 0}$ is positive when the measure is available (i.e., ${}^i\alpha_{ij} \in [a_{\min}, a_{\max}]$) and are zero otherwise. If the bearing formation is maintained, the estimates (31) will converge to the actual value. On the other hand for a sufficient large values of K_ξ, the estimate is not diverging even if the bearing formation is changing.

Our experimental setup consists of three quadrotors on the slave side, each equipped with a monocular camera, and two force feedback devices on the master side. The visual algorithm and the velocity tracking controller run on a small PC onboard the quadrotor. In order to obtain a fast prototyping, the high level formation control is delegated to an additional PC running Matlab. Both sides communicate by the means of wireless ethernet. The low level control loop regulating the motor speeds in order to achieve the desired attitude, thrust, and yaw rate is implemented on an onboard microcontroller. The haptic feedback devices are connected to a GNU-Linux machine and communicate via ethernet with the quadrotors to allow interaction with the operator.

C. The Master Device

An Omega.3 haptic feedback device is used together with an Omega.6 device[2] to capture the translational and dilational control commands in $\mathbb{R}^3$ and $\mathbb{R}^1$ respectively. The Omega.3 device features 3 actuated degree of freedom (DOF) while the Omega.6 has 6 DOF of which the 3 translational DOF are actuated. For the Omega.6, we constrained the motion to 1 DOF as indicated in Figure 8-d. The forces are presented to the human operator with a frequency of 2.5 kHz.

D. The Slave Robots

The quadrotors from Mikrokopter[3] measure 0.33 m by 0.33 m (without propellers) and have a total mass of 1.3 kg in flight. Each motor produces a maximum thrust of 5 N which allows a maximum vertical acceleration of 5.2 m/s^2. All computational components are centered in the middle of the UAV (Figure 9-c). The low level flight control is implemented on an Atmega 644p microcontroller. This low level control loop regulates the attitude (roll ϕ, pitch θ, yaw ψ) and the thrust of the quadrotor with a frequency of 480 Hz with respect to the data provided by an onboard IMU (Figure 9-c1). The desired values for ϕ, θ and ψ are read with a frequency of about 100 Hz from the serial port. They are further converted into motor commands through a PID-controller. The input to the microcontroller is provided by a velocity tracking system implemented on a small

[2]http://www.forcedimension.com
[3]http://www.mikrokopter.de

Intel Atom QSeven board which is mounted underneath the quadrotor (Figure 9-c3). We will refer to this computer as *Mini PC*. The CPU operates with 1,66 GHz and uses 1 GB of RAM. The Linux operating system loads from a 8 GB flash ROM chip and can use additional storage on an external SD-Card. A WiFi adapter as well as the camera are attached via default USB plugs. The PC further features Gigabit Ethernet and a PCI express slot. A GPU is not available.

In this approach, we use a Vicon[4] tracking system to obtain the velocities of the quadrotors while the relative bearings are calculated through a vision-based algorithm only. The Vicon system can be easily replaced by any reliable onboard velocity measurement system, e.g. a vision based optical flow algorithm or laser range finder.

E. The Vison-based Bearing Measurement

We decided to use a modified low cost Sony Playstation Eye 3[5] color camera, originally designed for the consumer market, for our experiments. The camera is able to provide 60 frames of 640x480 pixel per second and 120 frames in binning mode (320x240 pixel). With the casing removed, the camera weights 25 g including the lens and is therefore particularly suitable for use on small UAVs. In order to gain a wide FOV, we removed the original lens and attached a default M12 lens holder together with an IR filter. For our experiments, we used a $140°$ lens which projects an image of $90° \times 65°$ onto the $1/4''$ sensor. The camera in its final configuration can be seen in Figure 9-b. Black tape was used to shield the sensor from scattered light. The calibration was done individually for all cameras using the camera calibration toolbox for Matlab[6]. To save computational time, for each camera numerically calculated lookup tables holding the bearing angles for all pixel in the image were created.

Figure 9-a shows the configuration of one of our quadrotors. The camera is mounted 9 cm above the microcontroller. Small shock absorbers help to reduce vibrations carried from the motors to the camera. Reflective markers serve as tracking markers for the external Vicon tracking system. In order to allow a simple and reliable visual tracking among the UAVs, we equipped each quadrotor with a ball in an individual color associated with that UAV, installed 8 cm above the camera. This offset between camera and tracked ball results in an error which cannot be corrected without a distance measurement. Thus, in order to allow for an evaluation of the visual system itself, we will incorporate these offsets into the data obtained from the Vicon system in Section VII.

To ensure real time performance, we mainly relied on methods provided by the OpenCV[7] library. Our tracking algorithm first segments the input image for all of the predefined ball colors in HSV (Hue, Saturation and Value) color space. Than, noise in the resulting binary image is eroded and the remaining connected components are expanded to allow a more reliable blob detection even for distant objects. In the following, ellipses are fitted onto all connected components while non circular ellipses up to a threshold are rejected. If there is more than one remaining blob, the one closest to the last located position will be chosen. The lookup tables generated during the calibration phase are used to obtain the angles corresponding to the center of the remaining blob. As a final step, an outlier rejection is applied. Since neither the distance to the traced quadrotor nor the scale is known, the rejection is done based on thresholding the maximum allowed angle change. A visualization of the output in shown in Figure 8-c. Note that it is indeed possible to obtain an estimate of the distance to other UAVs exploiting the known diameter of the colored tracking ball. However, this is only reliable for low distance up to a couple of meters and not suitable for larger areal distances targeted by our work.

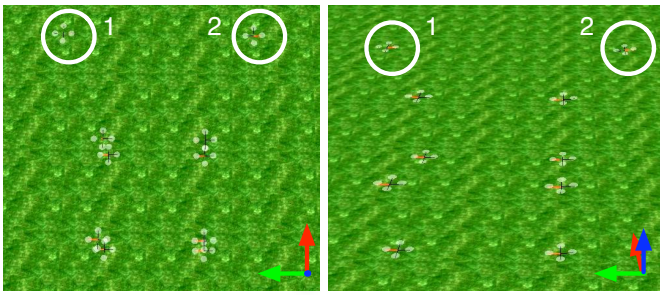

Fig. 10: Top and side view of 10 quadrotors in simulation with agent 1 and 2 highlighted. The other 8 agents form a cube.

The mounting of the camera introduces a small offset between the direction of the camera and the sagittal axis of the UAV. For a compensation, we measured the average offset for both azimuth and elevation over the entire FOV prior to the experiments.

F. Simulation Environment

In order to prove the stability of the controller, we tested our approach in simulation prior to the experiments. The simulation was developed in our lab and makes use of the free 3D engine OGRE[8] as well as PhysX[9] for the simulation of forces and physical interactions. The controller is able to communicate via network with the simulation in the same way as is would communicate with the real robot. Thus, our simulation is able to provide a testbed even for large swarms of robots. Figure 10 depicts a top and side view of a formation of 10 simulated quadrotors.

We simulated a swarm of 10 quadrotors with random starting positions and orientations which differed widely from the desired bearing angles. Figure 11 shows the convergence of the actual bearing angles to the desired angles for all agents.

G. Experimental Design

For the validation of our approach with real robots, we first chose a triangular configuration of three robots with each being

[4]http://www.vicon.com/
[5]http://www.playstation.com/
[6]http://www.vision.caltech.edu/bouguetj/calib_doc/
[7]http://opencv.willowgarage.com
[8]http://www.ogre3d.org
[9]http://www.nvidia.com/object/physx_new.html

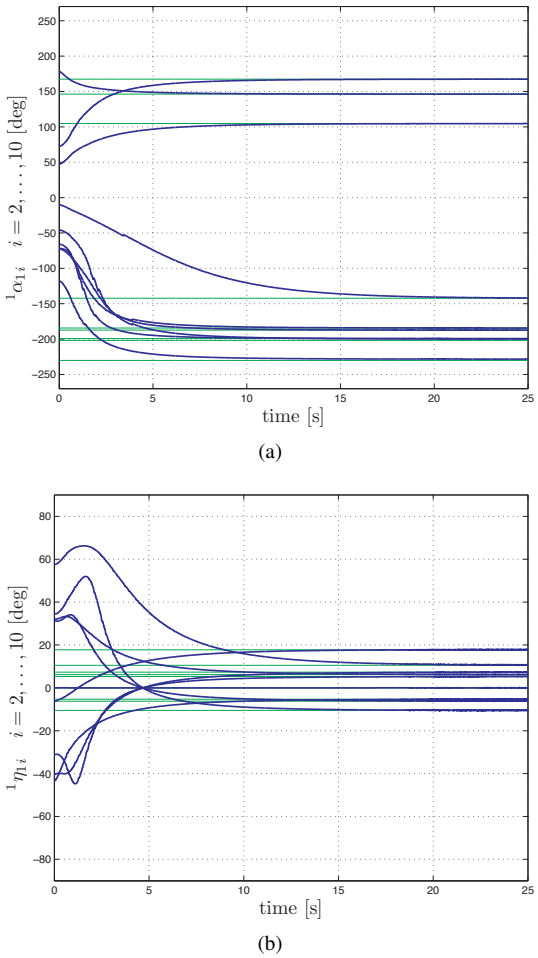

Fig. 11: Convergence of (a) azimuth and (b) elevation angle of 10 simulated quadrotors, seen from agent 1, to the desired angles from random starting locations.

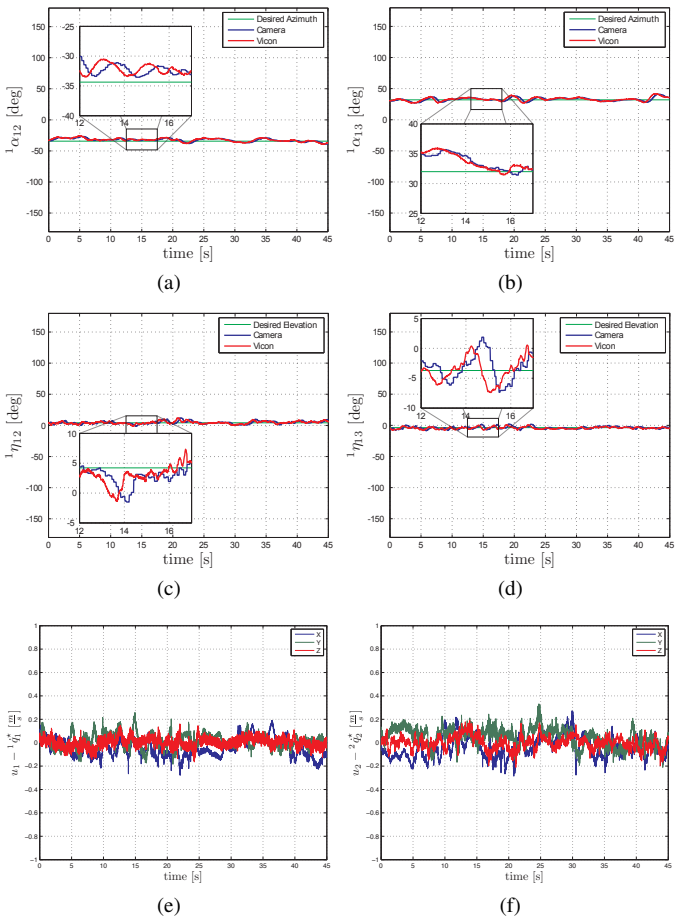

Fig. 12: Bearing Angles obtained from our visual system in comparison with data from the Vicon tracking system and the desired bearing angle in the first experiment. (a) Azimuth between agent 1 and 2 and (b) between agent 1 and 3. (c) Elevation between agent 1 and 2 and (d) between agent 1 and 3. (e) Tracking error of agent 1 and (f) agent 2

able to see all other agents at the same time (Figure 8-a). Thus, we were able to use the original controller presented in Part I [9]. All quadrotors were roughly oriented towards the center of gravity of the formation, thus inducing relative bearing angles around $-30°$ and $30°$. In the starting configuration (approximately 2.10 m distance between the quadrotors), the hight was set to 0.7 m, 0.55 m and 0.85 m for agent 1, 2 and 3 respectively.

In a second set of experiments, we altered the formation such that one angle of the triangle exceeded $90°$ and therefore the corresponding agent 3 was only able to retrieve the relative bearing to one of the two other agents at a time. Thus, agent 3 was forced to use the controller for a limited FOV as presented in Section VI-A while agent 1 and 2 did not rotate. We used a sinusoidal profile with an amplitude of $26°$ and a frequency of 0.8 Hz for the scanning rotation $\theta_{\text{rot}}(t)$. For this second experiment, the initial hight was set to 0.7 m for all quadrotors. W.l.o.g, the estimator was initialized with the help of data obtained from the Vicon system for a few seconds to allow for a sequential take-off procedure of the UAVs.

VII. Experimental Results and Discussion

In the following, we will describe several experiments to validate our approach for visually controlled bearing formations. First, we will analyze the accuracy of the visual system. We will then evaluate the system with three real robots. Finally, we will show the behavior of a simulated group of 10 quadrotors. The experiments are further documented in the accompanying video.

First, to validate our bearing-only multiple UAV teleoperation system, we compared the output of our algorithm with the ground truth obtained from the Vicon tracking system. For this experiment, we recorded the real bearing angles calculated from the positions and orientations given by the Vicon system together with the relative bearing angles obtained from our visual system during an actively controlled flight with three UAVs. W.l.o.g. we picked the angles between agent 1 and 2 as well as between agent 1 and 3 for evaluation. The results are presented in Figure 12.

We were able to prove real time performance on the

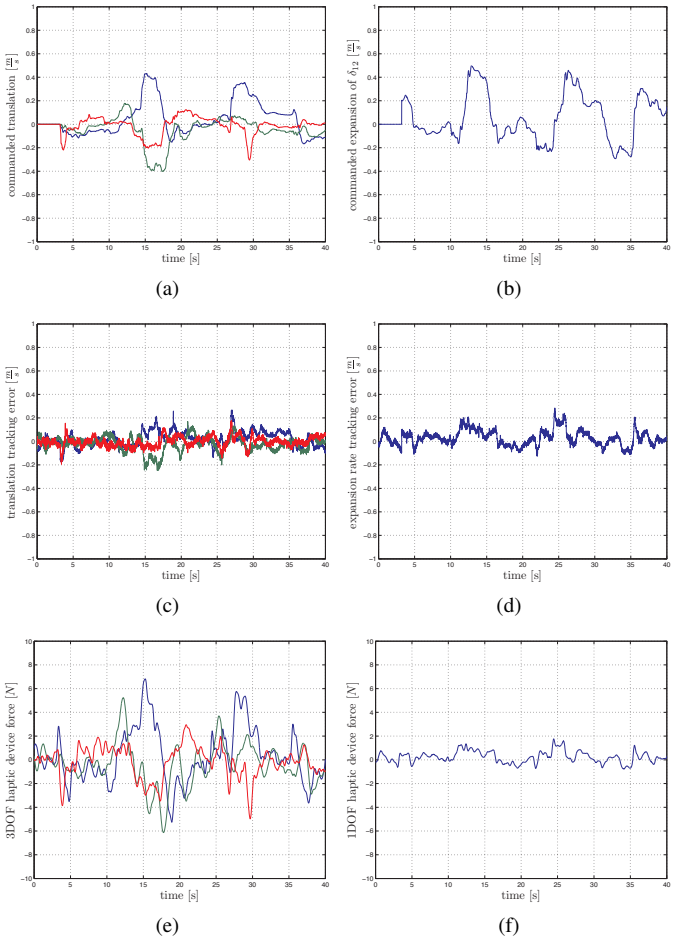

Fig. 13: In (positions) and output (forces) of the force feedback master device in the first set of experiments. (a) Commanded translational velocity and (b) expansion rate. (c) Translational velocity and (d) expansion rate tracking error. (e) Translational velocity and (f) expansion rate force on the haptic device.

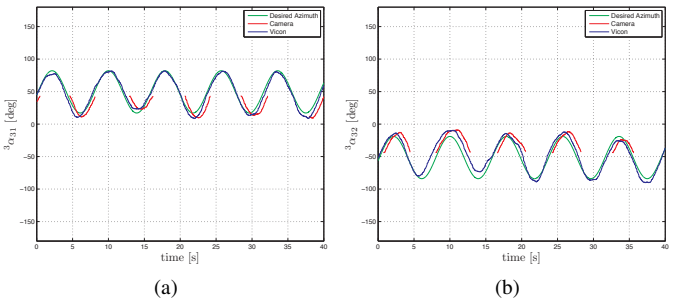

Fig. 14: Azimuth angles during the second experiment. (a) Azimuth between rotating agent 3 and 1 and (b) between agent 3 and 2. The output of the vision tracking algorithm (red) is only plotted when the corresponding other agent was within the FOV of agent 3.

described Mini PC. The algorithm runs with a frequency of 7 Hz. However, we measured a constant temporal lag of three frames compared to the data obtained from the Vicon system. The lag of three frames was observed independently of the underlying hardware system. We assume that it was caused by the use of the default video-for-linux drivers. Thus, on the mobile setup, this lag measured in average 500 ms in contrast to 120 ms on desktop hardware, where the algorithm is executed with a frequency of 30 Hz.

For the presented data, we found a mean error of $-0.24°$ and $-0.44°$ with a standard deviation of $0.30°$ and $0.40°$ for the azimuth between agent 1 and 2 (Figure 12-a) and agent 1 and 3 (Figure 12-b) respectively. The mean elevation error was $-0.51°$ and $-0.53°$ with a standard deviation of $1.46°$ and $1.45°$ respectively. The higher standard deviation for the elevation is due to the less stable hight than yaw control of our quadrotors. The controller was able to stabilize azimuth and elevation up to a mean error of $-1.70°$ and $-4.62°$ with a standard deviation of $2.72°$ and $2.36°$ for azimuth and elevation compared to the desired values respectively. The high standard deviation is mainly a consequence of the ongoing human control. The quadrotors demonstrated to follow the commanded velocities very precisely. The mean velocity tracking error of agent 1 was $0.0006\frac{m}{s}$, $0.002\frac{m}{s}$ and $-0.006\frac{m}{s}$ with a standard deviation of $0.11\frac{m}{s}$, $0.056\frac{m}{s}$ and $0.042\frac{m}{s}$ for the x, y and z axis respectively.

The data obtained for the master control device is presented in Figure 13. The plots visualizes clearly that both haptic feedbacks are proportional to the corresponding errors. The expansion rate tracking error relies mainly on the expansion rate commands and is independent of the velocity tracking error. The velocity error, however, is dependent on the expansion rate. This validates the intended behavior shown in Figure 8: during expansion, all agents move towards agent 0. Thus, in average, the formation measures a velocity directed to agent 0 which is reflected in an increased velocity tracking error. Obviously, a joint translation of all agents does not affect the expansion rate. Furthermore, it is observable that low level noise in the error signal is filtered before fed back into the force feedback devices.

In the second set of experiments, agent 3 was set to perform a scanning movement in order to detect agent 1 and 2 periodically. The relative azimuth angles of agent 3 are presented in Figure 14. Note that the outer end of the FOV is reached at approximately $43.0°$ and $-43.5°$ for agent 1 and 2 respectively. Thus, the mean error between our algorithm and the ground truth was computed only when the agent was visible within the FOV and is $0.32°$ with a standard deviation of $1.52°$. The error between our visual system and the desired angle is $0.10°$ with a deviation of $4.25°$. For the elevation, as expected, we found results similar to the first experiment, namely $0.25°$ for the mean error and $1.41°$ for the standard deviation.

The corresponding data of the master device obtained during this second experiment is shown, for reasons of completeness, in Figure 15.

In order to compensate for roll and pitch motion associated with accelerations of the quadrotor, we tested a compensator based on IMU readings in simulation. However it turned out

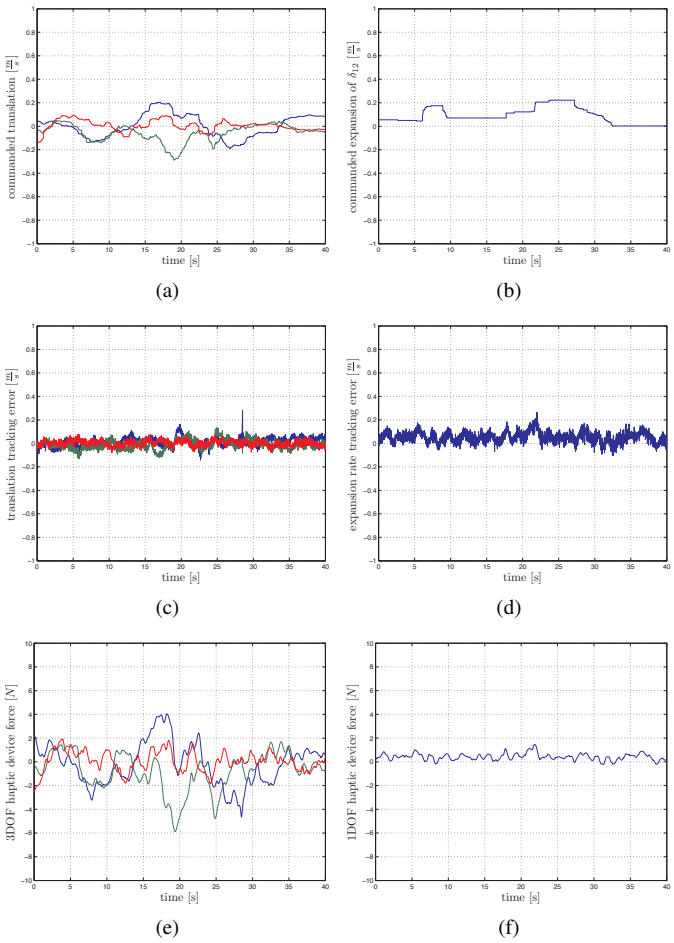

Fig. 15: In (position) and output (forces) of the force feedback master device during the second experiment. (a) Commanded translational velocity and (b) expansion rate. (c) Translational velocity and (d) expansion rate tracking error. (e) Translational velocity and (f) expansion rate force on the haptic device.

that this compensation was not improving the performance of the system significantly. Thus, we decided to omit the compensation which would have introduced another lag caused by the need of an additional communication pathway between the microcontroller and the Mini PC.

VIII. Conclusions

In this notes, we have concisely reviewed some recent advancements in the field of aerial teleoperation, i.e., how to bilaterally couple a single human operator with a remote fleet of semi-autonomous UAVs. In this Part (the second of two) we have briefly summarized the top-down approach, where the group is abstracted as a semi-rigid body, and is subject of reversible deformation when interacting with the environment. We remand the interested reader to [8, 9, 10] for further details.

References

[1] 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*, 25(5-6):431–447, 2006.

[2] 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*, pages 3515–3522, Kobe, Japan, May 2009.

[3] 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*, 30 (3), 2010.

[4] 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*, 48(1):162–186, 2009.

[5] P. Yang, R. A. Freeman, and K. M. Lynch. Multi-agent coordination by decentralized estimation and control. *IEEE Trans. on Automatic Control*, 253(11):2480–2496, 2008.

[6] S. Martinez, F. Bullo, J. Cortes, and E. Frazzoli. On synchronous robotic networks - Part II: Time complexity of rendezvous and deployment algorithms. *IEEE Trans. on Automatic Control*, 52(12):2214–2226, 2007.

[7] L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira. Sensing and coverage for a network of heterogeneous robots. In *47th IEEE Conf. on Decision and Control*, pages 3947–3952, Cancun, Mexico, Dec. 2008.

[8] 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*, pages 1341–1347, Shanghai, China, May 2011.

[9] A. Franchi, C. Masone, H. H. Bülthoff, and P. Robuffo Giordano. Decentralized bearing-only formation control for the bilateral teleoperation of multiple UAVs - Part I: Theory. In *Submitted to 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Francisco, CA, Sep. 2011.

[10] V. Grabe, C. Masone, M. Ryll, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Decentralized bearing-only formation control for the bilateral teleoperation of multiple UAVs - Part II: Experiments. In *Submitted to 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Francisco, CA, Sep. 2011.

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

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

[13] P. Robuffo Giordano. Decentralized bilateral aerial teleoperation of multiple UAVs - part i: a bottom-up perspective, Jun. 2011.

[14] 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*, pages 898–905, Shanghai, China, May 2011.

[15] 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.

[16] 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 *Submitted to 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Francisco, CA, Sep. 2011.

[17] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and

defect of nonlinear systems: Introductory theory and examples. *International Journal of Control*, 61(6):1327–1361, 1995.

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

[19] Nicolas Guenard, Tarek Hamel, and Robert Mahony. A practical visual servo control for an unmanned aerial vehicle. *IEEE Trans. on Robotics*, 24(2):331–340, 2008.

[20] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE Int. Conf. on Robotics and Automation*, pages 2520–2525, Shanghai, China, May. 2011.

[21] D. J. Lee and K. Huang. Passive-set-position-modulation framework for interactive robotic systems. *IEEE Trans. on Robotics*, 26(2):354–369, 2010.

[22] C. Secchi, S. Stramigioli, and C. Fantuzzi. *Control of Interactive Robotic Interfaces: a port-Hamiltonian Approach*. Springer Tracts in Advanced Robotics. Springer, 2007. ISBN 9783540497127.

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