

Environment-independent Formation Flight for Micro Aerial Vehicles

Tobias Nägeli¹, Christian Conte², Alexander Domahidi³, Manfred Morari², Otmar Hilliges¹

Abstract—Some aerial tasks are achieved more efficiently and at a lower cost by a *group* of independently controlled micro aerial vehicles (MAVs) when compared to a single, more sophisticated robot. Controlling formation flight can be cast as a two-level problem: **stabilization of relative distances of agents (formation shape control)** and **control of the center of gravity of the formation**. To date, accurate shape control of a formation of MAVs usually relies on external tracking devices (e.g. fixed cameras) or signals (e.g. GPS) and uses centralized control, which severely limits its deployment. In this paper, we present an **environment-independent approach for relative MAV formation flight**, using a distributed control algorithm which relies only on embedded sensing and agent-to-agent communication. In particular, an **on-board monocular camera** is used to acquire relative distance measurements in combination with a **consensus-based distributed Kalman filter**. We evaluate our methods in- and outdoors with a formation of three MAVs while controlling the formation’s center of gravity manually.

I. INTRODUCTION

Swarms of autonomous robots have recently received a lot of attention in the literature, including coordinated groups of terrestrial [1], aerial [2], [3], [4] or aquatic [5] robots. Teams of small, cheap robots can collectively achieve many tasks more efficiently, and in many cases at a lower total cost than a single, more sophisticated robot. Application scenarios include rapid establishment of adhoc wireless communication [4] or sensor [6] networks, aerial surveillance and mapping [7], entertainment [8] as well as search and rescue missions [4]. Micro aerial vehicles (MAVs) are of particular interest in this context, as they can navigate freely in 3D space, carry payloads, can hover and are very robust due to mechanical simplicity.

In order to achieve a stable formation, individuals require knowledge about the positions of, or distances to, other swarm members [9], [10]. Existing approaches to formation flight therefore rely either on low precision sensors, which result in large inter-robot distances, or on external infrastructure such as high-precision cameras. Small formations of less than five MAVs, performing acrobatic aerial maneuvers, have been demonstrated using a room-fixed, external tracking system (e.g., VICON) and centralized, off-board control [2]. Larger formations with up to 16 agents were studied in [3], also relying on external localization and centralized control.



Fig. 1: Inspired by natural swarms, we present a distributed algorithm for highly accurate relative formation flight of MAVs. The method does not rely on external tracking or communication infrastructure and works in indoor and outdoor environments.

Hörtner et al. [8] demonstrated a large formation of 50 quadrotors called Spixels, performing aerial choreography. The system leverages GPS localization and relies on a central, off-board formation controller. Due to GPS inaccuracies, inter-agent distances have to remain relatively large. The sFly project [7] developed visual SLAM based MAV localization methods that have been applied to swarm flight. However, there is no real-time interaction between the swarm members, which makes precise and agile formation flight very difficult. Other testbeds include localization based on wireless network signals [6] or on on-board acoustic sensors [4].

In this paper, we propose a completely self-contained, distributed algorithm for the accurate control of relative formation flight, without relying on external infrastructure for this task. We focus on the problem of stabilizing the relative formation dynamics and assume that the center of gravity of the swarm is controlled separately, as this can be achieved by existing navigation and guidance techniques for single aerial vehicles. Our approach is inspired by work on deep space satellite formations [11], where accurate external localization information is not present.

For the first time, we experimentally demonstrate stable formation flight with an **inter-agent distance of less than 1 m**. Our method does not rely on any global positioning system and uses only on-board vision to identify and localize neighboring agents. The proposed method is completely **distributed**, i.e. the agents rely on local computations and inter-agent communication without central coordination. The accuracy and agility of the resulting formation is significantly higher than it would be possible with GPS or visual SLAM

¹Advanced Interactive Technologies Lab, Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland
naegelit|otmar.hilliges@inf.ethz.ch

²Automatic Control Laboratory, Department of Information Technology and Electrical Engineering, ETH Zurich, 8092 Zurich, Switzerland
cconte|morari@control.ee.ethz.ch

³inspire AG, Technoparkstrasse 1, 8005 Zurich, Switzerland
domahidi@inspire.ethz.ch

based navigation and therefore admits much more flexibility in terms of dynamic maneuvers. Finally, our approach works indoors and outdoors. To our knowledge, this is the first implementation of a flying formation relying on a completely distributed control and estimation architecture, using on-board cameras only. Whilst demonstrated using MAVs, the developed algorithms are not limited to quadrotors nor to the sensors used, and can be applied to other robotic swarms.

II. PRELIMINARIES

A. Coordinate Systems

We use two types of coordinate systems, illustrated in Fig. 2. A **body-fixed coordinate system** B_i is used to describe the individual agents' dynamics. Furthermore a predefined, **global inertial frame** I serves as a reference for all agents in the formation. The rotation matrix $R_{Bi}^I \in \mathbb{SO}(3)$, where \mathbb{SO} is the special Lie group of all rotation matrices, describes the transformation from body frame B_i to inertial frame I , or in other words the *attitude* of a quadrotor i . Since R_{Bi}^I is used to transform measurements and control inputs from the body frame of agent i into the inertial frame, accurate estimation of R_{Bi}^I is fundamental to MAV control.

B. Modeling

The nonlinear attitude dynamics of the quadrotors are significantly faster than the translational dynamics. Therefore, the two dynamics are modeled separately.

1) Agent Model: The four rotors of a quadrotor are mounted in fixed positions w.r.t. the body frame and can be controlled individually as indicated in Fig. 2. Attitude control of a quadrotor is achieved by differential control of the rotor thrusts. Pitch, roll and total thrust T is controlled by varying the individual motor speeds. Yaw is controlled by the average speed of the clockwise and anticlockwise rotating rotors. Note that the system is under-actuated.

We use the quadrotor model derived in [12]. The angular acceleration around the center of gravity is given as

$$\ddot{\omega} = J^{-1} [-\omega \times J\omega + u_a], \quad (1)$$

where $J \in \mathbb{R}^{3 \times 3}$ is the moment of inertia matrix of the quadrotor and $u_a \in \mathbb{R}^3$ is the input, which is applied by the four motor thrust forces. The rotational velocity of the body coordinate frame w.r.t. the inertial frame is given by

$$\dot{R}_{Bi}^I = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} R_{Bi}^I. \quad (2)$$

The skew symmetric matrix in (2) is denoted by $\text{SKEW}(\omega)$.

2) Formation Model: Given the thrust T_i , the rotation matrix R_{Bi}^I and the gravitational force F_g , the dynamics of each agent i are modeled as a point mass in the inertial frame:

$$\ddot{p}_i m = F_{I,i} + F_g \in \mathbb{R}^3 \quad \text{with} \quad F_{I,i} = R_{Bi}^I [0 \ 0 \ T_i]^T. \quad (3)$$

The individual mass point dynamics are used to describe the dynamics of the formation in a relative representation $r_i = p_i - p_j$ in the inertial frame, see Fig. 3. The formation is defined by $N - 1$ relative vectors, see [13], [14], [11] for

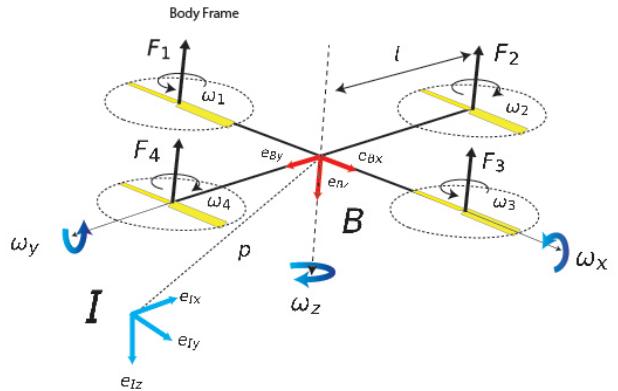


Fig. 2: The basic model of a quadrotor with the inertial system I and the body system B , as well as the rotor thrust forces $F = [F_1, F_2, F_3, F_4]$, the body rotational rates $\omega = [\omega_x, \omega_y, \omega_z]$ and the rotor speeds $\omega_m = [\omega_1, \omega_2, \omega_3, \omega_4]$.

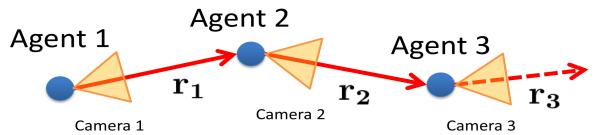


Fig. 3: Basic formation model, consisting of agents approximated by masspoints; r_i are measured by onboard cameras.

a detailed analysis. Using (3), the dynamics of the N -agent system in Fig. 3 is given by $N - 1$ differential equations:

$$\ddot{r}_i = \ddot{p}_i - \ddot{p}_{i+1} = \frac{F_{I,i}}{m} - \frac{F_{I,i+1}}{m} \in \mathbb{R}^3, \quad i = 1, \dots, N - 1. \quad (4)$$

The formation has $(N - 1) \times 6$ states, i.e. all relative positions and velocities in all three dimensions, updated according to the discrete-time state space model

$$x_{k+1} = Ax_k + Bu_k, \quad z_k = C_k x_k, \quad t_k = H_k x_k,$$

where $x = [r_1^T \ r_2^T \ \dots \ r_{N-1}^T \ \dot{r}_{N-1}^T]^T \in \mathbb{R}^{6(N-1)}$ is the state of the formation, $A \in \mathbb{R}^{6(N-1) \times 6(N-1)}$ the system matrix, $u = [u_1^T \ \dots \ u_N^T]^T \in \mathbb{R}^{3N}$ the input vector with $u_1, \dots, u_N := F_{I,1}, \dots, F_{I,N}$ and $B \in \mathbb{R}^{6(N-1) \times 3N}$ the input map. Furthermore, $z = [z_1^T \ \dots \ z_{N-1}^T]^T \in \mathbb{R}^{nc}$ is the measurement vector, $C_k \in \mathbb{R}^{nc \times 6(N-1)}$ the output map, $t = [t_1^T \ \dots \ t_{N-1}^T]^T \in \mathbb{R}^{nr}$ the communication vector and $H_k \in \mathbb{R}^{nr \times 6(N-1)}$ the communication map. The matrices C_k and H_k are time-varying since onboard cameras may capture different neighboring MAVs at different times. Furthermore, A , B , C_k and H_k are structured as

$$A = \begin{bmatrix} A_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A_{N-1} \end{bmatrix} \quad C_k = \begin{bmatrix} C_{k,1} \\ \vdots \\ C_{k,N} \end{bmatrix} \quad H_k = \begin{bmatrix} H_{k,1} \\ \vdots \\ H_{k,N} \end{bmatrix}$$

$$B = \begin{bmatrix} B_{11} & B_{12} & 0 & \cdots & 0 \\ 0 & B_{22} & B_{23} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & B_{N-1N-1} & B_{N-1N} \end{bmatrix}.$$

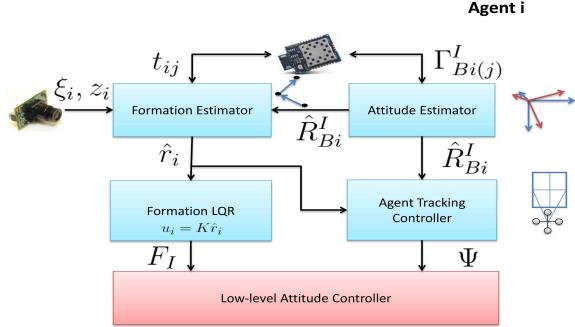


Fig. 4: Overview of control structure, with one instance running on each agent. The relative state information t_{ij} and the relative orientations $\Gamma_{Bi(j)}^I$ are communicated between agents.

III. DISTRIBUTED FORMATION CONTROL

The main goal of this work is the accurate control of MAV formations without centralized computing or remote sensing infrastructure, such as GPS or markers on the ground. Therefore, all sensing and computing is performed on-board and in a distributed manner. This leads to a number of challenges, three of which are described in the following:

a) *Distributed Formation State Estimation*: A single MAV cannot observe the state of the whole formation due to limited local measurements. Therefore, the formation state needs to be estimated in a distributed manner using inter-agent communication.

b) *Inertial Coordinate Frame Consensus*: Each agent uses local IMU and magnetic field measurements to estimate an inertial coordinate frame. Since these measurements are noisy and the local sensors might be differently calibrated, the local estimates may differ considerably from agent to agent. Therefore, the agents need to use communication to reach a consensus on the inertial frame.

c) *Visual Neighbor Tracking*: Due to the limited field of view of the onboard cameras, each MAV needs to actively track neighboring MAVs to ensure the availability of relative position measurements.

To address these challenges, we propose the distributed estimation and control architecture depicted in Fig. 4, the components of which are explained in detail in the following sections. In particular, in Section III-A, the *formation state estimator* is presented, which is based on a distributed consensus Kalman filter. In Section III-B, the linear quadratic regulator (LQR) based *formation controller* is discussed. Furthermore, in Section III-C the *inertial frame estimator* is presented, by which the MAVs ensure consensus on the global inertial frame and in Section III-D the *agent tracking controller* is discussed, by which the MAVs ensure availability of relative distance measurements.

A. Formation Estimator

The formation estimator is built as follows. Each MAV estimates the state vector of the complete formation, by using local measurements and agent-to-agent communication. In particular, each MAV runs a local Kalman filter instance,

which combines local relative distance measurements, obtained by on-board cameras, with state estimates received from other agents by communication. To ensure consensus on the local state estimates, the filter instances are coordinated by a consensus scheme originally proposed in [15]. In particular, we build upon an adaptation of this scheme by [16], which was designed for deep space satellite formations.

In the following, the local estimate of the formation state x by agent i is denoted by \hat{x}^i . The subscript $\{k|k\}$ denotes an a-posteriori and the $\{k|k-1\}$ an a-priori estimate, respectively. Extending the definition of the communication vector t in Section II-B.2, the vector t_{ij} is the message sent from agent j to agent i . The local filter equation is then given as

$$\dot{\hat{x}}_{k|k}^i = \hat{x}_{k|k-1}^i + L_i \left(z_i - C_{k,i} \hat{x}_{k|k-1}^i \right) + F_i \sum_{j=1}^N t_{ij} - H_{k,j} \hat{x}_{k|k-1}^j, \quad (5)$$

where $\hat{x}_{k|k-1}^i \triangleq A \hat{x}_{k-1|k-1}^i + B u_{k,i}$ is the model based prediction term, $L_i(z_i - C_{k,i} \hat{x}^i)$ is the measurement update term with Kalman gain L_i , and $F_i \sum_{j=1}^N (t_{ij} - H_{k,j} \hat{x}^j)$ is the additional communication consensus correction term with gain F_i , which is necessary to ensure convergence of the local estimates. Combining measurement and communication vector, the filter equation (5) can be written using the matrices

$$L_r \triangleq [L, F], \quad C_r \triangleq \begin{bmatrix} C \\ H \end{bmatrix}, \quad R_r \triangleq \begin{bmatrix} R & 0 \\ 0 & N \end{bmatrix}, \quad z_r \triangleq \begin{bmatrix} z \\ t \end{bmatrix},$$

and the estimation procedure can be written in standard Kalman filter form with prediction and innovation step. Details are given in Algorithm 1.

Remark 1: For theoretical stability/convergence guarantees of the estimator, the original formulation of Algorithm 1 as proposed in [15] additionally requires a consensus on the error covariance \hat{P}_i , and Gaussian noise distributions for both the measurement and communication errors. Due to limitations on the communication bandwidth, we have omitted the error covariance consensus in Algorithm 1. In addition, the Gaussian distribution assumption is likely to be violated in practice, especially during communication packet drops. Nevertheless, we have not observed any stability problems in our experiments, and the estimator performance was sufficiently good to achieve an accurate control of the formation shape in all cases.

Algorithm 1 Linear Kalman Filter with Consensus for Formation Estimation

Require: Proc., meas., comm. noise covariances: Q_r, R_r, N

Local Formation Prediction Step:

- 1: $\hat{x}_{k|k-1}^i = A \hat{x}_{k-1|k-1}^i + B u$ with $u = -K \hat{x}_{k-1|k-1}^i$
- 2: $P_{k|k-1} = A P_{k-1|k-1} A^T + Q_r$

Global Formation Update and Consensus Step:

- 3: $L_r = P_{k|k-1} C_r^T (C_r P_{k|k-1} C_r^T + R_r)^{-1}$
 - 4: $\hat{x}_{k|k}^i = \hat{x}_{k|k-1}^i + L_r (z_r - C_r \hat{x}_{k|k-1}^i)$
 - 5: $P_{k|k} = (I - L_r C_r) P_{k|k-1}$
-

B. Formation LQR

Since every agent estimates the full state vector locally, each agent can use full state feedback locally without additional communication. At time k , every agent uses a local LQR state feedback controller (see e.g. [17]) to compute the local input $u^i = -K^i \dot{x}_{k|k}^i$, where $\dot{x}_{k|k}^i$ is agent i 's estimate of the formation.

Remark 2: Note that formation only stabilizes relative positions. The absolute position of the formation, i.e. its center point, is steered manually in this work. Existing approaches from autonomous flight of one vehicle could be applied to steer the center point of the formation autonomously.

C. Inertial Frame Estimator

The rotation matrix R_I^{Bi} is used by each agent i to transform data from the local body-frame into the inertial-frame and vice versa. In order to compute an estimate of the rotation matrix, denoted in the following by \hat{R}_I^{Bi} , the inertial measurement unit (IMU) on the quadrotor is used, providing measurements of the body angular rates ω , the acceleration vector $a \in \mathbb{R}^3$ and the magnetic field strength $\mu \in \mathbb{R}^3$. These measurements are in general noisy and in case of the angular rate sensors drift over time. As a result, the local inertial frames will vary from agent to agent.

In order to correct for these local differences, we implement a consensus scheme on the inertial frame using communication among agents. In particular, an adapted version of the consensus algorithm for $\mathbb{SO}(3)$ as presented in [18] in combination with an extended Kalman filter (EKF) is used, and given explicitly in Algorithm 2. Similar to the linear Kalman filter (KF) in Algorithm 1, Steps 1 – 3 constitute the discrete-time nonlinear prediction of the EKF, which is obtained by a forward Euler discretization of Eq. (1) and Eq. (2) and the assumption $u_a = 0$. The variable $\lambda \triangleq [a, \mu] \in \mathbb{R}^6$ is introduced as a new state, pointing into the direction

Algorithm 2 Inertial Frame Estimator with Consensus

Local Attitude Prediction Step (IMU):

- 1: $\hat{\omega}_{k|k-1}^i = \hat{\omega}_{k-1|k-1}^i + J^{-1} \left[-\hat{\omega}_{k-1|k-1}^i \times J \hat{\omega}_{k-1|k-1}^i \right]$
- 2: $\hat{\lambda}_{k|k-1}^i = I_2 \otimes \exp \left[\text{SKEW}(\omega) \left(\hat{\omega}_{k|k-1}^i \right) dt \right] \hat{\lambda}_{k-1|k-1}^i$
- 3: $P_{k|k-1} = J_a P_{k-1|k-1} J_a^T + Q_a$

Local Attitude Update Step (IMU):

- 4: $L_a = P_{k|k-1} C_a^T (C_a P_{k|k-1} C_a^T + R)^{-1}$
- 5: $[\hat{\omega}, \hat{\lambda}]_{k|k-1}^T = [\hat{\omega}, \hat{\lambda}]_{k|k-1}^T + L_a (z_a - [\hat{\omega}, \hat{\lambda}]_{k|k-1}^T)$
- 6: $P_{k|k} = (I - L_a C_a) P_{k|k-1}$

Generate Local $R_{Bi}^I = [e_x, e_y, e_z]$:

$$7: e_z = \hat{a}_{k|k-1}^i \quad e_y = \hat{\mu}_{k|k-1}^i \times \hat{a}_{k|k-1}^i \quad e_x = e_y \times e_z$$

Global Consensus on R_{Bi}^I with other agents:

- 8: $R_{Bi}^{I+} = R_{Bi}^I \exp \left[\sum_{j=1}^N a_{ik} \log \left(R_{Bi}^{IT} \Gamma_{Bi(j)}^I \right) \right]$
 - 9: $\Gamma_{Bj(i)}^{I+} = \xi_j^i R_{Bi}^{I+}$
-



Fig. 5: Quadrotor setup based on an ARDrone frame and PX4 hardware. Each agent is identified via ARToolkit marker.

of the acceleration and the magnetic field. According to our convention, $\hat{\lambda}$ is the estimate of λ . J_a is the Jacobian around the linearization point and \otimes is the Kronecker product. Furthermore, $\Gamma_{Bi(j)}^I$ is the attitude of agent i estimated by agent j and ξ_j^i is the relative orientation of agent j with respect to i measured by the camera of agent i . Steps 4 – 6 are the corresponding update steps of the KF, and Step 7 generates a valid rotation matrix from the current estimate of gravity and the magnetic field. Finally, the consensus step 7 – 8 integrates messages from a neighbouring agent j , who observes relative orientations. Note that in Steps 4 and 6 the matrix $C_a \equiv I_9$, the 9-dimensional identity matrix, since all outputs of the system can be measured by local sensors. More details on Algorithm 2 can be found in [18].

D. Agent Tracking Controller

Each agent i tracks exactly one agent j by keeping him in the center of its camera's field of view. The assignment is done in such a way that each N different relative positions, defining the full formation state, are measured. For our experiments, where the MAVs operate in a horizontal plane, each agent controls the yaw rate as a function of the observed agent's deviation from the camera's center of view. For this task, a simple local PI controller is used.

IV. EXPERIMENTAL RESULTS

A. Hardware Setup

The hardware used in all experiments is based on the commercially available AR.Drone (see Fig. 5), where the proprietary electronics were replaced with a Pixhawk PX4 Autopilot, available as open source hard- and software. A quad core ARM Cortex-A9 single-board PC processes imagery acquired from a PointGrey Firefly MV camera, running at 30Hz (3.6mm focal length lens). To attain relative distance and orientation measurements of adjacent agents, we use the ARToolkitPlus library [19] and markers on each agent. Finally, in poorly lit environments (at night) we use electroluminescent foil to back-light the fiducials (see Fig. 5). For the communication channel we use a wireless UART bridge running at 5Hz and with a message packet size of 50 bytes.

B. Experiments

We conducted a number of experiments to evaluate the performance of the distributed estimation and control architecture and the stability of the formation flight, even under dynamic formation changes.

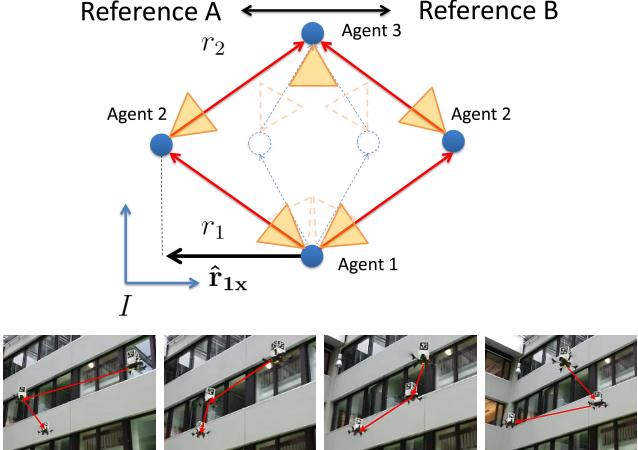


Fig. 6: During the experiment, Ref. A or B are transmitted to the airborne swarm, which dynamically changes its formation.

In our experiments, we flew three MAVs in a triangular shaped formation. Agent 1 is measuring the distance r_1 , agent 2 is measuring the distance r_2 and agent 3 is measuring the linear combination of this two $-r_1 - r_2$. To test the response of estimator and controller we give formation step change commands and switch between ref. A and ref. B as illustrated in Fig. 6

Experiment 1 (Vicon ground truth): To asses the performance of the distributed estimator, we stream the positional estimates of all agents i , denoted as $\hat{r}_{1x}(i)$, and compare them with ground truth positions from a Vicon system. Without loss of generality, we only show the x component of the reference vector r_1 . Fig. 7 summarizes the results.

Experiment 2 (Indoor formation flight): To demonstrate independence of any room-fixed positioning systems, we fly formations in different indoor environments, again with dynamic formation changes, see Fig. 8 middle. Fig. 1 illustrates steering of the formation’s center of gravity with a remote control, which does not compromise the stability of the formation.

Experiment 3 (Outdoor Formation flight): Finally, we verify the outdoor performance by repeating Experiment 2 in an outdoor environment.

C. Results

The top plot in Fig. 7 shows the x -component of the estimated state \hat{r}_{1x} (blue) versus Vicon groundtruth (pink). While there are some positional deviations, the estimated state \hat{r}_{1x} and ground truth match closely most of the time. The quadrotors created significant turbulence in the small lab space, which explain some of the positional error. The conducted outdoor experiment and flights in larger rooms showed stable flight which confirm this assumption. The bottom plot in Fig. 7 shows a close-up of the region circled in red in the top plot. Here the estimates of all three agents i are shown alongside the standard deviation σ . The sector shows a sequence where the state estimates first diverge due to dropped communication packets, and then quickly converge again as soon as communication is re-established.

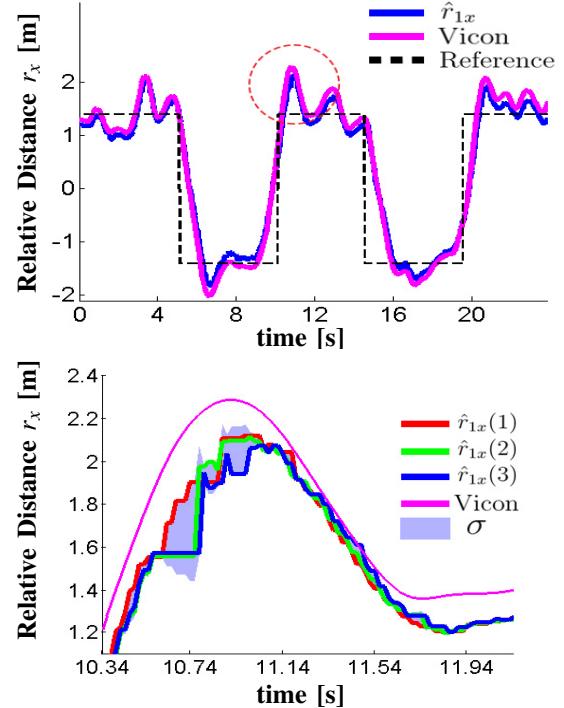


Fig. 7: Top: plot of the estimate $r_{1x}(1)$ of agent 1 compared with Vicon ground truth. Bottom: Close-up of circled region; illustrating a sequence where the state estimates first diverge, due to dropped communication packets, and then consensus is recovered as communication is re-established.

The consensus term in (5) enforces convergence between the distributed estimator, which is reached after 1s. The maximum difference between local estimates is 40 cm before convergence and quickly drops below 2.5 cm.

Beyond the estimator experiment in comparison to a Vicon vision system as described above, we also conducted two qualitative experiments to validate the capabilities to fly in different environments. The bottom row in Fig. 8 shows a free flying formation in a large indoor courtyard without any tracking or global positioning infrastructure. The center of gravity of the formation was steered manually with the formation following the manual input and maintaining its triangular arrangement. Fig. 8, middle illustrates dynamic formation changes in the same environment. Please also refer to the video accompanying this manuscript. Furthermore, a similar experiment was conducted outdoors, which is shown in the top plot in Fig. 8. This further demonstrates the benefits of the proposed consensus based distributed state estimation and control algorithm and the resulting accuracy, which can not be achieved with e.g. a GPS system.

V. CONCLUSION

We have presented a fully distributed estimation and control architecture that enables formation control of MAVs in arbitrary environments without any external sensing structure. The algorithm requires only onboard sensor data from an IMU and relative distance measurements from a marker-based onboard vision system as well as communicated state information from the other members of the formation. A



Fig. 8: Free flying formation during formation change. Top: Outdoor environment. Middle: Indoor environment. Bottom: Stable formation follows manually set position of center of gravity of the swarm.

distributed extended Kalman filter formulation computes a joint estimate of the formation's state. Due to multiple redundant measurements, the system is robust against occlusions as long as there is at least one measurement of each relative state in the formation. A proof-of-concept implementation with three quadrotors demonstrates the feasibility of the approach, while relying on onboard sensing and computation as well as on local communication only. A series of experiments demonstrates the convergence of the distributed state estimation algorithm and the robustness of the proposed technique for both indoor and outdoor formation flight.

While the proposed method scales theoretically seamlessly to larger formations with a high number of agents, there is a practical limitation of the maximal formation members due to the linear scale in bandwidth.

REFERENCES

- [1] M. Saska, V. Vonásek, and L. Přeučil, "Trajectory Planning and Control for Airport Snow Sweeping by Autonomous Formations of Ploughs," *ISI*, vol. 72, pp. 239–261, Apr. 2013.
- [2] S. Lupashin, A. Schollig, M. Hehn, and R. D'Andrea, "The flying machine arena as of 2010," in *IEEE ICRA '11*, pp. 2970–2971, 2011.
- [3] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed," *Robotics Automation Magazine, IEEE*, vol. 17, no. 3, pp. 56–65, 2010.
- [4] P. Lima and D. Floreano, "Audio-based Relative Positioning System for Multiple Micro Air Vehicle Systems," *RSS2013*, no. 266470, 2013.
- [5] M. A. Joordens and M. Jamshidi, "Consensus Control for a System of Underwater Swarm Robots," *IEEE Systems Journal*, vol. 4, pp. 65–73, Mar. 2010.
- [6] A. Purohit, Z. Sun, F. Mokaya, and P. Zhang, "SensorFly: Controlled-mobile sensing platform for indoor emergency response applications," in *IPSN*, pp. 223–234, IEEE, 2011.
- [7] M. Achtelik *et al.*, "Sfly: Swarm of micro flying robots," in *Proceedings of IROS Conf.*, pp. 2649–2650, 2012.
- [8] AÉ FutureLab, "Ars Electronica Quadcopter Swarm," 2012.
- [9] J. Pugh and A. Martinoli, "Relative localization and communication module for small-scale multi-robot systems," in *IEEE ICRA*, 2006.
- [10] M. J. Mataric, "Behaviour-based control: examples from navigation, learning, and group behaviour," *JETAI*, vol. 9, pp. 323–336, Apr. 1997.
- [11] M. V. Subbotin, *Distributed Decentralized Estimation*. PhD thesis, UNIVERSITY OF CALIFORNIA Santa Barbara, 2008.
- [12] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *ICRA, 2011 IEEE International Conference on*, pp. 2520–2525, IEEE, 2011.
- [13] R. Smith and F. Hadaegh, "Closed-loop dynamics of cooperative vehicle formations with parallel estimators and communication," *IEEE Transactions on Automatic Control*, vol. 52, pp. 1404–1414, Aug. 2007.
- [14] R. Smith and F. Hadaegh, "Control topologies for deep space formation flying spacecraft," in *Proc. of the American Control Conf.*, vol. 4, pp. 2836–2841 vol.4, 2002.
- [15] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *Decision and Control, 2005 and 2005 ECC. CDC-ECC'05. 44th IEEE Conference on*, pp. 8179–8184, IEEE, 2005.
- [16] M. Subbotin and R. Smith, "Distributed estimator design for a formation with markovian communication topology," in *Proc. of the American Control Conf.*, pp. 5046–5051, 2007.
- [17] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd ed., 2000.
- [18] R. Tron, R. Vidal, and A. Terzis, "Distributed pose averaging in camera networks via consensus on SE (3)," in *IEEE/ACM ICDS '08*, 2008.
- [19] D. Wagner and D. Schmalstieg, "ARToolkitPlus for pose tracking on mobile devices," in *Proc. CVWW*, pp. 139–146, 2007.