

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Decentralized and Fully Onboard: Range-Aided Cooperative Localization and Navigation on Micro Aerial Vehicles

Abhishek Goudar¹ and Angela P. Schoellig¹

Abstract—Controlling a team of robots in a coordinated manner is challenging because centralized approaches (where all computation is performed on a central machine) scale poorly, and globally referenced external localization systems may not always be available. In this work, we consider the problem of range-aided decentralized localization and formation control. In such a setting, each robot estimates its relative pose by combining data only from onboard odometry sensors and distance measurements to other robots in the team. Additionally, each robot calculates the control inputs necessary to collaboratively navigate an environment to accomplish a specific task, for example, moving in a desired formation while monitoring an area. We present a block coordinate descent approach to localization that does not require strict coordination between the robots. We present a novel formulation for formation control as inference on factor graphs that takes into account the state estimation uncertainty and can be solved efficiently. Our approach to range-aided localization and formation-based navigation is completely decentralized, does not require specialized trajectories to maintain formation, and achieves decimeter-level positioning and formation control accuracy. We demonstrate our approach through multiple real experiments involving formation flights in diverse indoor and outdoor environments.

Index Terms—Multi-robot Systems, distributed robot systems, localization, aerial robots.

I. INTRODUCTION

A team of robots, micro aerial vehicles (MAVs) in this case, that interact in a collaborative manner to fly in a desired formation are suitable in applications that require a wide area coverage such as monitoring and search and rescue [1]. To execute such a formation flight, the robot team must be able to (i) localize, i.e., estimate the pose of the individual robots and, (ii) navigate a planned path in a pre-defined formation. Additionally, it is desirable to perform estimation and navigation in a decentralized and distributed setting (where the computation is done onboard the individual robots) as it is robust to a single point failure and can benefit from distributed storage and computation.

A common approach to cooperative localization is to use an infrastructure of known landmarks, such as the global



Fig. 1: A snapshot of three micro aerial vehicles (MAVs) flying in a triangular formation under a tree canopy (top) and in a GPS-denied environment inside a metal dome (bottom) using our proposed method for localization and distance-based formation control. All estimation and control is done onboard the MAVs in a decentralized manner. A video of all the experiments can be found at following link: http://tiny.cc/mara_loc_nav.

positioning system (GPS) [2] in outdoor environments or a network of ultra-wideband (UWB) radios [3] in indoor environments. Infrastructure-based positioning may not be feasible in many cases, as it is generally expensive and unavailable in many environments. An alternative approach is to use a combination of visual and inertial sensors to localize against visual landmarks [4]. Such an approach requires persistent visual landmarks and can be computationally expensive for real-time applications.

In this work, we propose a decentralized localization approach that uses range, visual, and inertial sensors to perform collaborative localization without the need for pre-installed infrastructure of sensors or building a persistent map of the environment. The combination of range, visual, and inertial sensors for decentralized estimation has been proposed before; however, the previous approaches generally require either strict coordination when solving the associated optimization problem or exchanging high-rate sensor data between agents to maintain consistent estimates. In contrast, our proposed approach has no such requirements.

The problem of coordinating multiple robots to move in a desired formation is typically modeled as a leader-follower problem [5], where the leader is commanded a pre-defined trajectory and the follower robots calculate control inputs to maintain the desired formation. The formation can be specified in terms of relative positions or distances. In this work, we consider distance-based formation control. Formation-based navigation has received considerable attention previously, however, in many cases, the effects of sensor noise and estimation un-

Manuscript received: June, 5, 2025; Revised September, 4, 2025; Accepted October, 13, 2025.

This paper was recommended for publication by Editor M. Ani Hsieh upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and in part by the Canada CIFAR AI Chairs Program.

¹The authors are with the Learning Systems and Robotics Lab at the Technical University of Munich, Germany, and the University of Toronto Institute for Aerospace Studies, Canada. They are also affiliated with the University of Toronto Robotics Institute, the Munich Institute of Robotics and Machine Intelligence (MIRMI), and the Vector Institute for Artificial Intelligence. E-mail: abhishek.goudar@robotics.utoronto.ca, angela.schoellig@tum.de

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

certainty are not considered. For instance, off-the-shelf range sensors are noisy and affected by non-line-of-sight conditions, resulting in outliers and biased measurements. These non-idealities can result in poor tracking performance [6]. In this work, we model formation planning as an inference problem on factor graphs [7] where we use continuous-time motion priors to generate smooth inputs for formation control. In summary, the contributions of our work are:

- a novel approach to distributed distance-based formation control as an inference problem that is sparse, considers state uncertainty, and can be solved efficiently,
- a decentralized block coordinate descent approach to cooperative localization using range, visual, and inertial sensors that does not require exchanging odometry data or synchronous communication between agents, and
- demonstration of joint localization and distance-based formation control on resource-constrained micro aerial vehicles in multiple diverse environments.

II. RELATED WORK

In this section, we first review previous works that focus on decentralized cooperative localization, distance-based formation control, and joint localization and formation control.

Decentralized localization: Our approach to decentralized cooperative localization is related to *decentralized* state estimation. In [8], distributed localization using a decomposed extended Kalman filter (EKF) with a densely connected network is shown. A distributed and decentralized approach to collaborative simultaneous localization and mapping (SLAM) using an EKF with distributed contributions is demonstrated in [9]. While this approach recovers a globally-consistent estimate, it does so with delayed state estimates, which is not ideal for online control. Alternatively, a provably consistent approach to cooperative localization using covariance intersection (CI) in an EKF is demonstrated in [10]. A combination of CI and odometry preintegration for collaborative localization, where each robot estimates the states of all its neighbors, is presented in [11]. In contrast, our proposed method does not communicate odometry information and each robot estimates its trajectory only.

Distributed estimation based on optimization-based approaches can benefit from improved accuracy by reducing linearization errors associated with filtering-based approaches. A distributed approach to localization based on maximum likelihood estimation (MLE), where each robot estimates its state assuming other robot states as constant, is shown in [12]. A maximum a posteriori (MAP) estimator that uses quantized measurements for efficient inference under communication constraints is demonstrated in [13]. Similarly, [14] proposes a MAP estimation approach to multi-robot cooperative localization that has convergence guarantees but requires synchronous communication, which can be difficult to obtain in real settings. A distributed and decentralized approach to SLAM is presented in [15], where consistency is enforced by subtracting double-counted information, but separate linearization between different robots is not accounted for. In this work, we use an asynchronous block coordinate descent (BCD)

approach to MAP inference, that is decentralized and where each robot optimizes a local version of the MAP objective.

Formation control: Formation control is typically modeled using a graph theoretic-approach [16], where the formation is modeled as a graph and the desired control inputs are those that preserve the *rigid* structure of the graph [17]. Optimization-based decentralized control has been applied to high-speed formations flights with variable shapes [18]. An optimal control problem formulation for formation control is provided in [19], with extensions that incorporate unmodeled disturbances. In the previous cases, the effect of sensor noise and estimation uncertainty is not considered in formation control, which can be limiting when using range sensors with relatively large noise values. We present a probabilistic approach to formation control that takes the estimation uncertainty into account, and by extension the sensor noise. Factor graphs have been applied to multiple areas of robotics beyond state estimation, including planning [20] and control [7], [21]. We extend the application of factor graphs to joint localization and formation control.

Cooperative localization and navigation: Next, we consider the body work on joint cooperative localization and navigation. The lightweight and inexpensive nature of visual and inertial sensors makes them ideal for multiagent formation control [4], [22]. Reliance on maintaining visual line-of-sight to landmarks can be challenging and limiting in many cases. In [23], formation control using onboard visual inertial odometry (VIO) and distributed formation execution is shown. Over time, VIO may accumulate drift and become inconsistent. A possible remedy is to generate a global consistent map [24], which can be computationally expensive and challenging in visually degraded conditions. Our proposed localization approach combines VIO with range sensors to obtain drift-free *relative* state estimates that is computationally tractable and robust to visual occlusions. In [25], a combination of range and inertial sensors is used for relative localization and formation control. A limitation of this estimation approach is that formation trajectories involving constant velocities are unobservable. Closely related to our work, [26]–[28] demonstrate joint localization and formation control using range, visual, and inertial sensors. However, these approaches require specialized trajectories involving persistent excitation to achieve formation, which our proposed method does not. More recently, impressive results involving decentralized navigation were presented in [29], where the authors show position-based formation flights with obstacle avoidance. In contrast, we present an inference-based approach to distance-based formation control that incorporates state uncertainty. Unlike [29], we do not consider obstacle avoidance in this work.

III. PROBLEM STATEMENT

The objectives of our work are to (i) estimate the poses of a team of MAVs in a decentralized manner, and (ii) generate smooth control inputs that consider state uncertainty to track a pre-specified distance-based formation pattern. An important objective of our work is to perform joint cooperative localization and navigation onboard the individual robots without centralized computing and coordination. We make the following assumptions. Each robot is equipped with a range sensor

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

(such as a UWB radio), a camera, an inertial measurement unit (IMU), and an onboard compute unit. We assume that the desired formation is *infinitesimally rigid* [16], specified in terms of inter-agent distances. Note that we do not assume a fully-connected graph for both estimation and formation control. We assume that each robot can communicate only with its immediate neighbors (and not the entire team) over a wireless network.

IV. MODELING

A. Decentralized Cooperative Localization

In this section, we provide a description of our proposed cooperative localization approach. To represent information exchange, we use an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{i | i \in \mathcal{S}\}$ is the set of vertices representing the robots as nodes in the graph, indexed using the set $\mathcal{S} = \{a, b, c, \dots\}$, and $\mathcal{E} = \{(i, j) | i, j \in \mathcal{V}\}$ is the set of edges. An edge between a pair of vertices represents a range measurement between the corresponding agents.

We begin by describing the joint probability distribution over all the robot states and their measurements over a finite time window. Consider a team with $N = |\mathcal{S}|$ robots, where $|\mathcal{S}|$ denotes the cardinality of set \mathcal{S} . The state of a robot a at time t is given by its pose: $\mathbf{x}_{a_t} \in SE(d)$, where $d \in \{2, 3\}$, $SE(d) \doteq \{(\mathbf{R}, \mathbf{p}) | \mathbf{p} \in \mathbb{R}^d, \mathbf{R} \in SO(d)\}$ is the special Euclidean group, $\mathbf{p} \in \mathbb{R}^d$ is the position, and $\mathbf{R} \in SO(d)$ represents the rotation as member of the special orthogonal group. In this case, \mathbf{x}_{a_t} refers to a 4×4 homogeneous matrix. The state of the team at some time t is given by $\mathbf{x}_t = [\mathbf{x}_{a_t}^T, \mathbf{x}_{b_t}^T, \dots, \mathbf{x}_{N_t}^T]^T$. We assume that each agent has two sensing modalities: (i) an estimate of its ego motion obtained from VIO and (ii) distance or range measurements to its neighboring agents.

The VIO algorithm estimates the pose of an agent relative to an arbitrary origin. Typically, these pose estimates are generated at rate higher than range measurements. One option is to include every pose estimate as a node in graph, however, this would result in increased computation cost with the size of the graph. We use *odometry preintegration* [6] where multiple VIO pose estimates are combined into a single relative pose between two time steps. We represent the VIO pose and the associated covariance matrix for an agent a at time t by the pair $(\bar{\mathbf{x}}_{o_{a_t}}, \Sigma_{o_{a_t}})$. Consider a sequence of VIO pose estimates: $[(\bar{\mathbf{x}}_{o_{a_t}}, \Sigma_{o_{a_t}}), (\bar{\mathbf{x}}_{o_{a_{t+1}}}, \Sigma_{o_{a_{t+1}}}), \dots, (\bar{\mathbf{x}}_{o_{a_{t'}}}, \Sigma_{o_{a_{t'}}})]$. The consecutive pose estimates can be combined into a single *relative* pose estimate $(\delta\bar{\mathbf{x}}_{o_{a_t}}, \Sigma''_{o_{a_t}})$ using the procedure outlined in [6]. The single relative pose estimate relates the state between two distinct time instants t and t' . It can be incorporated into the factor graph with the following model

$$\delta\bar{\mathbf{x}}_{o_{a_t}} = (\mathbf{x}_{a_t}^{-1} \mathbf{x}_{a_{t'}}) \exp(\boldsymbol{\eta}_{o_{a_t}}^\wedge), \quad (1)$$

where $(\cdot)^\wedge$ maps a vector to the corresponding skew-symmetric matrix representing an element of the Lie algebra $\mathfrak{se}(d)$, $\exp(\cdot)$ is the matrix exponential that maps an element of $\mathfrak{se}(d)$ to $SE(d)$, and $\boldsymbol{\eta}_{o_{a_t}} \sim \mathcal{N}(\mathbf{0}, \Sigma''_{o_{a_t}})$.

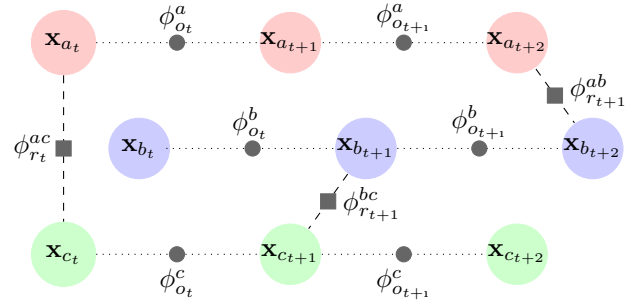


Fig. 2: Factor graph for range-aided cooperative localization with three agents. The states of the agents at different time steps are shown using colored nodes. The edges between the nodes represent intra-agent and inter-agent measurements. Each agent measures its ego motion between using an odometry sensor represented by factor $\phi_{o_t}^a = \phi_o(\mathbf{x}_{a_t}, \mathbf{x}_{a_{t+1}})$ (see (5)). Additionally, an agent a measures its distance to agent b represented by factor $\phi_{r_t}^{ab} = \phi_r(\mathbf{x}_{a_t}, \mathbf{x}_{b_t})$.

The generative model for the range measurement model is as follows. The distance measured by agent a to agent b at time t is modeled as

$$r_{ab_t} = \|\bar{\mathbf{p}}_{b_t} - \bar{\mathbf{p}}_{a_t}\|_2 + \eta_{ra_t}, \quad (2)$$

where for an agent a , $\bar{\mathbf{p}}_{a_t} = \mathbf{R}_{a_t} \mathbf{p}_l + \mathbf{p}_{a_t}$ with \mathbf{p}_l denoting the position offset between the IMU and the range sensor, and $\eta_{ra_t} \sim \mathcal{N}(0, \sigma_r^2)$ is the additive Gaussian noise. The factor graph for this setup is shown in Figure 2. The colored nodes in the graph represent robot states over time, the square factors between the nodes represent inter-agent range measurements, and the circular factors represent the preintegrated odometry factors. Assuming the Markov property over the states and independence of measurement noise, the joint probability distribution over the robots' states and measurements up to time step K is

$$p(\mathbf{x}, \mathbf{y}) = \prod_{i \in \mathcal{S}} p(\mathbf{x}_{i_0} | \check{\mathbf{x}}_{i_0}) \prod_{i \in \mathcal{S}} \prod_{t=1}^{K-1} p(\mathbf{x}_{i_{t+1}} | \mathbf{x}_{i_t}, \delta\bar{\mathbf{x}}_{o_{i_t}}) \prod_{i \in \mathcal{S}} \prod_{j \in \mathcal{S}} \prod_{t=1}^K \kappa_{ij_t} p(\mathbf{x}_{i_t}, \mathbf{x}_{j_t} | r_{ij_t})$$

where $\mathbf{x} = [\mathbf{x}_0^T, \mathbf{x}_1^T, \dots, \mathbf{x}_K^T]^T$, $\check{\mathbf{x}}_{i_0}$ is the known initial state, $\mathbf{y} = [\dots, \delta\bar{\mathbf{x}}_{o_{a_t}}^T, \delta\bar{\mathbf{x}}_{o_{b_t}}^T, \dots, r_{ab_t}, r_{bc_t}, \dots]^T$ is the set of all measurements, and κ_{ij_t} is an indicator variable that is 1 if a range measurement exists between agents a and b at time t and 0 otherwise. The robot state can be estimated by using MAP inference as

$$\mathbf{x}^* = \arg \max_{\{\mathbf{x}_{i_t} \in SE(d)\}} p(\mathbf{x} | \mathbf{y}) \propto \arg \max_{\{\mathbf{x}_{i_t} \in SE(d)\}} p(\mathbf{x}, \mathbf{y}).$$

With the monotonicity of the (negative) logarithm function, the MAP inference objective simplifies to the following non-linear least squares problem

$$J = \min_{\{\mathbf{x}_{i_t} \in SE(d)\}} \left(\sum_{i \in \mathcal{S}} \phi_{\text{init}}(\mathbf{x}_{i_0}) + \sum_{i \in \mathcal{S}} \sum_{t=1}^{K-1} \phi_o(\mathbf{x}_{i_t}, \mathbf{x}_{i_{t+1}}) + \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}} \sum_{t=1}^K \kappa_{ij_t} \phi_r(\mathbf{x}_{i_t}, \mathbf{x}_{j_t}) \right), \quad (3)$$

where

$$\phi_{\text{init}}(\mathbf{x}_{i_0}) = \|\log(\bar{\mathbf{x}}_{i_0}^{-1} \mathbf{x}_{i_0})^\vee\|_{\bar{\Sigma}_{\bar{\mathbf{x}}_{i_0}}}^2, \quad (4)$$

$$\phi_o(\mathbf{x}_{i_t}, \mathbf{x}_{i_{t+1}}) = \|\log((\mathbf{x}_{i_t}^{-1} \mathbf{x}_{i_{t+1}})^{-1} \delta \bar{\mathbf{x}}_{o_{i_t}})^\vee\|_{\Sigma''_{o_{i_t}}}^2, \quad (5)$$

$$\phi_r(\mathbf{x}_{i_t}, \mathbf{x}_{j_t}) = \|r_{ij_t} - (\|\bar{\mathbf{p}}_{i_t} - \bar{\mathbf{p}}_{j_t}\|_2)\|_{\sigma_r^2}^2, \quad (6)$$

are the potential functions associated with different measurement factors, $\log(\cdot)$ is the matrix logarithm that maps an element of the Lie group to the corresponding Lie algebra, $(\cdot)^\vee$ maps a skew-symmetric matrix to a vector, $\bar{\Sigma}_{\bar{\mathbf{x}}_{i_0}}$ is the uncertainty associated with initial state $\bar{\mathbf{x}}_{i_0}$, and $\|\mathbf{e}\|_{\mathbf{Q}}^2 = \mathbf{e}^T \mathbf{Q}^{-1} \mathbf{e}$. The only term in the objective that depends on multiple agents is the range measurement factor (6); the initial state and odometry-based factors depend only on a single agent. Specifically, the joint optimization problem (3) can be decomposed into smaller sub-problems, where each agent only solves the part of the factor graph involving its states using the neighbor agent's pose and covariance.

To solve (3) in a distributed manner, we adopt an asynchronous block coordinate descent (BCD)-based approach where each agent updates its state iteratively using the latest state estimates from its neighboring agents:

$$\mathbf{x}_a^k = \arg \min_{\{\mathbf{x}_{a_t} \in SE(d)\}} \left(\phi_{\text{init}}(\mathbf{x}_{a_0}) + \sum_{t=1}^{K-1} \phi_o(\mathbf{x}_{a_t}, \mathbf{x}_{a_{t+1}}) + \sum_{j \in \mathcal{N}_{a_t}} \sum_{t=1}^{K-1} \kappa_{aj_t} \tilde{\phi}_r(\mathbf{x}_{a_t}, \mathbf{x}_{j_t}^{k-1}) \right), \quad (7)$$

where \mathcal{N}_{a_t} is the set of neighbors of agent a at time t and $\mathbf{x}_{j_t}^{k-1}$ is state of agent j from iteration $k-1$. The range factor $\tilde{\phi}_r$ is modified to account for the uncertainty in state $\mathbf{x}_{j_t}^{k-1}$: $\tilde{\phi}_r(\mathbf{x}_{i_t}, \mathbf{x}_{j_t}^{k-1}) = \|r_{ij_t} - (\|\bar{\mathbf{p}}_{i_t} - \bar{\mathbf{p}}_{j_t}^{k-1}\|_2)\|_{\tilde{\sigma}_r^2}^2$, where $\tilde{\sigma}_r = \sigma_r + \text{tr}(\Sigma_{\mathbf{x}_{j_t}^{k-1}})$, where $\text{tr}(\cdot)$ is the matrix trace.

The number of neighbors, $|\mathcal{N}_{a_t}|$, for any agent is dictated by the formation specification, which is presented in the next section. The nonlinear least squares problem associated with (7) is solved using the Levenberg-Marquardt algorithm. The estimated state \mathbf{x}_a^k is then shared with the neighboring agents to update their respective states $\{\mathbf{x}_{j_t}^k | j \in \mathcal{N}_{a_t}\}$. To keep computational cost low, we use a *fixed-lag smoother* approach, where states and measurements older than a particular fixed time window are marginalized. The iterations (7) are run periodically at a fixed rate asynchronously; i.e., each agent solves an iteration of (7) independently. The motivation for asynchronous BCD is to accommodate unreliable data networks, where solving (3) in coordinated manner cannot be guaranteed. We found in practice that asynchronous BCD achieves sufficient accuracy for reliable navigation.

B. Decentralized Formation Control

The desired formation is represented using the graph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E}_d)$ where $\mathcal{E}_d \subseteq \mathcal{E}$. The desired formation is parameterized by inter-agent distances and is assumed to be infinitesimally rigid. We formulate decentralized formation control as an instance of MAP inference that takes into account two sources of information: a motion prior that dictates the agent's motion

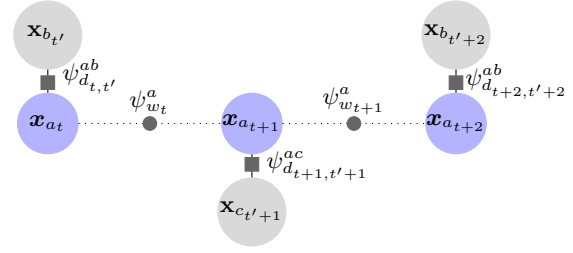


Fig. 3: Factor graph associated with agent a 's formation control step. The state at a time t , \mathbf{x}_{a_t} , consists of the agent's position and velocity input. A Gaussian process-based smooth motion prior is added as a factor between two consecutive time stamps: $\psi_{w_t}^a = \psi_w(\mathbf{x}_{a_t}, \mathbf{x}_{a_{t+1}})$ (see (14)). Additionally, the desired distances to neighbor agents, for maintaining a pre-specified formation, are represented by the factors: $\psi_{d_{t,t'}}^{ab} = \psi_d(\mathbf{x}_{a_t}, \mathbf{x}_{b_{t'}})$ (see (15)), where the neighboring agent's state is fixed to its latest estimate (as indicated by the grayed out node).

over a finite time horizon and the distance constraints to neighboring agents for maintaining the desired formation.

We generate a motion prior for each agent using the framework of continuous-time Gaussian process (GP) regression [30]. The state $\mathbf{x}_{a_t} = [\mathbf{p}_{a_t}^T, \mathbf{u}_{a_t}^T] \in \mathbb{R}^{2d}$ consists of the agent's position $\mathbf{p}_{a_t} \in \mathbb{R}^d$ and the velocity control input $\mathbf{u}_{a_t} \in \mathbb{R}^d$. The motion model for an agent a is:

$$\underbrace{\begin{bmatrix} \mathbf{p}_{a_t} \\ \mathbf{u}_{a_t} \end{bmatrix}}_{\mathbf{x}_{a_t}} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{p}_{a_t} \\ \mathbf{u}_{a_t} \end{bmatrix}}_{\mathbf{x}_{a_t}} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}}_{\mathbf{L}} \mathbf{w}_{a_t}, \quad (8)$$

where \mathbf{I} is the identity matrix and $\mathbf{w}_{a_t} \sim \mathcal{N}(0, \mathbf{Q}_{a_w} \delta(t-t'))$ is a zero-mean GP with power spectral density matrix \mathbf{Q}_{a_w} . The trajectory prior (mean and covariance) are calculated in closed-form [30, Section 3.4.2] by integrating the motion model (8) over a finite time horizon using the latest estimated state (and covariance) as the initial condition. The model (8) is similar to a double-integrator model, but different in that it incorporates unmodeled disturbances with an additive noise term. The mean of the GP prior between two consecutive time steps is

$$\mathbf{x}_{a_{t+1}} = \Phi_{t,t+1} \mathbf{x}_{a_t} \quad (9)$$

with

$$\Phi_{t,t+1} = \exp(\mathbf{A} \Delta t) = \begin{bmatrix} \mathbf{I} & \Delta t \\ \mathbf{0} & \mathbf{I} \end{bmatrix},$$

where Δt is the duration between time steps t and $t+1$. The covariance associated with this incremental motion is

$$\begin{aligned} \mathbf{Q}_{a_t} &= \int_t^{t+1} \Phi_{t,s} \mathbf{L} \mathbf{Q}_{a_w} \mathbf{L}^T \Phi_{s,t+1}^T ds, \\ &= \begin{bmatrix} \frac{1}{3} \Delta t^3 \mathbf{Q}_{a_w} & \frac{1}{2} \Delta t^2 \mathbf{Q}_{a_w} \\ \frac{1}{2} \Delta t^2 \mathbf{Q}_{a_w} & \Delta t \mathbf{Q}_{a_w} \end{bmatrix}. \end{aligned} \quad (10)$$

In addition to the motion model, at each time step, agent a must also maintain specific distance to its neighbors in order to achieve the desired formation. One approach is to incorporate this distance requirement as *constraints* in the optimization problem [31]. We consider an unconstrained formulation where the desired distances are incorporated into the cost function with the following generative model:

$$d_{aj_t} = \|\mathbf{p}_{a_t} - \mathbf{p}_{j_t}\|_2 + \eta_{j_t}, \quad j \in \mathcal{N}_{a_t}, \quad (11)$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

where $\mathbf{p}_{j,t'}$ is the estimated position of agent j at some time t' (typically older than t). The additive noise $\eta_{j,t'} \sim \mathcal{N}(0, Q_{j,t'})$ takes into account the marginal uncertainty of agent j 's position: $Q_{j,t'} = \text{tr}(\hat{\Sigma}_{\mathbf{p}_{j,t'}})$, where $\hat{\Sigma}_{\mathbf{p}_{j,t'}}$ is the sub-matrix corresponding to the estimated uncertainty in position. The motivation for our proposed formulation is as follows. Firstly, the motion prior of an agent is generated using its own estimation uncertainty and its motion model uncertainty. This has the effect of generating smooth priors when the estimation uncertainty or model uncertainty is high. Secondly, we use the estimated states of the neighboring agents $\mathbf{p}_{j,t'}$, instead of their predicted states, as using predicted states would require exchanging information at the formation controller frequency, which can be expensive and cannot be guaranteed in practice. Thirdly, our approach incorporates estimation uncertainty of neighboring agents as well: if an agent has high state uncertainty, the corresponding distance constraint gets penalized more in the optimization problem, which is desirable. Finally, we incorporate the desired inter-agent distances in the cost function (instead of as equality constraints) as it may not be possible to strictly adhere to inter-agent distances for short periods of time in lieu of unreliable communication network or other critical tasks.

The factor graph associated with the formation controller of a single agent is shown in Figure 3. Note that the structure of the formation control problem is similar to that of the localization problem from the previous section. Agent a 's objective function for formation control is as follows:

$$\tilde{J}_a = \min_{\{\mathbf{x}_{a_t} \in \mathbb{R}^{2d}\}} \left(\psi_{\text{init}}(\mathbf{x}_{a_K}) + \sum_{t=K}^{K'-1} \psi_w(\mathbf{x}_{a_t}, \mathbf{x}_{a_{t+1}}) + \sum_{j \in \mathcal{S}} \sum_{t=1}^{K'} \kappa_{aj_t} \psi_d(\mathbf{x}_{a_t}, \mathbf{x}_{j_t'}) \right), \quad (12)$$

where $K' > K$, $\mathbf{x}_a = [\mathbf{x}_{a_K}^T, \mathbf{x}_{a_{K+1}}^T, \dots, \mathbf{x}_{a_{K'}}^T]^T$,

$$\psi_{\text{init}}(\mathbf{x}_{a_K}) = \|\bar{\mathbf{x}}_{a_K} - \mathbf{x}_{a_K}\|_{\mathbf{Q}_{\mathbf{x}_{a_K}}}^2, \quad (13)$$

$$\psi_w(\mathbf{x}_{a_t}, \mathbf{x}_{a_{t+1}}) = \|\mathbf{x}_{a_{t+1}} - \Phi_{t,t+1} \mathbf{x}_{a_t}\|_{\mathbf{Q}_{a_t}}^2, \quad (14)$$

$$\psi_d(\mathbf{x}_{a_t}, \mathbf{x}_{j_t'}) = \|d_{aj_t} - (\|\mathbf{p}_{a_t} - \mathbf{p}_{j_t'}\|_2)\|_{\mathbf{Q}_{j_t'}}^2, \quad (15)$$

and $\bar{\mathbf{x}}_{a_K} = [\mathbf{p}_{a_K}^T, \mathbf{v}_{a_K}^T]^T$ consists of the estimated position \mathbf{p}_{a_K} , and the velocity estimate \mathbf{v}_{a_K} from odometry. The uncertainty matrix $\mathbf{Q}_{\mathbf{x}_{a_K}}$ is a block diagonal matrix composed of the marginal covariance of agent a 's estimated position uncertainty $\hat{\Sigma}_{\mathbf{p}_{a_t}}$ and the uncertainty associated with velocity reported by the VIO algorithm \mathbf{Q}_{v_K} : $\mathbf{Q}_{\mathbf{x}_{a_K}} = \text{diag}(\hat{\Sigma}_{\mathbf{p}_{a_t}}, \mathbf{Q}_{v_K})$, where the $\text{diag}(\cdot)$ operator constructs a block-diagonal matrix from its arguments. As before, κ_{aj_t} is an indicator variable that is 1 if a distance constraint exists between agents a and i at time t and 0 otherwise. The above formulation takes the estimated state uncertainty into account in the factor $\psi(\mathbf{x}_{a_K})$. The desired positions and control inputs to achieve the desired formation are obtained by minimizing (12):

$$\mathbf{x}_a^* = \arg \min_{\{\mathbf{x}_{a_t} \in \mathbb{R}^{2d}\}} \tilde{J}_a. \quad (16)$$

Solving (16) gives agent a 's positions and the corresponding velocity control inputs over a finite time horizon. We can view the calculated state $\mathbf{x}_{a_t}^* = [\mathbf{p}_{a_t}^{*T}, \mathbf{u}_{a_t}^{*T}]^T$ as consisting of a position control input and a velocity control input to achieve the desired formation. Depending on the lower-level control interface, either input can be chosen. We observed that the position control inputs resulted in smoother control as the velocity inputs were more susceptible to errors in presence of higher measurement noise. To keep the computational cost low, we solve the optimization problem (16) in a fixed-lag smoother fashion over a finite time horizon. We use only the last control input $\mathbf{x}_{a_{K'}}^*$ and the optimization problem is run periodically at a fixed rate. Each agent independently executes its version of (16) for formation control.

C. Implementation Details

The asynchronous BCD approach does not require strict coordination while solving the back-end optimization problem. However, network delays and packet losses can cause measurements to be delayed or missed entirely. Note that the only data exchanged between the agents are the range measurements and the corresponding agent poses and covariances. To delay with network delays and packet losses, we use the dual smoother architecture outlined in [6]. Specifically, the FLS associated with the decentralized estimator runs at a lower rate and does not require measurements to arrive in real time. Concretely, if a range measurement is delayed due to network latency, it is added to the factor graph at the corresponding time step in the past. Similarly, if a neighbor agent does not respond with its pose for a particular range measurement or the corresponding packet is lost, the range measurement is removed from the factor graph. To facilitate closed-loop navigation, the low-rate pose output of the decentralized estimator is then combined with the latest VIO estimates in another smoother that runs at a higher rate for closed-loop navigation.

V. EVALUATION

We evaluate our proposed cooperative localization and navigation approach through *leader-follower* formation flights in simulation and real experiments. In this case, only the leader agent is commanded a pre-defined trajectory, the other agents (followers) compute control inputs to follow the leader while maintaining a desired formation. For evaluation, we assume that only the leader has a source of absolute pose to track its designated trajectory. Note that this assumption is for evaluation only and can be relaxed as shown in the accompanying video. The localization and formation control algorithms are implemented using the GTSAM library [32]. Inter-agent communication is done over WiFi using ROS2 Jazzy (with eProsima Fast DDS backend) middleware.

We compare our proposed localization approach against centralized batch trajectory estimation (referred to as *batch*) as a baseline. To evaluate the localization performance, we use the average position *root mean square error* (RMSE) of all the follower agents as a metric. We compare our proposed formation control approach with the gradient control method (referred to as *gcm*) outlined in [17], which is a decentralized

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

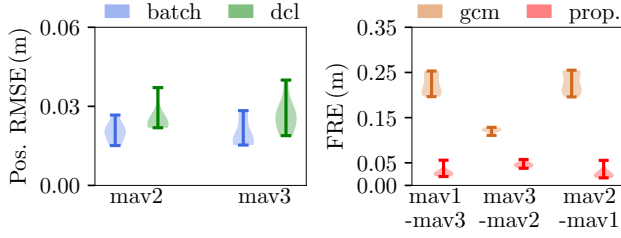


Fig. 4: Error distribution plots for localization (left) and formation control (right) from 10 simulations. The proposed decentralized cooperative localization (dcl) method has positioning RMSE similar to centralized batch trajectory estimation. The formation RMSE (FRE) (see (17)) distributions show that the proposed (prop.) approach achieves lower formation error than the baseline gradient control (gcm) approach.

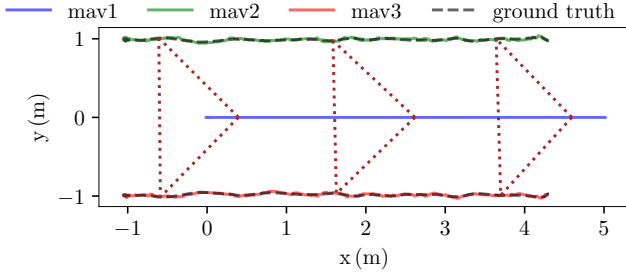


Fig. 5: Estimated and ground-truth trajectories from a leader-follower flight in simulation. The leader (mav1) tracks a straight path, and the other agents (mav2 and mav3) estimate their state and the control inputs to follow the leader in a triangular formation (indicated by the red dotted lines).

approach to rigidity-based formation control. The formation control methods are evaluated by comparing the desired and observed distances using the *formation root mean square error* (FRE) metric:

$$\text{FRE} \doteq \frac{1}{K} \sqrt{\sum_{i=1}^K (\bar{d}_{ij_t} - d_{ij_t})^2}, \quad (i, j) \in \mathcal{E}_d \quad (17)$$

where \bar{d}_{ij_t} and d_{ij_t} are the desired and observed distances between agents i and j at time t , respectively.

A. Simulation

Our simulation setup is based on Ignition Gazebo and consists of a generic environment with multiple MAVs. Each MAV has (i) a range sensor and a generic odometry sensor (to simulate VIO) for estimation, and (ii) a low-level controller to convert high-level setpoints (position and velocity) into motor inputs for closed-loop control. The sensor data is corrupted by adding Gaussian noise reflective of real sensors: $\sigma_r = 0.05$ m. The desired formation is specified a priori. The simulations are run on a single laptop with the autonomy stack (including estimation and formation control) associated with different MAVs running in a decentralized manner.

One of the MAVs is designated as a leader and commanded predefined trajectories; the follower agents use the proposed method to estimate their state and generate control inputs to maintain the desired formation. To prevent the team collectively translating along z -axis (potentially flying into the ground plane), we add additional constraints to the formation control factor graph to maintain constant altitude relative to the terrain. Quantitative results showing estimation and formation RMSE with three MAVs flying in a triangular formation from 10 trials are shown in Figure 4. Qualitative results from

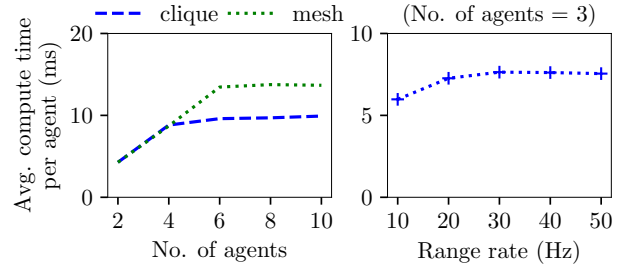


Fig. 6: Average localization solve times per agent vs. (left) number of neighbor agents under mesh (dense) and clique (sparse) connectivity, and (right) range update rate with a team of three agents.

one such experiment showing the estimated trajectories are shown in Figure 5. The proposed fixed-lag smoother-based decentralized cooperative localization (dcl) has positioning RMSE similar to centralized batch trajectory estimation. As highlighted in [6], with noisy range measurements, the estimated pose tends to be non-smooth. The estimation uncertainty captures this variation in the estimated pose, which is subsequently used in our formation control approach. As such, the setpoints from our formation control pipeline afforded smoother trajectory tracking. The baseline approach does not account for the estimation uncertainty and results in higher formation RMSE (see Figure 4) and unreliable control. This aspect is also highlighted in the accompanying video. Note that the the formation error magnitude is influenced by the estimation error, since we are doing closed-loop navigation.

To demonstrate the scalability of our method, we computed the average time taken by a single agent to solve its localization and formation control sub-problems with increasing number of robots. Since our approach is decentralized, we focus on the computation time of a single agent. To evaluate real-world scalability, we ran the localization and the formation control pipelines associated with one of the simulated agents on the (computationally-constrained) single board computer used in our real experiments. The rest of the simulated agents (and their processes) were run on a laptop. The range update rate is fixed to 10 Hz. We evaluated two communication paradigms: (i) clique, where each agent communicates with a maximum of four neighbors, and (ii) mesh, where each agents communicates with every other in the team. We stress here that such a mesh network is not needed for our proposed method, but presented here as the worst-case communication scenario. The average compute time with increasing number of agents in shown in Figure 6 (left). The information exchange between the agents is governed by the range update rate. The average compute time of a single agent (in a group of three agents) with increasing range update rate is shown in Figure 6 (right). The results show that our approach is scalable as the compute time does not increase significantly with increasing number of agents. Compute times for formation control follow a similar trend. Qualitative results showing multiple formations with varying number of agents can be found in the accompanying video.

B. Real-world Experiments

The MAVs used in our experiments are equipped with a Qualcomm Flight Pro single board computer (Quad-core

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

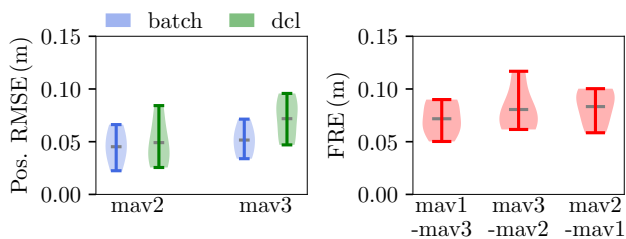


Fig. 7: Localization (left) and formation (right) error distributions from five real experiments where the leader is commanded a straight line path. The proposed decentralized cooperative localization (dcl) method has positioning RMSE close to the centralized batch trajectory estimation. The proposed formation control approach achieves the desired formation as indicated by the formation RMSE (FRE) distributions.

2.15 GHz CPU with 4GB RAM) with an onboard camera and an IMU, a Loco-Positioning UWB node ($\sigma_r = 0.1\text{m}$), and a Pixracer flight control unit (FCU) running the PX4-Autopilot firmware. We use UWB radios in two-way ranging (TWR) mode, where only two agents can measure distance to each other at any given time. To facilitate range measurements across multiple agents, we use time division to share the UWB channel: each agent operates its UWB radio in a pre-specified time slot to measure the distance to its neighbors. The range measurement rate for the entire team is set to 12 Hz. The onboard VIO generates pose estimates at 30 Hz. As before, for safety, we use a downward-facing altimeter to enforce constant altitude constraint to avoid collisions with uneven terrains and bushes in the environment. We use setpoints calculated using our proposed method, that are subsequently sent to the FCU's onboard controller. Each MAV runs the necessary processes, including the localization and formation control pipelines, on the onboard computer. The sensor data is collected for offline comparison with the batch approach. We evaluated our approach through 22 flight experiments in three different environments. Our proposed method was able to successfully execute formation flights in all of the experiments.

Our first test environment is a flight arena equipped with a motion capture system, which is used as the source of ground-truth position. We performed multiple leader-follower flights where the leader agent is commanded a straight path and a circular path. We discuss results for the straight path first. The leader agent (mav1) uses the motion capture system for tracking the trajectory its commanded, while the follower agents (mav2 and mav3) estimate their state using our proposed method. The follower agents calculate the necessary control inputs to maintain a pre-specified triangular formation. Box plots showing position RMSE from five such leader-follower flights are shown Figure 7. Qualitative results showing a subset of the estimated trajectories are shown in Figure 8 (left). We see that our proposed decentralized cooperative localization (dcl) approach achieves positioning RMSE similar to centralized (batch) trajectory estimation. Note that the proposed dcl method uses only a subset of measurements in the fixed-lag smoother, whereas the batch approach uses the entire history of measurements. These results are promising as the UWB radios used in our experiments have a range precision of $\pm 0.1\text{m}$. The bounded formation RMSE in Figure 7 shows that the proposed method is able to maintain the desired formation. The maximum linear speed of the agents in these

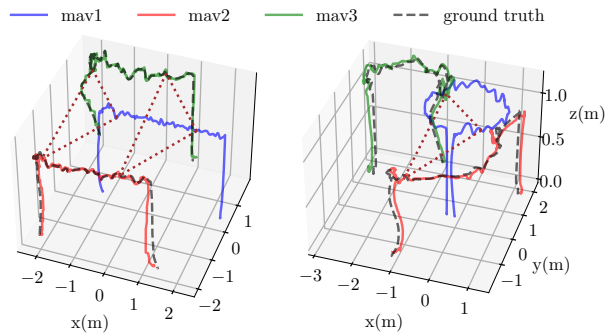


Fig. 8: Estimated and ground-truth trajectories from real flights. The leader agent (mav1) tracks a straight path (left) and a circular path (right), and the other agents (mav2 and mav3) estimate their state and the control inputs to follow the leader in a triangular formation (shown by the red dotted lines).

experiments was 0.4 m/s. Note that our proposed formation control approach is reactive in nature: each agent calculates the necessary control inputs after receiving states from its neighbors. As such, computation and communication-related delays result in larger tracking error at higher speeds. Qualitative results demonstrating our approach at higher speeds can be found in the accompanying video. Unlike in simulation, we found that in real experiments, the baseline formation controller (gcm) was very susceptible to computation and communication-related delays, which resulted in unreliable tracking behavior.

Next, we discuss results from the second set of experiments where the leader agent is commanded a circular path. The average localization and formation RMSE from three such experiments are shown in Table I. Qualitative results showing the estimated trajectories from one such experiment are shown in Figure 8 (right). The localization RMSE is similar to the previous case, however, the formation RMSE is on the higher end. Additionally, with a circular trajectory, minor oscillations in the leader's position can lead to bigger oscillations in other agents' positions for a larger radius.

Qualitative results for the experiments below can be found in the accompanying video. The high-bandwidth nature of UWB allows measuring distances across obstacles, such as foam, cardboard, and other non-metallic objects, where visual line of sight (LOS) is obstructed but radio LOS persists. We found that the foam and cardboard had minimal impact on the ranging accuracy. Our method can execute formation flights in such settings as well. Our second test environment is a metal dome, where three MAVs collaboratively navigate in a triangular formation to inspect a simulated crash site. This environment is challenging as there is no GPS (see Figure 1). We also evaluated our approach in an outdoor environment

TABLE I: Localization and formation control RMSE from three experiments where the leader is commanded a circular trajectory. (Left) The mean and standard deviation (std) of positioning RMSE from the proposed approach (dcl) are comparable to the centralized (batch) approach. (Right) The formation RMSE (FRE) indicates that the proposed approach achieves the desired formation.

	Position RMSE		Formation RMSE	
	mean \pm std (m)		mean \pm std (m)	
	dcl	batch	mav1-mav2	0.20 \pm 0.02
mav2	0.10 \pm 0.01	0.09 \pm 0.02	mav1-mav3	0.22 \pm 0.03
mav3	0.10 \pm 0.01	0.10 \pm 0.03	mav2-mav3	0.10 \pm 0.01

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

under a tree canopy (where GPS can be unreliable) in the presence of wind disturbances.

VI. CONCLUSION

In this work, we presented an approach to joint range-aided cooperative localization and distance-based formation control that is truly decentralized and distributed. We validated our approach through real experiments involving resource-constrained micro aerial vehicles executing formation flights in multiple diverse environments. The results show that our proposed localization and formation control methods achieve decimeter-level positioning and formation accuracy. For future work, we plan to investigate increasing the speed of the formation flights and incorporating obstacle avoidance in the factor graph framework.

REFERENCES

- [1] Soon-Jo Chung, Aditya Avinash Paranjape, Philip Dames, Shaojie Shen, and Vijay Kumar. A Survey on Aerial Swarm Robotics. *IEEE Transactions on Robotics*, 34(4):837–855, August 2018.
- [2] A. Bahr, M.R. Walter, and J.J. Leonard. Consistent cooperative localization. In *Proc. of the International Conference on Robotics and Automation (ICRA)*, pages 3415–3422, Kobe, May 2009. IEEE.
- [3] Michael Hamer and Raffaello Dandrea. Self-Calibrating Ultra-Wideband Network Supporting Multi-Robot Localization. *IEEE Access*, 6:22292–22304, 2018.
- [4] Roberto Tron, Justin Thomas, Giuseppe Loianno, Kostas Daniilidis, and Vijay Kumar. A Distributed Optimization Framework for Localization and Formation Control: Applications to Vision-Based Measurements. *IEEE Control Systems*, 36(4):22–44, August 2016.
- [5] J.P. Desai, J. Ostrowski, and V. Kumar. Controlling formations of multiple mobile robots. In *Proc. of the International Conference on Robotics and Automation (ICRA)*, volume 4, pages 2864–2869, Leuven, Belgium, 1998. IEEE.
- [6] Abhishek Goudar, Wenda Zhao, and Angela P. Schoellig. Range-Visual-Inertial Sensor Fusion for Micro Aerial Vehicle Localization and Navigation. *IEEE Robotics and Automation Letters*, 9(1):683–690, January 2024.
- [7] Frank Dellaert. Factor Graphs: Exploiting Structure in Robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 4(1):141–166, May 2021.
- [8] Stergios I. Roumeliotis and George A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, 2002.
- [9] Keith Y.K. Leung, Timothy D. Barfoot, and Hugh H.T. Liu. Distributed and decentralized cooperative simultaneous localization and mapping for dynamic and sparse robot networks. In *Proc of the International Conference on Robotics and Automation (ICRA)*, pages 3841–3847. IEEE, 2011.
- [10] Luis C. Carrillo-Arce, Esha D. Nerurkar, Jose L. Gordillo, and Stergios I. Roumeliotis. Decentralized multi-robot cooperative localization using covariance intersection. In *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, pages 1412–1417, 2013.
- [11] Charles Champagne Cossette, Mohammed Ayman Shalaby, David Saussié, and James Richard Forbes. Decentralized state estimation: An approach using pseudomeasurements and preintegration. *The International Journal of Robotics Research*, 43(10):1573–1593, September 2024.
- [12] Andrew Howard, Maja J. Matarić, and Gaurav S. Sukhatme. Localization for Mobile Robot Teams: A Distributed MLE Approach. In Bruno Siciliano and Paolo Dario, editors, *Experimental Robotics VIII*, volume 5, pages 146–155. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.
- [13] Nikolas Trawny, Stergios I. Roumeliotis, and Georgios B. Giannakis. Cooperative Multi-Robot Localization under Communication Constraints. In *Proc. of the International Conference on Robotics and Automation (ICRA)*, pages 4394–4400. IEEE, 2009.
- [14] Esha D. Nerurkar, Stergios I. Roumeliotis, and Agostino Martinelli. Distributed Maximum a Posteriori Estimation for Multi-Robot Cooperative Localization. In *Proc. of the International Conference on Robotics and Automation (ICRA)*, pages 1402–1409. IEEE, 2009.
- [15] Alexander Cunningham, Vadim Indelman, and Frank Dellaert. DDF-SAM 2.0: Consistent distributed smoothing and mapping. In *Proc. of the International Conference on Robotics and Automation (ICRA)*, pages 5220–5227. IEEE, 2013.
- [16] Changbin Yu, B. Fidan, and Julien Hendrickx. Rigid graph control architecture for autonomous formations: Applying classical graph theory to the control of multiagent systems. *IEEE Control Systems Magazine*, 28(6):46–63, 2008.
- [17] Laura Krick, Mireille E. Broucke, and Bruce A. Francis. Stabilization of infinitesimally rigid formations of multi-robot networks. In *Proc. of the 47th IEEE Conference on Decision and Control (CDC)*, pages 477–482, Cancun, Mexico, 2008. IEEE.
- [18] Matthew Turpin, Nathan Michael, and Vijay Kumar. Decentralized formation control with variable shapes for aerial robots. In *Proc. of the International Conference on Robotics and Automation*, pages 23–30, St Paul, MN, USA, May 2012. IEEE.
- [19] Reza Babazadeh. *Distance-Based Formation Control of Multi-Agent Systems*. PhD thesis, Concordia University, Montreal, Quebec, Canada, December 2021.
- [20] Mustafa Mukadam, Jing Dong, Xinyan Yan, Frank Dellaert, and Byron Boots. Continuous-time Gaussian process motion planning via probabilistic inference. *International Journal of Robotics Research*, 37(11):1319–1340, 2018.
- [21] Duy-Nguyen Ta, Marin Kobilarov, and Frank Dellaert. A factor graph approach to estimation and model predictive control on Unmanned Aerial Vehicles. In *Proc. of the International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 181–188, Orlando, FL, USA, May 2014. IEEE.
- [22] Dinesh Thakur, Yuezhan Tao, Rebecca Li, Alex Zhou, Alex Kushleyev, and Vijay Kumar. Swarm of Inexpensive Heterogeneous Micro Aerial Vehicles. In Bruno Siciliano, Cecilia Laschi, and Oussama Khatib, editors, *Experimental Robotics*, volume 19, pages 413–423. Springer International Publishing, Cham, 2021.
- [23] Parker C. Lusk, Xiaoyi Cai, Samir Wadhwan, Aleix Paris, Kaveh Fathian, and Jonathan P. How. A Distributed Pipeline for Scalable, Deconflicted Formation Flying. *IEEE Robotics and Automation Letters*, 5(4):5213–5220, October 2020.
- [24] Hao Xu, Yichen Zhang, Boyu Zhou, Luqi Wang, Xinjie Yao, Guotao Meng, and Shaojie Shen. Omni-Swarm: A Decentralized Omnidirectional Visual-Inertial-UWB State Estimation System for Aerial Swarms. *IEEE Transactions on Robotics*, 38(6):3374–3394, 2022.
- [25] Steven Van Der Helm, Mario Coppola, Kimberly N. McGuire, and Guido C. H. E. De Croon. On-board range-based relative localization for micro air vehicles in indoor leader-follower flight. *Autonomous Robots*, 44(3-4):415–441, March 2020.
- [26] Thien-Minh Nguyen, Zhirong Qiu, Thien Hoang Nguyen, Muqing Cao, and Lihua Xie. Distance-Based Cooperative Relative Localization for Leader-Following Control of MAVs. *IEEE Robotics and Automation Letters*, 4(4):3641–3648, October 2019.
- [27] Thien Minh Nguyen, Zhirong Qiu, Thien Hoang Nguyen, Muqing Cao, and Lihua Xie. Persistently Excited Adaptive Relative Localization and Time-Varying Formation of Robot Swarms. *IEEE Transactions on Robotics*, 36(2):553–560, 2020.
- [28] Kun Cao, Muqing Cao, and Lihua Xie. Similar Formation Control via Range and Odometry Measurements. *IEEE Transactions on Cybernetics*, pages 1–12, 2023.
- [29] Xin Zhou, Xiangyong Wen, Zhepei Wang, Yuman Gao, Haojia Li, Qianhao Wang, Tiankai Yang, Haojian Lu, Yanjun Cao, Chao Xu, and Fei Gao. Swarm of micro flying robots in the wild. *Science Robotics*, 7(66), 2022.
- [30] Timothy D. Barfoot. *State Estimation for Robotics: Second Edition*. Cambridge University Press, January 2024.
- [31] Yetong Zhang, Gerry Chen, Adam Rutkowski, and Frank Dellaert. Efficient Range-Constraint Manifold Optimization with Application to Cooperative Navigation. In *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, volume 2022-Octob, pages 9950–9956. IEEE, 2022.
- [32] Frank Dellaert and GTSAM Contributors. borglab/gtsam. <https://github.com/borglab/gtsam>, May 2022.