Coordination of a Flock of Drones using Ephemeral Environment Learning

Andreas Brandstätter

March 9, 2023

Research report submitted to Austrian Marshall Plan Foundation by Andreas Brandstätter (Research Division for Cyber-Physical Systems, TU Wien, Austria) as part of research exchange at Department of Computer Science, Stony Brook University, USA. Supervised by Scott A. Smolka (Department of Computer Science, Stony Brook University, USA), Scott D. Stoller (Department of Computer Science, Stony Brook University, USA), Ashish Tiwari (Microsoft, USA), and Radu Grosu (Research Division for Cyber-Physical Systems, TU Wien, Austria).

Abstract

We introduce a method to control Multi-agent systems (MASs) only based on relative distance measurements. This means our approach is able to work in GPS-denied environments. It is fully distributed and therefore does not need any information exchange between the individual agents. We evaluate this aproach on the problem of drone flocking, thereby demonstrating that our system is able to form and maintain a flock of drones. Relative distance measurements to other drones and information about its own relative movement are used to estimate the current state of the environment. This makes it possible to perform lookahead and estimate the next state for any potential next movement. A distributed cost function (composted of separation, cohesion and target-finding) is then used to determine the best next action in every time step. Using a high-fidelity simulation environment, we show that our approach is able to form and maintain a flock for a set of drones.

Contents

1 Introduction

Multi-agent systems (MASs) can collaboratively perform tasks that individual agents are hardly capable of. For search-and-rescue (SAR) applications, there is a wide variety of multi-agent approaches using ground-, aerial-, surface-, and underwater-vehicles [1]–[3]. Further scenarios for multi-agent systems include transportation [4], [5] and structural inspection [6], [7]. Environmental monitoring, space exploration, agriculture, entertainment, and industrial maintenance are further application areas of MASs [8]. All these applications inherently need some type of coordination and control method to keep the desired formation of the MAS.

Using a global navigation satellite system (GNSS), agents are aware of their absolute position information and can use this for coordination. For indoor applications there are comparable types of localization systems which also provide a global reference frame [9], [10]. However there are still scenarios (e.g. SAR, underwater, mining, industrial exploration) where no localization system is available to be used by the MAS. We therefore study the coordination of MASs solely based on inter-agent-distances, which can be gathered by onboard sensors, and local information measured by an Inertial Measurement Unit (IMU). In two of our case-studies we describe a modified version of our control approach by additionally including altitude sensing capabilities.

1.1 Drone Flocking

Our primary application of this new method are drone swarms, which are a quintessential example of a multi-agent systems. They can carry out tasks that cannot be accomplished by individual drones alone [11]. They can, for example, collectively carry a heavy load while still being much more agile than a single larger drone [4], [5]. In search-and-rescue applications, a swarm can explore unknown terrain by covering individual paths that jointly cover the entire area [2], [3], [12]. While flocking provides a number of advantages over individual flight, it also poses a significant challenge: the need for a distributed control mechanism that can maintain flock formation and its stability [13]. These collective maneuvers can be expressed as the problem of minimizing a *positional cost function*, i.e., a cost function that depends on the positions of the drones (and possibly information about their environment). In our formulation, every agent is identical, which means there is no designated leader.

To work with such a positional cost function, an absolute localization system is needed. This can be an optical or radio-based system for indoor applications or GPS-based localization for outdoor scenarios. In this work, we study the problem for scenarios that lack an absolute localization system (GPS-denied environments, as mentioned earlier). We only have the ability to measure the distance to other drones and to measure the acceleration and rotational velocity of the own drone using an onboard Inertial Measurement Unit (IMU). For flock formation, we observe that the positional cost function can be replaced by a function based solely on relative distances. This obviates the need for absolute

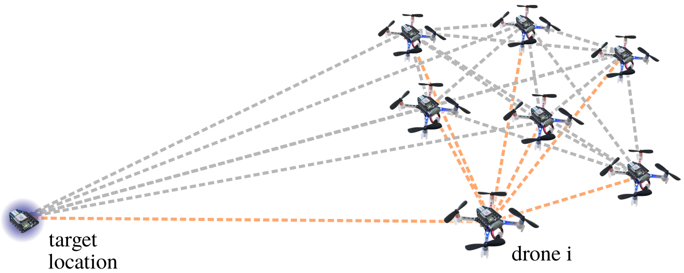

Figure 1: Our distributed controller forms and maintains a flock based on relative distance measurements to other agents of the flock. The target location is shown in blue. Distance measurements for drone i to other drones and to the target location are shown in orange.

localization. We propose a method to simultaneously learn properties of the environment (inter-agent distance changes), while at the same time maintaining the flock formation solely on relative distance information.

In this research report, we address the following Challenge Problem: Design a distributed controller that controls a set of MASs based solely on inter-agent distance measurements.

To solve this problem, we introduce a method to estimate changes of the environment based on the observed changes for previous movements and thereafter use this information to minimize the cost-function over a set of candidate positions. We build upon our previous work that introduced Spatial Predictive Control (SPC) [14] to select the best next action from the set of candidate positions. However we have a substantially different problem here, since we have limited observation capability: in the previous work [14], absolute positions of all the drones were available; whereas in this work we can only measure relative distances. This also changes our possibilities how to apply SPC: whereas in the previous work it was possible to optimize the direction based on the cost function's gradient, we need to do a search on possible candidate positions in all directions in this work.

For case studies Aerial-A and Aerial-G, our agent's observations consist of its own acceleration in three-dimensional space, rotational velocity along three axes, and the relative distance to other agents, as well as the distance to a fixed target location (as shown in Figure 1). (The target location is currently only used to counteract drifting tendencies of the whole flock.) There is no communication or central coordination, which makes our approach fully distributed. Our flocking objective is formulated as a cost function (see Section 3.3) which is based on these distance measurements. The corresponding action of each agent is a relative spatial vector, to which the drone should move, to minimize its cost function's value.

1.2 Research Report Outline

Section 2 introduces our method of ephemeral environmental learning, which collects information about local movements, changes in measurements and saves this information for the estimation step. Section 3 describes the scenarios of our case studies, introduces cost functions for flocking with target seeking and related performance metrics. Section 4 introduces our method to make use of this environmental knowledge and thereafter describes our distributed flocking controller. Section 5 presents the results of our experimental evaluation. Section 6 considers related work. Section 7 offers our concluding remarks.

2 Estimating Distance Changes Based on Sensor Measurements

We describe the procedure from the perspective of agent i. The measured sensor data for agent i consists of the current distances to the neighboring agent (and the fixed target) and local IMU data. To represent the relevant data history, agent i keeps two matrices, a (3×3)-matrix $\mathcal{M}$ and a $(3 \times (|D| + 1))$-matrix $\mathcal{P}$. The j-th row of $\mathcal{M}$ is a displacement vector for agent i. The j-th row of $\mathcal{P}$ is a vector of change in distances of every other agent and the target to agent i (as seen by agent i when it moved by displacement vector in j-th row of $\mathcal{M}$). In particular, $\mathcal{P}_{kj}$ is the change in distance of agent j (or target if $j = |D| + 1$) as seen by agent i when it moved by the vector $\mathcal{M}_{k*}$. The notation $\mathcal{M}_{k*}$ denotes the k-th row vector of matrix $\mathcal{M}$. Let us see how the matrices $\mathcal{M}$ and $\mathcal{P}$ are populated.

2.1 Data accumulation

Using the onboard IMU, each agent is capable of measuring its own acceleration vector in three dimensions $\vec{a_i}$. By integration, the velocity vector $\vec{v_i}$ can be derived. Agent i continuously updates the matrices $\mathcal{M}$ and $\mathcal{P}$ as follows:

1. Save the observations of time instant t. Let $d_{ij,t}$ denote the distances to agent j, and let $l_{i,t}$ denote the distance to the fixed target, at this time instant t (as obtained from the sensors).

$$d_{ij,t} = d_{ij} \,|\, j \in H_i, t \tag{1}$$

$$l_{i,t} = l_i \,|\, t \tag{2}$$

2. Integrate velocity vector to keep track of its own position changes, which gives the displacement vector $\vec{u_i}$:

$$\vec{u_i} = \int_{t-\Delta t}^{t} \vec{v_i} \, dt \tag{3}$$

3. If the norm of the change in position is larger than a threshold $||\vec{u_i}|| > s_{thr}$, calculate the changes in distances as follows:

$$\Delta d_{ij} = d_{ij,t} - d_{ij,t-\Delta t} \tag{4}$$

$$\Delta l_i = d_{i,t} - d_{i,t-\Delta t} \tag{5}$$

   Here $d_{ij,t-\Delta t}$ denotes the observed distance to agent j at the previous time instant $t - \Delta t$.

   (a) If the length of the displacement vector $\vec{u_i}$ is not larger than the threshold s_{thr}, we go back to Step (2), continuing with measurements.

   (b) If the elapsed time, since the observations were saved, is larger than a threshold $\Delta t > t_{thr}$, we delete $d_{ij,t-\Delta t}$ and $l_{i,t-\Delta t}$ and go back to Step (1). This means the agent did not move considerably, and we therefore discard these measurements.

4. Select the row k in $\mathcal{M}$, which is most similar to $\vec{u_i}$, by $k = \text{argmax}_{r\in\{1,2,3\}}\{|\mathcal{M}_{r*} \cdot \vec{u_i}|\}$.

   Add the normalized displacement vector $\frac{\vec{u_i}}{||\vec{u_i}||}$ of agent i, replacing row k in matrix $\mathcal{M}$ and add the vector $\frac{1}{||\vec{u_i}||} \cdot \langle \Delta d_{i1}, \dots, \Delta d_{i|D|}, \Delta l_i \rangle$, replacing row k in matrix $\mathcal{P}$.

   Note that we have assumed here that the neighborhood H_i of Drone i is the full set D, but the details can be easily adapted to the case when $H_i \subsetneq D$.

5. The process starts again at (1) and we thus keep updating rows in the matrices $\mathcal{M}$ and $\mathcal{P}$, representing recent displacements of agent i and associated changes in distance measurements.

Note that $\vec{u_i}$ is obtained by double integration and therefore it is prone to acceleration sensing errors, and also numerical errors. This influence is however limited, since integration times Δt are small (bounded by t_{thr}).

2.2 Exploration

As the matrices $\mathcal{M}$ and $\mathcal{P}$ are initially empty, agents first need to perform some exploration. This is only necessary initially, as in the later process the matrices are also updated during control actions. Exploration is done as follows:

1. If all rows of $\mathcal{M}$ and $\mathcal{P}$ are empty: Sample three gaussian-distributed random variables with $\sigma = 1\,m$ and zero mean as the components of vector $\vec{r}$. Apply the vector $q_{expl,1}$ as control action, where w_{expl} is a scaling parameter.

$$q_{expl,1} = w_{expl} \cdot \frac{\vec{r}}{||\vec{r}||} \tag{6}$$

2. Only $\mathcal{M}_{1*}$ is not empty: Sample vector $\vec{r}$ in the same way as above. Apply the vector $q_{expl,2}$ as control action (which is by construction orthogonal to $\mathcal{M}_{1*}$).

$$q_{expl,2} = w_{expl} \cdot \frac{\mathcal{M}_{1*} \times \vec{r}}{\|\mathcal{M}_{1*} \times \vec{r}\|} \tag{7}$$

3. Only $\mathcal{M}_{1*}$, and $\mathcal{M}_{1*}$ are not empty: Apply the vector $q_{expl,3}$ as control action (which is by construction orthogonal to $\mathcal{M}_{1*}$ and $\mathcal{M}_{2*}$).

$$q_{expl,3} = w_{expl} \cdot \frac{\mathcal{M}_{1*} \times \mathcal{M}_{2*}}{\|\mathcal{M}_{1*} \times \mathcal{M}_{2*}\|} \tag{8}$$

4. All entries in $\mathcal{M}$ are non-empty:

   (a) There exist two rows k and $m \neq k$, pointing in a similar direction ($|\mathcal{M}_{k*} \boldsymbol{\cdot} \mathcal{M}_{m*}| > \kappa_{thr}$), where κ_{thr} is a parameter to quantify this similarity. Apply the vector $q_{expl,4}$ as control action (which is by construction orthogonal to these row vectors).

$$q_{expl,4} = w_{expl} \cdot \frac{\mathcal{M}_{k*} \times \mathcal{M}_{m*}}{\|\mathcal{M}_{k*} \times \mathcal{M}_{m*}\|} \tag{9}$$

   (b) If such rows do not exists, the vectors $\mathcal{M}_{1*}$, $\mathcal{M}_{2*}$, $\mathcal{M}_{2*}$ are linearly independent – that is, they are all different from each other ($\mathcal{M}_{1*} \neq \mathcal{M}_{2*} \neq \mathcal{M}_{3*}$, ensured by 4a), nonzero ($\mathcal{M}_{1*} \neq \vec{0}$, $\mathcal{M}_{2*} \neq \vec{0}$, $\mathcal{M}_{3*} \neq \vec{0}$, ensured by $||\vec{u_i}|| > s_{thr}$), and not in a common plane ($(\mathcal{M}_{1*} \times \mathcal{M}_{2*}) \boldsymbol{\cdot} \mathcal{M}_{3*} \neq \vec{0}$, ensured by 2,3, and 4).

2.3 Exploitation

The three vectors $\mathcal{M}_{1*}$, $\mathcal{M}_{2*}$, $\mathcal{M}_{2*}$ form a basis in the three-dimensional space. Using a basis transform it is therefore possible to estimate the change for distances for any movement vector $\vec{u}$. Specifically, if

$$\vec{u} = \lambda_1 \cdot \mathcal{M}_{1*} + \lambda_2 \cdot \mathcal{M}_{2*} + \lambda_3 \cdot \mathcal{M}_{3*} \tag{10}$$

then we can compute the estimated change in distances of each of the other drones, $\widehat{\Delta d_{i*}}(\vec{u})$, and the target, $\widehat{\Delta l_i}(\vec{u})$ for this displacement $\vec{u}$ as follows:

$$\langle \widehat{\Delta d_{i1}}, \ldots, \widehat{\Delta d_{i|D|}}, \widehat{\Delta l_i} \rangle = \lambda_1 \cdot \mathcal{P}_{1*} + \lambda_2 \cdot \mathcal{P}_{2*} + \lambda_3 \cdot \mathcal{P}_{3*} \tag{11}$$

(addition and multiplication in Equation (11) are applied element-wise on the vectors).

Likewise we can compute the estimated distances after the agent i would have moved by displacement vector $\vec{u}$:

$$\widehat{d_{ij}} = d_{ij} + \widehat{\Delta d_{ij}} \tag{12}$$

$$\widehat{l_i} = l_i + \widehat{\Delta l_i} \tag{13}$$

3 Case Studies

In this section we provide a background on flocking, which is the objective of our primary case study. We introduce our cost function for flocking with aerial target seeking, and then present quality metrics of a flock to assess the quality of our controller.

3.1 Drone Flocking (Aerial-A and Aerial-G)

A set of agents, D, is in a flock formation if the distance between every pair of agents is range bounded; that is, the drones are neither too close to each other nor too far apart. Our approach to flock formation is based on defining a cost function such that the agents form a flock when the cost is minimized. The requirement that the inter-agent distance is range bounded is encoded as the first two terms of our cost function, namely the *cohesion* and *separation* terms shown in the next section. Note that the Reynolds rules for forming a flock [15] also include a term for aligning the drone's velocities, apart from the cohesion and separation terms. By not including velocity alignment term, we potentially allow a few more behaviors, such as circling drones, but some of those behaviors are eliminated by our third term, namely the *target seeking* term.

3.2 Underwater groups (Maritime)

While drone flocking is certainly our main application area, the principle described in this report can also be applied to other types of control problems. One such problem is the coordination of groups of unmanned underwater vehicles. Cost function terms of Aerial-A and Aerial-G can be applied analogously if the z-axis is flipped. This means that the altitude above ground becomes the depth of submersion.

3.3 Cost Function for Aerial-A

Consider drones i and j, where $i, j \in D$. Let d_{ij}, when it appears in the local cost function of drone i, denote the distance between drone i and drone j as it appears to drone i; this may differ from the actual distance due to sensing error. Similarly l_i denotes the distance between drone i and the fixed target location p_{tar}. All distances are measured from the center of the mass of the respective drone. Let r_{drone} denote the radius of each drone (specifically the radius of the circumscribed sphere including propellers). In our formulation for the cost function, drone i has access to distances of only a subset $H_i \subseteq D$ of drones, namely its local neighbors. Hence, we define a local cost function, parameterized by i, which uses only the distances to drones in H_i. Using the neighborhood radius parameter r_H, we define H_i as follows:

$$H_i = \{j \,|\, d_{ij} < r_H, j \in D \setminus \{i\}\} \tag{14}$$

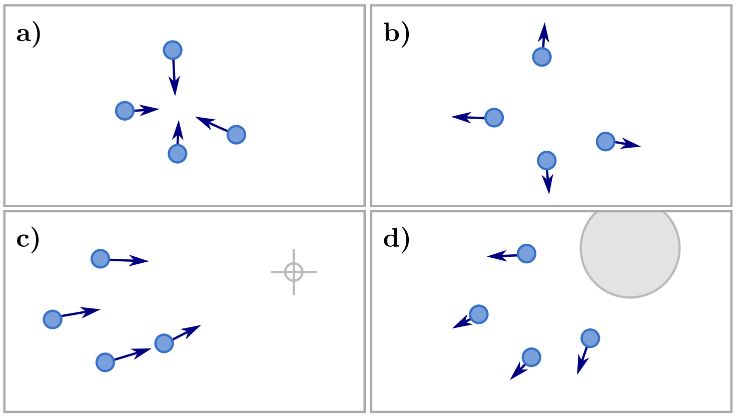

Figure 2: Directional movements (indicated by arrows) induced by cost-function terms: **a**: *Cohesion*, **b**: *separation*, **c**: *target seeking*, and **d**: *obstacle avoidance* (currently not implemented in our case-study).

Let d_{H_i} denote the tuple of distances from drone i to drones in H_i. The cost function c we use in AERIAL-A is defined for every drone $i \in D$ as in Equation (15).

$$c(d_{H_i}, l_i) = c_{coh}(d_{H_i}) + c_{sep}(d_{H_i}) + c_{tar}(l_i) \tag{15}$$

The value of the *cohesion* term increases as drones drift apart, and the *separation* term increases as drones get closer. Each term has a weight, denoted by a subscripted ω.

Cohesion term:

$$c_{coh}(d_{H_i}) = \frac{\omega_{coh}}{|H_i|} \cdot \sum_{j \in H_i} {d_{ij}}^2 \tag{16}$$

Separation term:

$$c_{sep}(d_{H_i}) = \frac{\omega_{sep}}{|H_i|} \cdot \sum_{j \in H_i} \frac{1}{max(d_{ij} - 2r_{drone} - \chi_{sep}, \hat{0})^2} \tag{17}$$

Here $\hat{0}$ denotes a very small positive value. The function $max(., \hat{0})$ ensures the denominator remains nonzero, especially because sensor noise can cause distance measurements to have errors. We introduced the slack parameter χ_{sep} which can be modified to influence the minimum distance between two drones. This parameter is different from previous works, such as [15]. During experimental validation, this parameter showed significant performance improvements.

To prevent the flock from moving in random directions in AERIAL-A, we use a *aerial target seeking* term with a fixed target location, denoted by p_{tar}, for the entire flock. Here l_i denotes the distance between the center of drone i and the fixed target location p_{tar}.

Aerial target seeking term:

$$c_{tar}(l_i) = \omega_{tar} \cdot l_i \tag{18}$$

With only *cohesion* and *separation*, the whole flock would form and move in random directions and random locations in absolute world coordinates. Our *aerial target seeking* term avoids this behaviour. All drones use the same target location; thus, this last term assumes shared global knowledge of the target. The control algorithm will still be fully distributed. A way to avoid having a fixed target location would be to designate one of the drones as the leader of the flock. This leader could be equipped with additional sensors to get information about its absolute position, allowing it to employ a different control scheme. As the availability of absolute position information would contradict our main statement of the problem formulation, we alternatively describe the case Aerial-G with an alternative target seeking term.

Calm term:

$$c_{calm}(\vec{v}) = \omega_{calm} \cdot \|\vec{v}\| \tag{19}$$

This optional term avoids small noisy movements of the drones, as it imposes a cost proportional to the velocity vector. Note, that the calm term is not used in our total cost function. In our experiments, it turned out that it does not have a beneficial effect.

3.4 Cost Function for Aerial-G

As described in the previous section, the drones can be supplied with a target term that aims to reach an artificial place which must not be somewhere in the air, as the drones would be located around this position. This is impractical for actual hardware implementation and therefore we describe an alternative target seeking term, if the target location is on the ground. Note, that for this cost function term, there is additional sensing capability for the absolute altitude of the individual drones needed.

Ground target seeking term:

$$c_{tar}(l_i, z_i) = \omega_{tar} \cdot \sqrt{max({l_i}^2 - {z_i}^2, 0)} \tag{20}$$

Where z_i denotes the z-coordinate of the position of agent i, which is it's altitude.

Elevation term:

$$c_{elev}(z_i) = \omega_{elev} \cdot (\zeta_{elev} - z_i)^2 \tag{21}$$

This term keeps the flock at a certain altitude, which is determined by the reference elevation ζ_{elev}.

3.5 Flock-Formation Quality metrics

We define two quality metrics to assess the quality of the flock formation achieved by a flocking controller. To compute these quality metrics, we assume to have access to full ground truth information about the absolute positions of the drones. The position (center of mass) of drone i is denoted by p_i.

Collision avoidance: To avoid collisions, the distance between all pairs of drones $\mathrm{dist}(D)$ must remain above a specified threshold dist_{thr}. We define the metric for the minimum distance between any pair of drones as follows:

$$\mathrm{dist}(D) = \min_{i,j \in D; i \neq j} \|p_i - p_j\| \tag{22}$$

$$\mathrm{dist}(D) \geq \mathrm{dist}_{thr} \tag{23}$$

We set $\mathrm{dist}_{thr} = 2 \cdot r_{drone} + r_{safety}$, where r_{safety} is a parameter for the safety margin.

Compactness: Compactness of the flock is determined by the flock *radius*. Radius is defined as the maximum distance of any drone from the centroid of the flock:

$$\mathrm{radius}(D) = \max_{i \in D} \left\| \frac{\sum_{j \in D} p_j}{|D|} - p_i \right\| \tag{24}$$

The drones are said to be in a compact flock formation if $\mathrm{radius}(D)$ stays below some threshold radius_{thr}; otherwise the drones are too far apart, not forming a flock.

$$\mathrm{radius}(D) \leq \mathrm{radius}_{thr} \tag{25}$$

The value for radius_{thr} is picked based on the drone model and other parameters governing the flock formation problem. We set it to $\mathrm{radius}_{thr} = F \cdot r_{drone} \cdot \sqrt[3]{|D|}^{-1}$, where we use the drone radius r_{drone} to incorporate the physical size and multiply by a factor F.

4 Distributed Flocking Controller using Relative Distances

In our distributed approach to flock formation, each drone picks the best action at every time step. The action here is a target displacement vector. Each drone picks the optimal displacement vector for itself by looking ahead in different spatial directions and finding a location that would minimize the cost *if this drone moved there.* To perform this search, each drone needs capability to *estimate* the relative distances to other drones when it moves to different potential target locations. To perform this estimation, each drone stores some measurements from past time steps, which is described in Section 2. Thereafter, Section 4.1 shows how this stored knowledge is used by each drone to estimate relative distances of other drones for different possible displacements of itself.

4.1 Distributed flocking controller

We now describe our control approach based on the cost function introduced in Section 3.3 and on the environmental knowledge representation described in Section 2.

The set of candidate positions Q is defined as follows:

$$Q = \left\{ \epsilon_Q \cdot n \cdot \frac{\vec{q}}{\|q\|} \,\middle|\, \vec{q} \in Q_E, n \in \{1, \ldots, N_Q\} \right\} \cup \vec{0} \tag{26}$$

$$Q_E = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \,\middle|\, x, y, z \in \{-1, 0, 1\} \right\} \setminus \vec{0} \tag{27}$$

$\vec{0}$ is used as shorthand notation for $(0, 0, 0)^T$.

This gives a set of $N_Q \cdot 26 + 1$ points which are spaced by a distance of ϵ_Q each in each direction, including diagonals. Over this set Q the best action q_{next} is searched by minimizing the cost function c:

$$q_{next} = \underset{q \in Q}{\operatorname{argmin}} \{c(\widehat{d}_{i*}(q), \widehat{l}_i(q), q)\} \tag{28}$$

If two candidate positions q_1 and q_2 both have the same minimum value for the cost function c, our implementation of argmin takes the last one based on the implementation of the enumeration. The function $\widehat{d}_i(q)$ estimates the distances to drones, where $\widehat{l}_i(q)$ estimates the distance to the target, if the action q is applied. For each $q \in Q$, the vector $\widehat{d}_i(q)$ (and the value $\widehat{l}_i(q)$) is calculated by first computing the estimates of the *change vector* $\bar{d}_i(q)$, and the change $\bar{l}_i(q)$ using Equation 11. Now the distances can be estimated by just adding the estimated change to the currently measured distances d_{i*} and l_i:

$$\widehat{d}_i(q) = d_{i*} + \bar{d}_i(q) \tag{29}$$

$$\widehat{l}_i(q) = l_i + \bar{l}_i(q) \tag{30}$$

Each drone minimizes its local cost function (Eq. 28) in order to recompute the desired set-point at every time step. As we similarly did in [14], this set-point is then handed off to a low-level controller that controls the thrust of the drone's motors so that it steers towards this set-point.

4.2 Non-collision and non-dispersion

Using our method we cannot give formal guarantees on non-collision and non-dispersion of the flock, but we can informally reason about the behaviour provide a foundation for potential further analysis.

We show why non-collision holds for two agents:

1. Let us assume, the set-point for velocity which we give to the LLC is actually applied to the drones (as given by control laws of the LLC). Therefore we assume for the further analysis that the set-point velocity equals the actual velocity.

2. Case distinction by relative velocity to each other: If they are moving away from each other, non-collision holds. If relative velocity equals 0, it also holds.

3. In case the agents are approaching each other, we make another case distinction. If the agents are further away than the equilibrium distance (determined by separation and cohesion weight factors), the controller correctly commands the agents to come closer to each other. They will eventually come closer than the equilibrium distance, which leads to the second case.

4. At least one drone must have nonzero velocity (as they are approaching each other). Therefore the integration of the velocity will eventually lead to a norm of the change in position that is larger than the threshold (Section 2, Step 3). This leads to an updated displacement vector in the direction of the current movement. By construction the state-estimator will therefore estimate the change in distances (if continuing in this direction) to decrease. Therefore the controller (Eq. 28) will apply set-point velocities that move the agents away from each other.

Reasoning for non-dispersion is analogous to non-collision. It just needs to switch directions for respective movements.

5 Experiments

We evaluated our method using simulation experiments. The goal of the experiments was to investigate and demonstrate the ability to form and maintain a stable flock while holding position at a target location.

5.1 Simulation Experiments

As a simulation framework, we use *crazys* [16], which is based on the *Gazebo* [17] physics and visualization engine and the Robotic Operating System (ROS) [18]. Our algorithm is implemented in C++ as a separate ROS node. As shown in Figure 3, it receives the measured distances to neighboring drones,

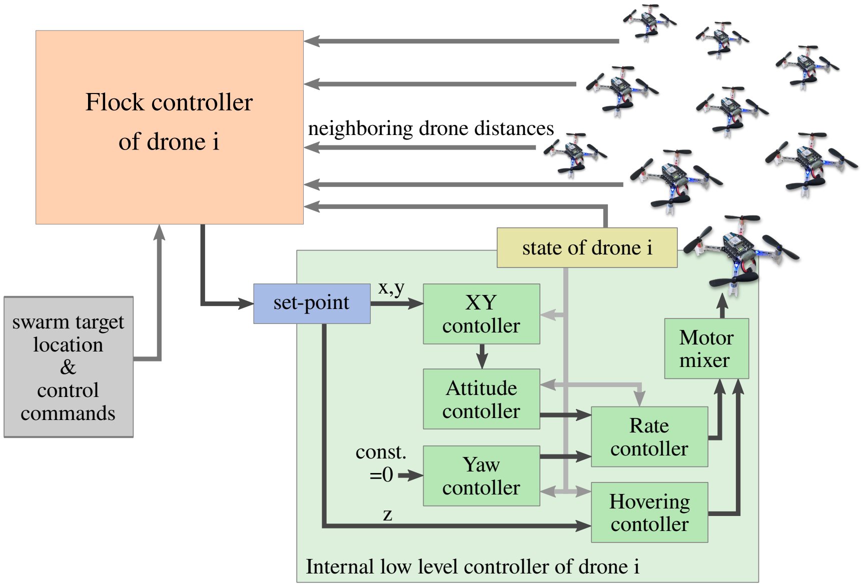

Figure 3: The ROS-node of the SPC controller for drone i receives distance measurements to neighboring drones and control messages (e.g. swarm target location, start/stop command). It outputs the set-point for the internal low level controller.

and control messages, such as the target location or a stop command, from the human operator. It calculates the best next action according to Equations (26)-(30). The parameter ϵ_Q is determined empirically and fixed throughout the whole simulation. Auxiliary functions, like hovering at the starting position, and landing after the experiments are finished, are also implemented in this node.

In order to evaluate the control mechanism and its implementation, we fixed the target location, as described above. This avoids drifting behaviour of the whole flock, which could not be detected by relative distance measurements in any way. Simulations were done with flocks of size $|D|=5$, 9, and 15. Figure 4 shows a screenshot of a simulation with 5 drones. All simulations use global neighborhood ($H_i = D$) for now.

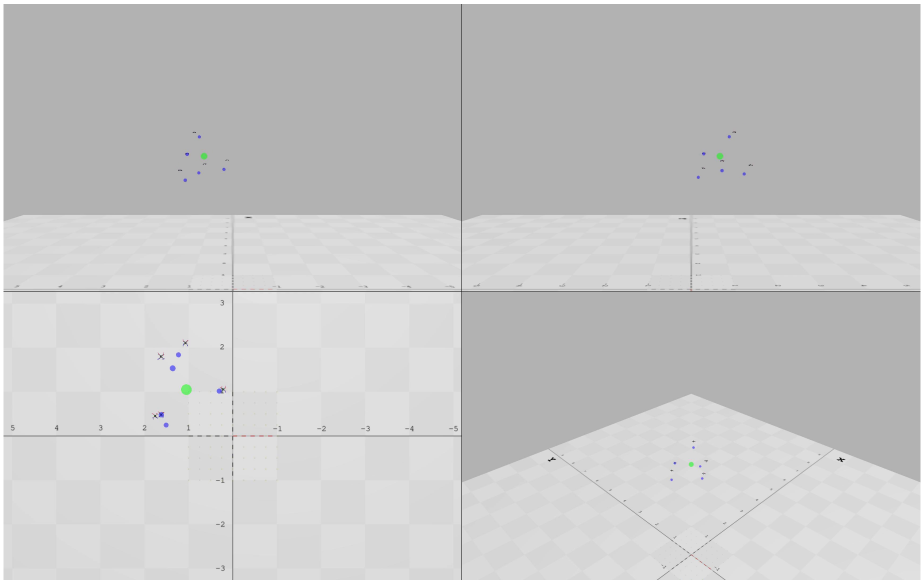

Figure 4: Screenshot of the end of the simulation with 5 drones. Shown from four different camera views after the flock reached its target. The green dot indicates the target location. The blue dots visualize the next action which is supplied to the lower level controller.

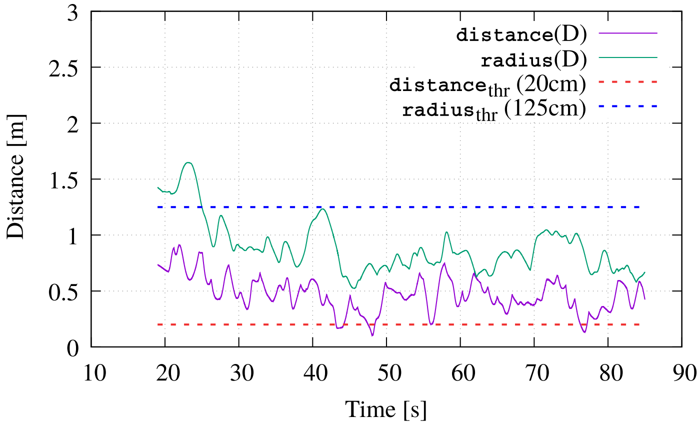

Figure 5: Quality metrics for simulation with 5 drones. Threshold dist$_{thr}$ for collision avoidance is satisfied most of the time. After settling in, the swarm radius remains below the threshold radius$_{thr}$, thus showing the ability to form a compact flock in the simulation. (Quality metric recordings start at $t = 19\,s$ after initialization procedure.)

5.2 Results

Early results show that our approach is able to properly form and maintain a flock with only relative position measurements. Figure 5 shows performance metrics over time for a simulation with 5 drones. The analysis of the quality metrics for *collision avoidance*, and *compactness* show that our control approach successfully maintains a stable flock (threshold dist$_{thr}$ is only violated for very short moments). Note that these results are already obtained before extensive controller tuning. Using carefully adjusted values for ω_{coh} and ω_{sep} should lead to even better results and maintain the threshold throughout the whole simulation.

6 Related Work

Relative localization is studied in various works with focus on different aspects. Agents equipped with cameras and/or LIDAR sensors are able to use this information for localization, which is studied in [19]–[22]. However in our work such sensors are not available. Other works [23]–[26] use relative position measurements to other agents, which provide more information than only distance measurements. In [27] agents engaging in a combination of circular motion and linear motion are described of, but these agents are restricted to a 2D plane only. Distributed formation control is studied in [23], [28], but these works also limited to 2D. If agents are able to interchange messages they can communicate their distance measurements, acceleration, and possibly further information. In

[29]–[33], the proposed method uses message interchange (transmitting acceleration, angular velocity, or other data) between at least some of the agents, which is not the case in our work. In [34] a centralized approach is presented, which differs from our work, as we are fully distributed. Other works study relative localization for different scenarios: In [35] a single leader moving with constant velocity is tracked by a single follower. In [36] one leading and two following agents are described. The work [37] proposes a control scheme with three alternating periods: identification, control, and resting; where in the last of these phases the agent needs to remains stationary. Formation shape control based on distance measurements, which involves sinusoidal perturbations on the agent's actions, is described in [38]. Frequencies for these perturbations are assumed to be pairwise distinct, which limits the application for larger number of agents, if in practice the allowed range for this frequencies is bounded by the agent's mechanical limitations.

Reynolds [15] was the first to propose a flocking model, using cohesion, separation, and velocity alignment force terms to compute agent accelerations. Reynolds' model was extensively studied [39] and adapted for different application areas [11]. Alternative flocking models are considered in [13], [40], [41], [42], [43], and [44]. In all these approaches, absolute position measurements and/or inter-agent communication were available. In our work, we only work with relative distance measurements and a fully distributed formulation. In addition to these largely theoretical approaches, in [45] and [46], flocking controllers are implemented and tested on real hardware. However, the approach of [46] involves the use of nonlinear model-predictive control (NMPC). In contrast to our work, [45] also requires the velocity of neighboring drones.

7 Conclusions

We introduced a method to control a flock only based on relative position measurements to neighboring drones, and demonstrated its utility on the drone flocking problem. We performed simulation experiments using a physics engine with a detailed drone model. Our results demonstrated the ability to form and maintain a flock, and hold its position on a target location.

7.1 Future work

As we currently have only intermediate results of the experiments with limited number of agents, we plan to do more extensive testing with a wide set of different scenarios, including larger number of drones, and local neighborhood ($H_i \subset D$). Neighborhood might be defined by euclidean distance, or alternatively by topological distance, as introduced in [47]. As further directions of future work, we plan to extend our approach with obstacle avoidance capabilities. We also plan to test it for moving target locations and various path tracking scenarios. To prepare for the transfer to real hardware we plan to introduce sensor noise in the simulation and test the robustness of our method to cope with such disturbances. As next goal it should then be implemented on real drones, specifically, *Crazyflie 2.1*-quadcopters [48].

Acknowledgments

This work was conducted as part of a research exchange at Department of Computer Science, Stony Brook University, USA by Andreas Brandstätter which was supported by Austrian Marshall Plan Foundation. Radu Grosu was partially supported by EU-H2020 Adaptness and AT-BMBWF DK-RES and CPS/IoT Ecosystem. This work was supported in part by NSF grants CCF-1954837, CCF-1918225, and CPS-1446832.

References

[1] J. P. Queralta, J. Taipalmaa, B. Can Pullinen, *et al.*, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191 617–191 643, 2020, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3030190. [Online]. Available: https://ieeexplore.ieee.org/document/9220149/ (visited on 12/01/2022).

[2] D. Câmara, "Cavalry to the rescue: Drones fleet to help rescuers operations over disasters scenarios," in *2014 IEEE Conference on Antenna Measurements Applications (CAMA)*, 2014, pp. 1–4. DOI: 10.1109/CAMA.2014.7003421.

[3] N. Michael, S. Shen, K. Mohta, *et al.*, "Collaborative mapping of an earthquake damaged building via ground and aerial robots," in *Proceedings of 8th International Conference on Field and Service Robotics (FSR '12)*, Jul. 2012, pp. 33–47.

[4] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011. DOI: 10.1007/s10514-010-9205-0.

[5] G. Loianno and V. Kumar, "Cooperative transportation using small quadrotors using monocular vision and inertial sensing," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 680–687, 2018. DOI: 10.1109/LRA.2017.2778018.

[6] S. Mian, T. Garrett, A. Glandon, *et al.*, "Autonomous spacecraft inspection with free-flying drones," in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, San Antonio, TX, USA: IEEE, Oct. 11, 2020, pp. 1–9, ISBN: 978-1-72819-825-5. DOI: 10.1109/DASC50938.2020.9256569. [Online]. Available: https://ieeexplore.ieee.org/document/9256569/ (visited on 12/01/2022).

[7] W. Jing, D. Deng, Y. Wu, and K. Shimada, "Multi-UAV coverage path planning for the inspection of large and complex structures," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA: IEEE, Oct. 24, 2020, pp. 1480–1486, ISBN: 978-1-72816-212-6. DOI: 10.1109/IROS45743.2020.9341089. [Online]. Available: https://ieeexplore.ieee.org/document/9341089/ (visited on 12/01/2022).

[8] M. Schranz, M. Umlauft, M. Sende, and W. Elmenreich, "Swarm robotic behaviors and current applications," *Frontiers in Robotics and AI*, vol. 7, p. 36, Apr. 2, 2020, ISSN: 2296-9144. DOI: 10.3389/frobt.2020.00036. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2020.00036/full (visited on 12/01/2022).

[9] A. F. G. Ferreira, D. M. A. Fernandes, A. P. Catarino, and J. L. Monteiro, "Localization and positioning systems for emergency responders: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2836–2870, 2017, ISSN: 1553-877X. DOI: 10.1109/COMST.2017.2703620. [Online]. Available: http://ieeexplore.ieee.org/document/7927385/ (visited on 12/01/2022).

[10] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, "A survey of enabling technologies for network localization, tracking, and navigation," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3607–3644, 2018, ISSN: 1553-877X, 2373-745X. DOI: 10.1109/COMST.2018.2855063. [Online]. Available: https://ieeexplore.ieee.org/document/8409950/ (visited on 12/01/2022).

[11] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018. DOI: 10.1109/TRO.2018.2857475.

[12] A. Boggio-Dandry and T. Soyata, "Perpetual flight for UAV drone swarms using continuous energy replenishment," in *2018 9th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, 2018, pp. 478–484. DOI: 10.1109/UEMCON.2018.8796684.

[13] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006. DOI: 10.1109/TAC.2005.864190.

[14] A. Brandstätter, S. A. Smolka, S. D. Stoller, A. Tiwari, and R. Grosu, *Multi-agent spatial predictive control with application to drone flocking (extended version)*, 2022. DOI: 10.48550/ARXIV.2203.16960. [Online]. Available: https://arxiv.org/abs/2203.16960.

[15] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '87, New York, NY, USA: Association for Computing Machinery, 1987, pp. 25–34, ISBN: 0897912276. DOI: 10.1145/37401.37406.

[16] G. Silano, E. Aucone, and L. Iannelli, "CrazyS: A software-in-the-loop platform for the Crazyflie 2.0 nano-quadcopter," in *2018 26th Mediterranean Conference on Control and Automation (MED)*, 2018, pp. 1–6. DOI: 10.1109/MED.2018.8442759.

[17] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, 2149–2154 vol.3. DOI: 10.1109/IROS.2004.1389727.

[18] Stanford Artificial Intelligence Laboratory et al., *Robotic operating system*, version ROS Melodic Morenia, May 23, 2018. [Online]. Available: https://www.ros.org.

[19] J. Lu, H. Shen, L. Pan, X. Zhang, B. Tian, and Q. Zong, "Autonomous flight for multi-UAV in GPS-denied environment," in *Proceedings of 2021 5th Chinese Conference on Swarm Intelligence and Cooperative Control*, Z. Ren, M. Wang, and Y. Hua, Eds., vol. 934, Series Title: Lecture Notes in Electrical Engineering, Singapore: Springer Nature Singapore, 2023, pp. 892–901. DOI: 10.1007/978-981-19-3998-3_85. [Online]. Available: https://link.springer.com/10.1007/978-981-19-3998-3_85 (visited on 10/03/2022).

[20] M. Saska, T. Baca, J. Thomas, *et al.*, "System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization," *Autonomous Robots*, vol. 41, no. 4, pp. 919–944, Apr. 2017, ISSN: 0929-5593, 1573-7527. DOI: 10.1007/s10514-016-9567-z. [Online]. Available: http://link.springer.com/10.1007/s10514-016-9567-z (visited on 10/03/2022).

[21] K. Mohta, M. Watterson, Y. Mulgaonkar, *et al.*, "Fast, autonomous flight in GPS-denied and cluttered environments," *Journal of Field Robotics*, vol. 35, no. 1, pp. 101–120, Jan. 2018, ISSN: 15564959. DOI: 10.1002/rob.21774. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/rob.21774 (visited on 10/03/2022).

[22] D. O. Wheeler, D. P. Koch, J. S. Jackson, *et al.*, "Relative navigation of autonomous GPS-degraded micro air vehicles," *Autonomous Robots*, vol. 44, no. 5, pp. 811–830, May 2020, ISSN: 0929-5593, 1573-7527. DOI: 10.1007/s10514-019-09899-4. [Online]. Available: http://link.springer.com/10.1007/s10514-019-09899-4 (visited on 10/04/2022).

[23] K.-K. Oh and H.-S. Ahn, "Distance-based undirected formations of single-integrator and double-integrator modeled agents in n -dimensional space: DISTANCE-BASED UNDIRECTED FORMATIONS," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 12, pp. 1809–1820, Aug. 2014, ISSN: 10498923. DOI: 10.1002/rnc.2967. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/rnc.2967 (visited on 11/29/2022).

[24] M. Aranda, G. Lopez-Nicolas, C. Sagues, and M. M. Zavlanos, "Distributed formation stabilization using relative position measurements in local coordinates," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3925–3935, Dec. 2016, ISSN: 0018-9286, 1558-2523. DOI: 10.1109/TAC.2016.2527719. [Online]. Available: http://ieeexplore.ieee.org/document/7403871/ (visited on 10/05/2022).

[25] S.-M. Kang, M.-C. Park, and H.-S. Ahn, "Distance-based cycle-free persistent formation: Global convergence and experimental test with a group of quadcopters," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 380–389, Jan. 2017, ISSN: 0278-0046, 1557-9948. DOI: 10.1109/TIE.2016.2606585. [Online]. Available: http://ieeexplore.ieee.org/document/7565512/ (visited on 10/04/2022).

[26] X. Lan, W. Xu, and Y.-S. Wei, "Adaptive 3d distance-based formation control of multiagent systems with unknown leader velocity and coplanar initial positions," *Complexity*, vol. 2018, pp. 1–9, Sep. 6, 2018, ISSN: 1076-2787, 1099-0526. DOI: 10.1155/2018/1814653. [Online]. Available: https://www.hindawi.com/journals/complexity/2018/1814653/ (visited on 10/04/2022).

[27] B. Jiang, M. Deghat, and B. D. O. Anderson, "Simultaneous velocity and position estimation via distance-only measurements with application to multi-agent system control," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 869–875, Feb. 2017, Circular motion! ISSN: 0018-9286, 1558-2523. DOI: 10.1109/TAC.2016.2558040. arXiv: 1411.3841[cs]. [Online]. Available: http://arxiv.org/abs/1411.3841 (visited on 10/04/2022).

[28] F. He, Y. Wang, Y. Yao, L. Wang, and W. Chen, "Distributed formation control of mobile autonomous agents using relative position measurements," *IET Control Theory & Applications*, vol. 7, no. 11, pp. 1540–1552, Jul. 2013, ISSN: 1751-8652, 1751-8652. DOI: 10.1049/iet-cta.2012.1034. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1049/iet-cta.2012.1034 (visited on 11/29/2022).

[29] B. D. O. Anderson and C. Yu, "Range-only sensing for formation shape control and easy sensor network localization," in *2011 Chinese Control and Decision Conference (CCDC)*, Mianyang, China: IEEE, May 2011, pp. 3310–3315, ISBN: 978-1-4244-8737-0. DOI: 10.1109/CCDC.2011.5968829. [Online]. Available: http://ieeexplore.ieee.org/document/5968829/ (visited on 11/30/2022).

[30] R. Sharma and C. Taylor, "Cooperative navigation of MAVs in GPS denied areas," in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul: IEEE, Aug. 2008, pp. 481–486, ISBN: 978-1-4244-2143-5. DOI: 10.1109/MFI.2008.4648041. [Online]. Available: http://ieeexplore.ieee.org/document/4648041/ (visited on 10/04/2022).

[31] Y. Zou, C. Wen, and M. Guan, "Distributed adaptive control for distance-based formation and flocking control of multi-agent systems," *IET Control Theory & Applications*, vol. 13, no. 6, pp. 878–885, Apr. 2019, ISSN: 1751-8652, 1751-8652. DOI: 10.1049/iet-cta.2018.6001. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1049/iet-cta.2018.6001 (visited on 11/23/2022).

[32] K. Guo, X. Li, and L. Xie, "Ultra-wideband and odometry-based cooperative relative localization with application to multi-UAV formation control," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2590–2603, Jun. 2020, ISSN: 2168-2267, 2168-2275. DOI: 10.1109/TCYB.2019.2905570. [Online]. Available: https://ieeexplore.ieee.org/document/8680745/ (visited on 10/04/2022).

[33] B. Jiang, B. D. O. Anderson, and H. Hmam, "3-d relative localization of mobile systems using distance-only measurements via semidefinite optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 3, pp. 1903–1916, Jun. 2020, Highly relevant, ISSN: 0018-9251, 1557-9603, 2371-9877. DOI: 10.1109/TAES.2019.2935926. [Online]. Available: https://ieeexplore.ieee.org/document/8805133/ (visited on 10/04/2022).

[34] Y. Cai and Y. Shen, "An integrated localization and control framework for multi-agent formation," *IEEE Transactions on Signal Processing*, vol. 67, no. 7, pp. 1941–1956, Apr. 2019, ISSN: 1053-587X, 1941-0476. DOI: 10.1109/TSP.2019.2897968. [Online]. Available: https://ieeexplore.ieee.org/document/8636259/ (visited on 10/05/2022).

[35] M. Cao, C. Yu, and B. D. O. Anderson, "Coordination with the leader in a robotic team without active communication," in *2009 17th Mediterranean Conference on Control and Automation*, Thessaloniki, Greece: IEEE, Jun. 2009, pp. 252–257, ISBN: 978-1-4244-4684-1. DOI: 10.1109/MED.2009.5164548. [Online]. Available: http://ieeexplore.ieee.org/document/5164548/ (visited on 11/30/2022).

[36] S.-M. Kang, M.-C. Park, B.-H. Lee, and H.-S. Ahn, "Distance-based formation control with a single moving leader," in *2014 American Control Conference*, Portland, OR, USA: IEEE, Jun. 2014, pp. 305–310. DOI: 10.1109/ACC.2014.6858587. [Online]. Available: http://ieeexplore.ieee.org/document/6858587/ (visited on 10/05/2022).

[37] M. Cao, C. Yu, and B. D. Anderson, "Formation control using range-only measurements," *Automatica*, vol. 47, no. 4, pp. 776–781, Apr. 2011, ISSN: 00051098. DOI: 10.1016/j.automatica.2011.01.067. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0005109811000823 (visited on 10/05/2022).

[38] R. Suttner and Z. Sun, "Formation shape control based on distance measurements using lie bracket approximations," *SIAM Journal on Control and Optimization*, vol. 56, no. 6, pp. 4405–4433, Jan. 2018, ISSN: 0363-

0129, 1095-7138. DOI: 10.1137/18M117131X. [Online]. Available: https://epubs.siam.org/doi/10.1137/18M117131X (visited on 10/05/2022).

[39] J. Eversham and V. F. Ruiz, "Parameter analysis of Reynolds flocking model," in *2010 IEEE 9th International Conference on Cyberntic Intelligent Systems*, 2010, pp. 1–7. DOI: 10.1109/UKRICIS.2010.5898089.

[40] U. Mehmood, N. Paoletti, D. Phan, *et al.*, "Declarative vs rule-based control for flocking dynamics," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, ser. SAC '18, Pau, France: Association for Computing Machinery, 2018, pp. 816–823, ISBN: 9781450351911. DOI: 10.1145/3167132.3167222.

[41] S. Martin, A. Girard, A. Fazeli, and A. Jadbabaie, "Multiagent flocking under general communication rule," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 2, pp. 155–166, 2014. DOI: 10.1109/TCNS.2014.2316994.

[42] E. Soria, F. Schiano, and D. Floreano, "The influence of limited visual sensing on the Reynolds flocking algorithm," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, 2019, pp. 138–145. DOI: 10.1109/IRC.2019.00028.

[43] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Stability of flocking motion," University of Pennsylvania, Tech. Rep., 2003. [Online]. Available: https://www.georgejpappas.org/papers/boids03.pdf.

[44] M. Schwager, "A gradient optimization approach to adaptive multi-robot control," Ph.D. dissertation, Massachusetts Institute of Technology, 2009. [Online]. Available: https://dspace.mit.edu/handle/1721.1/55256.

[45] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek, "Optimized flocking of autonomous drones in confined environments," *Science Robotics*, vol. 3, no. 20, 2018. DOI: 10.1126/scirobotics.aat3536.

[46] E. Soria, F. Schiano, and D. Floreano, "Predictive control of aerial swarms in cluttered environments," in *Nature Machine Intelligence*, 2021. DOI: 10.1038/s42256-021-00341-y.

[47] M. Ballerini, N. Cabibbo, R. Candelier, *et al.*, "Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1232–1237, 2008. DOI: 10.1073/pnas.0711437105. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.0711437105. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.0711437105.

[48] W. Giernacki, M. Skwierczyński, W. Witwicki, P. Wroński, and P. Kozierski, "Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering," in *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2017, pp. 37–42. DOI: 10.1109/MMAR.2017.8046794.