2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

3D Robotic Swarmalators that Reconfigure, Navigate, and Avoid Obstacles

Zehui Xu[1*], Xinyue Xu[1*], and Steven Ceron[1]

***Abstract*— We realize 3D robotic swarmalators that reconfigure, navigate, and avoid obstacles with formal safety on Crazyflie drones. We incorporate ellipsoidal Control Barrier Functions to avoid downwash turbulence between drones, and a combination of Control Lyapunov Function and Control Barrier Function methods to enable the collective to move toward desired locations while avoiding collisions between drones or with nearby obstacles. We implement a global control scheme that moves the collective as a single entity, and a local control scheme that enables fluid-like flow around nearby obstacles while maintaining the same general collective formation. Finally, we demonstrate how the swarmalator model combined with these control schemes can be used to reconfigure and rotate a drone collective so it moves through a narrow passage without colliding with the surrounding environment. Our simulations and physical experiments quantify scalability limits and validate the feasibility of implementing 3D swarmalator-based control on real drone collectives.**

I. Introduction

Natural collective systems frequently exhibit two types of self-organization that in many cases can emerge separately from each other. The first form of self-organization is intimately related to swarming behaviors, where agents spatially self-organize by coupling to each other through spatial variables like cohesion and alignment; some of the most popular natural systems displaying these types of behaviors includes schools of fish, flocking birds, and packs of wolves [1]. The second form is temporal self-organization, and this is intimately related to the field of synchronization, where agents with internal temporally oscillating states, or phases, couple to each other to produce amazing patterns, some of the most basic being traveling waves and synchrony. This form of self-organization can be observed at all length scales in diverse systems including clapping humans, flashing fireflies, and chirping crickets [2].

In 2017, the swarming oscillator, swarmalator, model was introduced [3], and it enabled researchers to study some of the emergent collective behaviors that result from the interdependent coupling between the two forms of self-organization. This model consists of agents that coordinate through both their spatial positions and their internal phases; they move as a function of how they couple to other agents' phases, and they couple to each other's phases as a function of their positions and motions. Thus, the model combines the principles of spatial and temporal self-organization. Throughout the past few years, this model has been studied and diverse emergent collective behaviors have been found by researchers ranging from a variety of disciplines. These studies have focused on learning what happens when agents exhibit local coupling (can sense other agents up to a certain distance) [4], when external fields are introduced that affect the agents' pairwise coupling [5], and many other variations of the model [6]–[8]. Roboticists have also begun to study the model, and have realized diverse swarmalator behaviors in 2D through mobile terrain robots [9], [10]. Drones were also used to realize the swarmalator behaviors; however, although the drones were capable of flying in 3D, each of the behaviors realized remained in a 2D formation [10]. Other work has also explored the use of optimal control methods to guide simulated swarmalators' motions [11], and so that the spatial and temporal coupling between agents can be optimized so the collective remains in an annulus formation of desired proportions [12]. The diverse behaviors realized throughout these studies, almost all in simulation, suggest that the model can be used as a way of controlling robot collectives so they can assume diverse formations in 2D and 3D while exploiting a minimalistic coupling model. Our work studies how the 3D swarmalator behaviors can be realized in a physical robotic system, and how optimal control methods can be exploited for controllable reconfiguration, navigation, and obstacle avoidance.

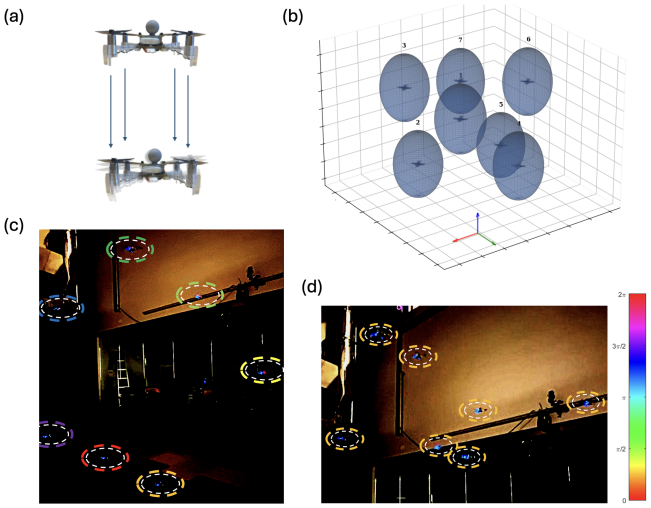

Fig. 1: (a) Depiction of Crazyflies used in our experiments and the downwash effect when one drone hovers above another. (b) Depiction of ellipsoidal barrier functions applied to prevent unsafe distances between drones. (c-d) Snapshots from physical experiments. (c) 3D static phase wave formation when $K = 0$ and $J = 1$. (d) 3D cluster forms when the coupling parameters are set to $K = 1$ and $J = 1$. Colored circles are added in to show each drone's phase; the colorbar corresponds to the phase values between 0 and 2π.

In our work, we exploit control barrier functions (CBFs),

*These authors contributed equally to this work.

[1]Robotics Department, University of Michigan, Ann Arbor, MI 48109, USA. sunnyx@umich.edu, xxinyue@umich.edu, sceron@umich.edu

which have emerged as a powerful method for guaranteeing obstacle avoidance and formal safety in robotics [13]. However, applying CBFs directly to multi-agent systems often incurs high computational cost, making large-scale deployment difficult [14]. By combining CBFs with the swarmalator model, we obtain a framework that is both computationally efficient and scalable. The swarmalator model enables us to control a group of drones to assemble in specific formations through global coupling parameters, while CBFs enforce safety around drones and obstacles.

A key challenge in aerial collectives is the downwash effect [15]; the turbulent airflow generated by one drone destabilizes others flying directly below it, as depicted in Figure 1a. To address this, we model each drone's safety region as an ellipsoid that is stretched in the vertical direction, as illustrated in Figure 1b. This design reflects the stronger aerodynamic interaction in the z direction while still allowing compact horizontal spacing. Our experimental platform consists of Crazyflies [16] (Figure 1a), which provide a lightweight and flexible testbed for validating this downwash-aware barrier formulation.

In addition to modeling inter-drone safety, we integrate CBFs and Control Lyapunov Functions (CLFs) [17] to guide the collective around obstacles. The CBFs enforce collision avoidance with obstacles [18] and maintain safe ellipsoidal spacing to eliminate downwash effects, while the CLF drives the collective toward target locations [19]. We investigate both global control, where the collective is treated as a single controllable unit, and local control, where each agent computes its own input based on nearby information. Together, the CBF-CLF framework maps state constraints into control constraints, which ensures that safety and navigation requirements are met simultaneously.

Some of the main features of our work are as follows: (1) we extend the swarmalator model from 2D to 3D so it can be realized by groups of Crazyflies, (2) we exploit ellipsoidal barrier functions to prevent drones from entering each other's downwash zones, (3) we combine the swarmalator model with a CBF-CLF controller to enable safe navigation around obstacles while maintaining collective formations in 3D, and (4) we enable automatic rotation of swarmalator behaviors in 3D so the group can safely pass through hollow obstacles.

II. The Model and Experimental Setup

In this section we discuss the swarmalator model and some of the challenges that need to be overcome to enable the realization of the swarmalator behaviors in 3D. We discuss our experimental setup in detail and hope that this paper enables other researchers to experiment with swarmalator behaviors using a similar setup.

A. Swarmalator Model

Agents are defined as swarmalators in the 3D model, where the velocity $\mathbf{v}_i^{\text{swm}}$ of each drone is being updated at each timestep by the equation below.

$$\mathbf{v}_i^{\text{swm}} = \frac{1}{N}\sum_{j\neq i}^{N}\left[\frac{\vec{r}_{ij}}{\|\vec{r}_{ij}\|}\left(A + J\cos(\theta_j-\theta_i)\right) - B\frac{\vec{r}_{ij}}{\|\vec{r}_{ij}\| d_{\min}(\vec{r}_{ij})}\right]$$

$$d_{\min}(\vec{r}_{ij}) = \begin{cases}\|\vec{r}_{ij}\|, & \text{if } \vec{r}_i > d_{\text{allow}}\\ \gamma d_{\text{allow}}, & \text{if } \vec{r}_i \leq d_{\text{allow}}\end{cases} \quad (1)$$

The attractive and repulsive forces are scaled by multiplicative factors to enable physically realizable collective behaviors. The attractive force is the unit vector of the positional displacement $\vec{r}_{ij} \in \mathbb{R}^3$ between the ith and jth drone, scaled by the factors A, J, and the phase difference $(\theta_j - \theta_i)$ between a pair of agents j and i.

The variable A is used to control the baseline strength of the attractive force between all agents, and is kept at one throughout this work. The coefficient J is used to adjust how much agents attract to each other as a function of their phase difference, and it is kept in the range $[-1, 1]$ throughout this work. $J = 0$ means there is no phase-dependent spatial attraction between agents, $J > 0$ enables agents to spatially attract to other agents if they have similar phases, and $J < 0$ enables repulsion between agents with similar phases. A distance-dependent repulsive term is governed by the coefficient B and is inversely proportional to the distance between agents. If the collective is unstable, increasing B could improve stability at the cost of having a larger collective radius. The function $d_{\min}$ is dependent on the distance between the ith and jth agent, which scales up the repulsive force to prevent collisions in certain conditions. When the distance is below the minimum allowable distance, d_{allow} is multiplied by the coefficient $\gamma < 1$ to amplify the repulsive force and quickly restore a safe distance between drones. The phase dynamics is governed by the following:

$$\dot{\theta}_i = \omega_i + \frac{K}{N}\sum_{j\neq i}^{N}\frac{\sin(\theta_j-\theta_i)}{\|r_{ij}\|} \quad (2)$$

Here, ω_i is the natural frequency of the ith agent, and the variable K is kept between $[-1, 1]$ to control how agents couple to each other's phases. When $K > 0$, agents' phases move toward synchronization, which enables clustering if $J > 0$; when $K = 0$ and $J > 0$, agents do not couple to each other's phases but instead exhibit a constant phase as they spatially rearrange to move toward other agents that have a similar internal phase value; and when $K < 0$, agents actively anti-couple to each other's phases. The change in phase is inversely scaled by the spatial distance between drones, which allows agents in close proximity to exert greater effects on each other.

B. Vehicle Dynamics

In simulations, swarmalators can be infinitesimally close to each other when arranging into a certain formation; however, this is impractical for physical experiments. Due to the physical dimensions of Crazyflies and turbulence they can induce on each other, sufficient spacing needs to be maintained between drones throughout the physical experiments. In particular, the downwash effect poses a challenge since it requires more spacing in the z direction than in the x and y directions to ensure aerial stability. Although the current swarmalator model enables separation between drones by applying the inverse of the $d_{\min}$ function in the repulsive

force, CBFs (Sec. III) are introduced to guarantee that the drones can safely fly outside the downwash 'danger' zones of other drones while still converging to a compact collective formation.

C. Experimental Setup

The experimental setup for the Crazyflie drone experiments is illustrated in Fig. 2. The open source pycrazyswarm Wrapper classes developed by USC-ACTLab [20] were used to control the drones throughout the experiments, providing low-level control abstraction. Due to the nature of wrapper class designs, only positional commands are offered as a stable alternative for control commands. As a result, the velocities calculated from the swarmalator model are modified to adapt to the change. A predefined timestamp Δt is multiplied with the velocities $\mathbf{v}_i$ to obtain the desired change in position $\Delta\vec{r}_i$ of each drone in the next timestep, which is then parameterized for translational motion commands. A motion capture system was used to provide fast and accurate tracking of all drones. The experimental setup includes 12 cameras in an enclosed space for implementation to mitigate any turbulence during flights. One marker is attached to each drone and the identification of individual drones are processed in the backend to keep track of individual drones. The positions are published through a ROS node to allow continuous access of the relative positions from the wrapper class; this enables the swarmalator model to use the real-time positions of the drones when calculating their motion for the next time step. Throughout the experiment, the positions of all the drones are recorded at each timestep and stored in a file. The physical experiment visualization figures in the paper are created from this data.

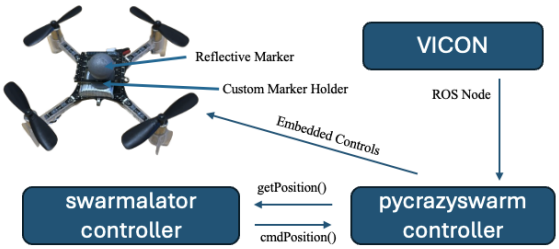

Fig. 2: Overview of Crazyfly system for demonstrating controllable 3D robotic swarmalator behaviors.

To demonstrate the capabilities of the physical setup and a preliminary performance of the swarmalator model in a closed-loop control environment, Fig. 3a shows the experiment-simulation comparison between the 2D versions of three swarmalator behaviors. In these experiments, each Crazyflie hovers at the same height and rearranges itself along that plane to achieve the desired configuration. As shown in Fig. 3a, the behaviors of the swarmalators within a closed-loop control setting is similar to that of the theoretical one. When $K = 1$ and $J = 1$, both simulation and visualization show that drones form a circular shape with synchronized phases, demonstrating the static sync behavior. The static phase wave behavior is shown when $K = 0$ and $J = 1$, where the drones form a circle outline with similar phases closest to each other. When $K = -1$ and $J = 1$, the active async state is formed, and it is demonstrated by the drones moving around as their phases continually anti-couple with each other and change over time.

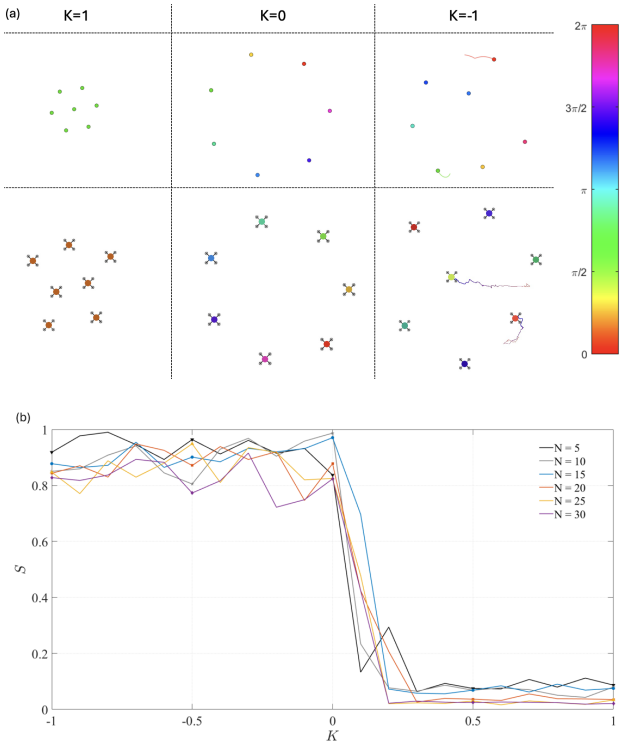

Fig. 3: (a) Simulation and visualization of the 2D swarmalator model with seven drones. The top row shows simulation results and the bottom row shows visualization of physical experiments. From left to right, the formations correspond to $J = 1, K = 1$ (static sync), $J = 1, K = 0$ (static phase wave), and $J = 1, K = -1$ (active async). In the active async case, the trajectories of selected drones are highlighted with lines. The colorbar on the right indicates the phase of each agent, ranging from 0 to 2π. (b) Space-phase order parameter S versus coupling parameter K for 3D collectives in simulation with $J = 1$ and number of drones $N = \{5, 10, 15, 20, 25, 30\}$. Values $S \approx 1$ indicate a phase wave, whereas $S \approx 0$ indicate a static sync state.

Throughout past studies, the space-phase order parameter, $S = \frac{1}{N} \max_{\pm} \left| \sum_{j=1}^{N} e^{i(\theta_j \pm \phi_j)} \right|$, has been used to quantify a collective's level of circumferential organization by phase. Here, ϕ_j is the angular position of agent j about the collective centroid. $S \approx 1$ means the phases are well organized around the centroid, while $S \approx 0$ suggests minimal spatial organization of the agents' phases. Collectives that exhibit formations similar to the static sync and active async behaviors in Fig. 3b have a low S order, while collectives that create formations similar to a static phase wave have a high S order. This plot was created from simulations of our drone system and it argues that drone collectives should still exhibit similar trends in spatial organization by phase

(quantified by the order parameter S) as a function of K even as the number of drones is increased from 5 to 30.

III. Control Method & Experimental Results

The swarmalator model provides a nominal translational velocity $\mathbf{v}_i^{\text{swm}} = f_i(x, \theta)$ that depends on the pairwise interactions. We add a control input that acts on translation and write the kinematics in control affine form:

$$\mathbf{v}_i = f_i(x, \theta) + g_i\, u_i \tag{3}$$

Here, $u_i \in \mathbb{R}^3$ and $g_i = I_3$. Our control design has three main goals: (1) prevent drones from entering each other's downwash regions, (2) ensure they can avoid static obstacles in the environment, and (3) guide the collective to the desired target location. To achieve these goals, we combine CBFs with CLFs. CBFs ensure safety by enforcing separation between drones and obstacles, while the CLF drives the collective to follow a target trajectory. As an additional enhancement, Rodrigues' rotation formula [21] is integrated to control the rotation of a collective formation, so the group can navigate through a channel-like passage.

We study two different control schemes. In the global control setting, the same control input is applied to all drones, such that the collective is treated as a single coordinated body. In the local control setting, each agent computes its own input based on local information, which allows for decentralized responses while still maintaining the formation.

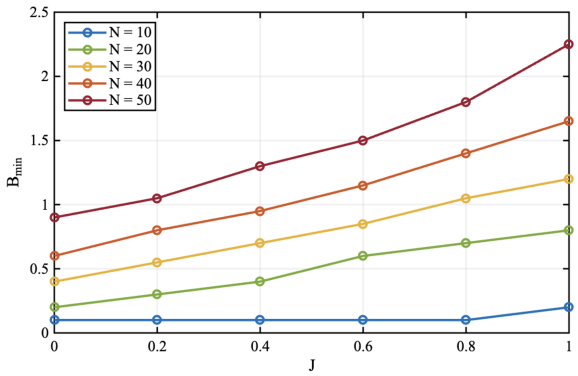

Fig. 4: Minimum B values required such that given the N and J constraints, the QP solver does not fail throughout the 3D formation experiment.

During the simulation phase, we observed that in certain scenarios, particularly when large adjustments were needed to rearrange drones into the collective body, the Quadratic Program (QP) solver could not always find feasible solutions under the existing constraints (the detailed formulation of the QP will be presented in the following section). This became especially prominent when there was high spatial attraction and a large number of agents in the collective. To analyze the effect, we measured the minimum repulsion parameter B in the simulation that required the solver to succeed if there was a certain level of phase-dependent spatial coupling given by the parameter J (Fig. 4). As the collective size increased, the available space per drone decreased, demanding a larger repulsion force to maintain feasible motion. Similarly, higher J values led to greater mobility, which also required stronger repulsion to preserve formation stability. These simulations provide us guidance on how to choose the repulsion parameter B when testing with increasing numbers of agents.

A. Control Barrier Function

As previously mentioned, due to the downwash effect of drones, a spherical repulsion zone cannot achieve a compact collective formation while maintaining aerial stability. We use an ellipsoidal safety zone that reserves more space vertically and less space horizontally. CBFs give a formal way to keep the state inside as a safe set at all times.

To describe safety in a collective, we first look at the full configuration space, denoted $\mathcal{X} \subset \mathbb{R}^{3N}$, which contains the positions of all N agents. Within this space we specify a safe region $\mathcal{S}$, which is the set of all collective states that respect collision-free spacing and obstacle avoidance. The safe region is defined by the barrier function $b : \mathcal{X} \to \mathbb{R}$ which is smooth and easy to evaluate. Each barrier function acts like a virtual wall: when $b(\mathbf{x}) \geq 0$ the current state $\mathbf{x}$ is inside the safety region, and when $b(\mathbf{x}) < 0$ the collective is in violation of safety. Collecting all such conditions gives $\mathcal{S} := \{\mathbf{x} \in \mathcal{X} : b(\mathbf{x}) \geq 0\}$.

We can use different 3D shapes to represent barriers, and in the following experiments, two types of obstacles are modeled, stacked ellipsoids to approximate box obstacles and a torus to represent a hoop obstacle. Each drone is also modeled as an ellipsoid, and is defined as:

$$\left(\frac{x_i - x_{\text{obs}}}{a}\right)^2 + \left(\frac{y_i - y_{\text{obs}}}{b}\right)^2 + \left(\frac{z_i - z_{\text{obs}}}{c}\right)^2 = 1 \tag{4}$$

Here, a and b denote the horizontal radii, chosen as 4.5 times the drone radius, while c denotes the vertical radius, chosen as 8.5 times the drone radius to account for the downwash effect. This scaling ensures the asymmetric spacing required for safe aerial flight.

Then, we define a barrier function for each pair of drones i and j:

$$b_{ij}(\mathbf{x}) = \left(\frac{x_i - x_j}{a}\right)^2 + \left(\frac{y_i - y_j}{b}\right)^2 + \left(\frac{z_i - z_j}{c}\right)^2 - 1 \tag{5}$$

To keep the state inside the safety set $\mathcal{S}$, we enforce the barrier condition on each constraint. For a control affine system (Eq. 3) the condition is

$$\sup_{u \in \mathcal{U}} \left[L_f b(\mathbf{x}) + L_g b(\mathbf{x}) u\right] + \alpha(b(\mathbf{x})) \geq 0 \tag{6}$$

Here, $\alpha : \mathbb{R} \to \mathbb{R}$ is a Lipschitz continuous extended class $\mathcal{K}$ function. The Lie derivatives of $b(\mathbf{x})$ are given by $L_f b(\mathbf{x}) = \frac{\partial b}{\partial \mathbf{x}} f(\mathbf{x})$ and $L_g b(\mathbf{x}) = \frac{\partial b}{\partial \mathbf{x}} g(\mathbf{x})$. In the inequality, Eq. 6, the function $\alpha(\cdot)$ limits how fast $b(\mathbf{x})$ can decrease near the boundary $b(\mathbf{x}) = 0$, so any admissible input u that satisfies the condition keeps the trajectories inside $\mathcal{S}$ and makes the safe set forward invariant. Based on these barrier conditions, we created a QP that determines the control input to ensure the inter-drones' safety, the resulting optimization

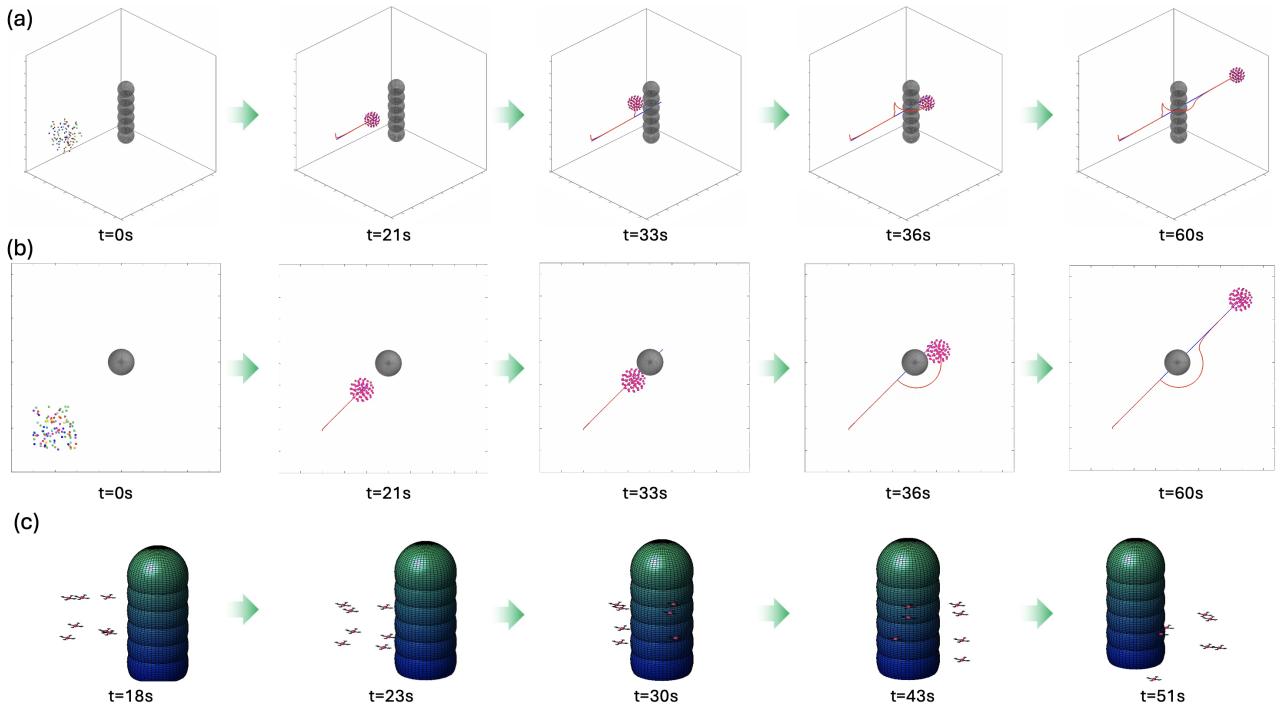

Fig. 5: Global control in simulation and physical experiment validation. (a-b) Sequential snapshots of a simulation of 100 drones converging into a 3D sync cluster and moving around a pillar toward a desired target location. The blue line shows the desired trajectory defined in the CLF while the red line shows the real trajectory of the whole collective. (a) Isometric view, (b) top view. (c) Snapshots of the experimental visualization of drones flying as a cluster around an obstacle using the global control method (supplemental material includes experiment video).

problem minimizes the deviation from the nominal input while enforcing all safety conditions:

$$\begin{aligned} &\min_{u_i \in \mathcal{U}} \quad \|u_i\|^2 \\ &L_f b_{ij}(\mathbf{x}) + L_g b_{ij}(\mathbf{x})\, u_i + \alpha\big(b_{ij}(\mathbf{x})\big) \;\geq\; 0, \quad \forall\, i \neq j \end{aligned} \tag{7}$$

B. Control Lyapunov Function

To guide the collective along a desired trajectory while preserving its formation, we integrate a CLF into the swarmalator model. The CLF introduces an artificial potential field that pulls the group's centroid from its current position toward the target. The collective's centroid is defined as $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$, where $\mathbf{x}_i$ is the position of agent i. Let $\mathbf{p}_{\text{target}} \in \mathbb{R}^3$ denote the desired target location. We choose the Lyapunov function $V(\mathbf{x}) = \|\bar{\mathbf{x}} - \mathbf{p}_{\text{target}}\|^2$, which measures the squared distance between the collective centroid and the target and thus serves as a measure of collective tracking error. We apply the CLF condition to ensure exponential convergence of the centroid to the target:

$$\inf_{u \in \mathcal{U}} \left[L_f V(\mathbf{x}) + L_g V(\mathbf{x}) u + \lambda V(\mathbf{x}) \right] \leq 0 \tag{8}$$

Here, $L_f V(\mathbf{x}) = \frac{\partial V}{\partial \mathbf{x}} f(\mathbf{x})$ and $L_g V(\mathbf{x}) = \frac{\partial V}{\partial \mathbf{x}} g(\mathbf{x})$ are the Lie derivatives of $V(\mathbf{x})$, $\frac{\partial V}{\partial \mathbf{x}} = 2(\bar{\mathbf{x}} - \mathbf{p}_{\text{target}})^\top$ is its gradient with respect to the collective state, and $\lambda > 0$ sets the rate of exponential convergence. This ensures that the chosen input u reduces the tracking error and drives the collective centroid toward the target while maintaining stability.

C. Global & Local Control Schemes

1) Global Control: In the global control scheme, the collective is treated as a single entity, acting as an extra control layer beyond the inter-drone's controls. A common control input $u \in \mathbb{R}^3$ is applied to all drones, which preserves the relative formation of the group. The collective's centroid is computed as the average position of all agents, and the collective radius is defined as the maximum distance between the centroid and any individual drone. This formulation allows the group to move as a coherent body while maintaining its shape.

In the global control scheme, the obstacle is modeled as five stacked ellipsoids, which together approximate the 3D volume of a block. This setup allows us to evaluate whether the global controller can preserve the formation and ensure collision-free navigation in a cluttered environment. With M obstacles in the environment, the control input is obtained by solving a CLF-CBF QP:

$$\begin{aligned} &\min_{u \in \mathcal{U}, \delta \geq 0} \quad \|u\|^2 + \rho\delta^2 \\ &\text{s.t.} \quad L_f V + L_g V u + \lambda V \leq \delta \\ &\qquad L_f b_m + L_g b_m u + \alpha(b_m) \geq 0,\ \forall m \in \{1, ..., M\} \end{aligned} \tag{9}$$

Here δ is a nonnegative slack variable that allows the CLF constraint to be relaxed when it cannot be satisfied exactly. A penalty term $\rho\delta^2$ in the cost makes sure δ stays small, so the constraint is only relaxed when necessary.

Fig. 5 shows the global control scheme experiments in

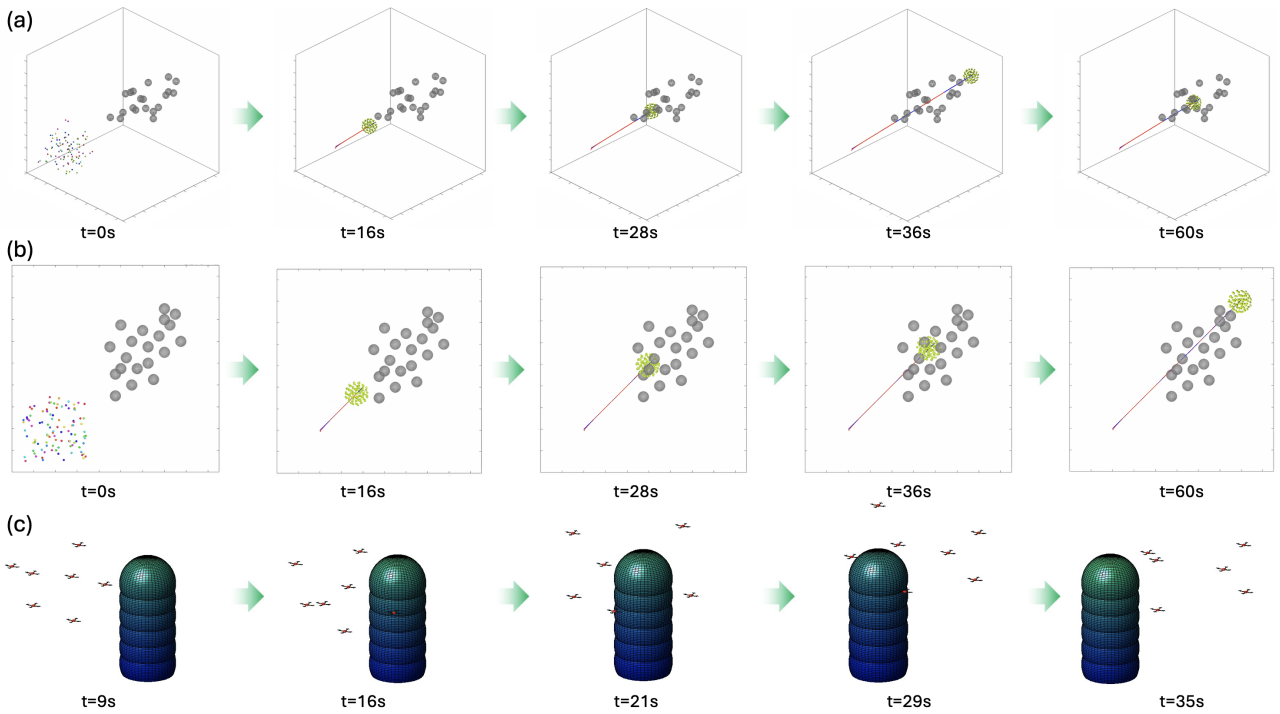

Fig. 6: Local control in simulation and experimental validation. (a-b) Sequential snapshots of a simulation of 100 drones converging into a 3D sync cluster and moving around a series of obstacles toward a desired target location. The blue line shows the desired trajectory defined in the CLF while the red line shows the real trajectory of the whole collective. (a) Isometric view, (b) top view. (c) Snapshots of the experimental visualization of drones flying as a cluster around a series of obstacles using the local control method (supplemental material includes corresponding experiment video).

simulation (Fig. 5a-b) and the physical implementation on drones (Fig. 5c). The agents were initialized with $K = 1$ and $J = 1$ so they would form an aggregated cluster. A single input from the CLF-CBF QP was applied to all agents, with the stacked ellipsoids used as the obstacle. The CLF specified a target beyond the obstacle and the CBF enforced clearance. In both simulation and physical experiment, the cluster preserved its shape, adjusted its course as a single body, and moved around the obstacle before rejoining the reference path. The centroid trajectory deviated in a smooth manner to avoid the obstacle and then steered back toward the target, with all drones maintaining collision-free motion.

2) Local Control: Compared with global control, the local control scheme is decentralized and assigns each drone its own control input. This allows every agent to interact with obstacles individually rather than following a single command for the entire collective. When no obstacles are nearby, the drones simply follow the swarmalator dynamics and maintain their collective formation. When obstacles are present, each drone adjusts its trajectory to avoid collisions, resulting in fluid-like motion as the group flows around the environment.

In the local control method, the CBFs that prevent drones from entering each other's downwash regions are applied individually to every agent. These inter-drone safety constraints can be incorporated into the QP together with the obstacle avoidance terms:

$$\begin{aligned} \min_{u_i \in \mathcal{U}_i, \delta_i \geq 0} \quad & \|u_i\|^2 + \rho \delta_i^2 \\ \text{s.t.} \quad & L_f V_i + L_g V_i u_i + \lambda V_i \leq \delta_i \\ & L_f b_{im} + L_g b_{im} u_i + \alpha_1(b_{im}) \geq 0, \ \forall m \in \{1, ..., M\} \\ & L_f b_{ij} + L_g b_{ij} u_i + \alpha_2(b_{ij}) \geq 0, \ \forall j \neq i \end{aligned} \tag{10}$$

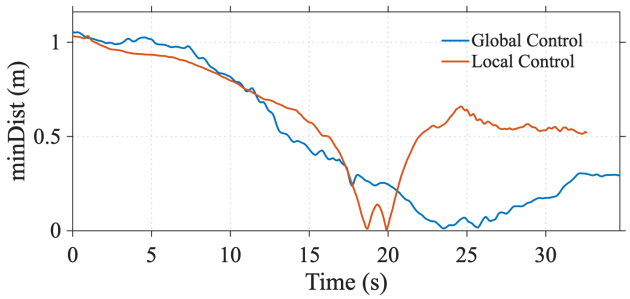

Fig. 7: Minimum distance between drones and the obstacle over time in the physical experiment. Both controllers maintained positive clearance throughout the run.

A set of experiments was conducted to demonstrate local control, as shown in Fig. 6. The drones were initialized with the same phase and formed a spherical shape at the start, with swarmalator parameters matching those used in the global control case. In the physical experiment (Fig. 6c), each drone adjusted its trajectory individually when approaching the obstacle, such that some passed on the left side, others went

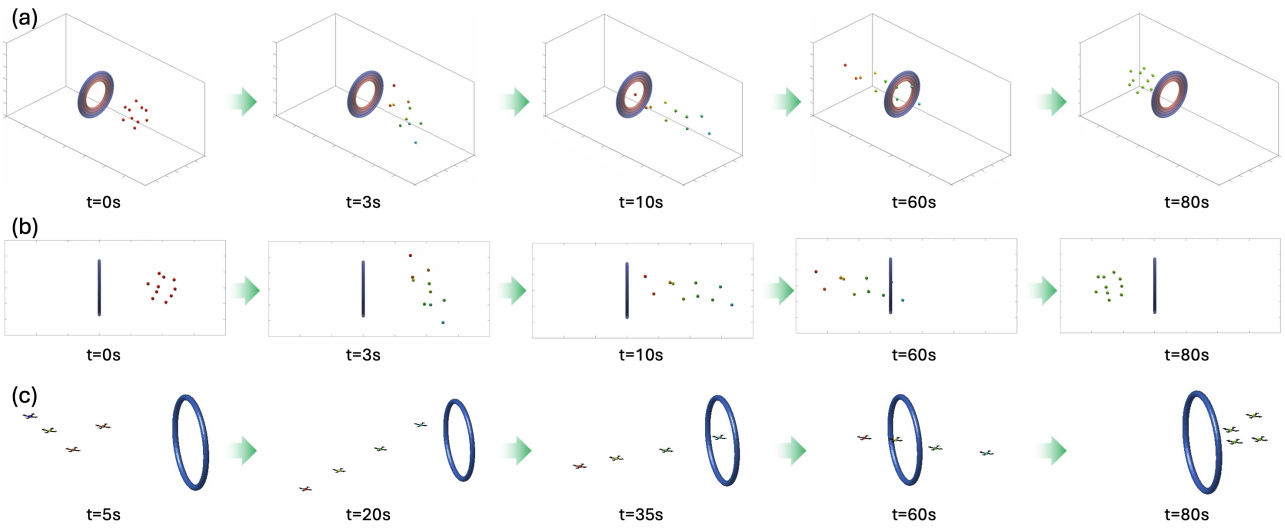

Fig. 8: Combined reconfiguration, navigation, and obstacle avoidance experiment in simulation and physical validation. (a-b) Sequential snapshots of a simulation of 10 drones converging into a 3D sync cluster, then reconfiguring into a ribbon, aligning with the direction of the hoop, and moving to the other side; (a) Isometric view, (b) top view. (c) Snapshots of the experimental visualization of the experiment (supplemental material includes corresponding experiment video).

around the right, and a few flew over the top. After leaving the obstacle, the group reformed into the original spherical shape. To assess scalability, we also performed a simulation with 100 agents navigating through an environment containing 20 obstacles (Fig. 6a-b). Each agent was governed by the integrated CLF-CBF framework, which guided the collective across the workspace while avoiding both obstacles and neighboring agents. The results show that the collective followed the desired collective path while avoiding collisions with both obstacles and neighboring agents. This fluid-like motion demonstrates the ability of the integrated CLF-CBF framework to guide drone collectives safely through complex environments. Despite frequent control inputs, the agents are still able to maintain their collective behavior all the time.

3) Control Performance Evaluation: In order to evaluate the performance of the control framework, we analyzed the minimum distance between the drones and the obstacle over time in the physical experiment on both global and local control. The trajectories of all drones were recorded using the motion capture system, and the minimum distance at each time step was computed as shown in Fig. 7. From the recorded trajectories, we obtained minimum clearances of $0.012\,\mathrm{m}$ for global control and $0.002\,\mathrm{m}$ for local control. In practice, the obstacle used in the physical experiments was set slightly smaller than the one defined for the control framework. This adjustment accounted for drone vibrations and the small errors inherent in the motion capture system, and ensured that the safety margins observed in the experiments remained strict.

D. Reconfiguration, Navigation, and Obstacle Avoidance

As a final demonstration of the usefulness of the swarmalator model in enabling a collective to reconfigure into formations that make it easier to pass through difficult-to-navigate environments, we show a proof-of-concept experiment in which the coupling parameter K is changed and the drones' phases are distributed so the collective can create a ribbon-like formation that allows it to pass through a narrow passage using the local control approach. Moreover, we show how reorientation of the emergent formation to enable the collective to safely move along the desired trajectory can be achieved by integrating Rodrigues' rotation formula ($\mathbf{r}^{+} = \mathbf{r}\cos\alpha + (\hat{\mathbf{n}}\times\mathbf{r})\sin\alpha + \hat{\mathbf{n}}\big(\hat{\mathbf{n}}\cdot\mathbf{r}\big)(1-\cos\alpha)$). We use it here to control the rotation of the collective so the group aligns its long axis perpendicular to the circular cross section of a hoop and thus passes safely through it. Applying the rotation formula to our specific use, $\mathbf{r}$ is the positional vector difference between the collective's centroid and the ith drone; $\hat{\mathbf{n}}$ represents the normalized target rotation axis in terms of yaw, pitch, and roll relative to the centroid; and $\alpha = \Omega\,\Delta t$ denotes the rotation angle for the step, computed from the commanded rotational speed Ω and the time step duration Δt. The pitch, yaw, and roll were calculated based on the vector formed by the two drones that are the furthest apart in the collective, and the standard x, y, z axes were used as a baseline to measure the yaw, pitch, and roll of the whole group. A PID controller was then applied to the yaw, pitch, and roll to calculate the desired rotational speed Ω of the group, which was then used to obtain α. The final rotation speed $\mathbf{v}_i^{\mathrm{rot}}$ was computed and added on top of the computed swarmalator velocity $\mathbf{v}_i^{\mathrm{swm}}$ to achieve the desired rotation while maintaining the collective formation.

We evaluated this rotation extension in both simulation and physical experiments, as shown in Figure 8. The drones started with $K = 1$ and $J = 1$, which produced a spherical formation at the beginning. To prepare the group to pass through the hoop, we reinitialized the drones' phases so they were uniformly distributed between 0 and π, and we set $K = 0$. This enabled the drones to cease any pairwise phase coupling as they spatially rearranged themselves by

phase similarity. Since the group only had phases between 0 and π, there was no attractive force pulling the group to form a loop, as would be the case if their phases were distributed between 0 and 2π, instead the collective organized itself by phase in the form of a ribbon-like shape. This allowed the drones to pass through the hoop to its other side; the whole time, we used the rotation formula to automatically adjust the ribbon's major axis yaw, pitch, and roll so it would remain perpendicular to the plane of the hoop's circular cross section. During the transit, a QP was the same as the local control one (Eq. 10), which combined the CLF tracking term with the CBF constraints so the inter-drone ellipsoids generated safe inputs.

Once all of the drones made it to the other side of the hoop, we changed the phase coupling parameter to $K = 1$. This enabled the drones to synchronize their phases, and since J was been kept equal to 1 throughout the whole process, the drones now moved toward a cluster formation. It is worth noting here that in our specific experimental setup, the collective could perhaps have also made it through and around the hoop if the collective remained as an aggregated cluster formation ($K = 1$, $J = 1$) and used the local control approach. However, the results here demonstrate the usefulness of the swarmalator model in enabling the collective to automatically switch to a formation in which it is more constricted along an axis, which would be useful in scenarios where the group needs to move through narrow passages and it is not possible to move through the exterior.

IV. Conclusion

Through this work, we studied the real-world implementation of the swarmalator model for 3D behaviors using Crazyflies, and have demonstrated in physical experiments and simulations how closed-loop control can be integrated to enable a swarmalator drone collective to navigate toward desired target locations while avoiding collisions between agents or with nearby obstacles. We also demonstrated how a collective formation could be made to rotate to align along a desired direction so the collective could reconfigure and navigate through a narrow passage created by a hoop. The control layers that we have added on to the swarmalator model have enabled us to physically realize swarmalator-driven behaviors through proof-of-concept experiments, and our simulations demonstrate how this model could be exploited in future scenarios where the environment may be even more complex or filled with many more obstacles using the same methods. Through future works, we plan to expand upon our existing experimental setup and control schemes to overcome some of the limitations of our current work. One of the physical limitations included the small workspace allowed by the current motion capture system, which led to a relatively low number of drones throughout all experiments and restricted the feasible number of objects we could include in our experiments. Moreover, all QPs were solved centrally on a single computer, which introduced computational delays and further limited scalability. Our future work will leverage drones with on-board computation and communication, and outdoor environments that will enable larger experiment arenas; these modifications will allow us to increase the scalability of the system, possibly reduce latency, and support larger robot collectives.

References

[1] H. Duan, M. Huo, and Y. Fan, "From animal collective behaviors to swarm robotic cooperation," *National Science Review*, vol. 10, no. 5, p. nwad040, 2023.

[2] E. Ott and T. M. Antonsen, "Frequency and phase synchronization in large groups: Low dimensional description of synchronized clapping, firefly flashing, and cricket chirping," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 5, 2017.

[3] K. P. O'Keeffe, H. Hong, and S. H. Strogatz, "Oscillators that sync and swarm," *Nature communications*, vol. 8, no. 1, pp. 1–13, 2017.

[4] H. K. Lee, K. Yeo, and H. Hong, "Collective steady-state patterns of swarmalators with finite-cutoff interaction distance," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 31, no. 3, 2021.

[5] J. U. Lizarraga and M. A. de Aguiar, "Synchronization and spatial patterns in forced swarmalators," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 5, 2020.

[6] K. O'Keeffe and H. Hong, "Swarmalators on a ring with distributed couplings," *Physical Review E*, vol. 105, no. 6, p. 064208, 2022.

[7] K. O'Keeffe and C. Bettstetter, "A review of swarmalators and their potential in bio-inspired computing," *Micro-and Nanotechnology Sensors, Systems, and Applications XI*, vol. 10982, pp. 383–394, 2019.

[8] A. Yadav, V. Chandrasekar, W. Zou, J. Kurths, and D. Senthilkumar, "Exotic swarming dynamics of high-dimensional swarmalators," *Physical Review E*, vol. 109, no. 4, p. 044212, 2024.

[9] R. Beattie, S. Ceron, and D. Rus, "Realizing emergent collective behaviors through robotic swarmalators," in *Accepted to 2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.

[10] A. Barciś and C. Bettstetter, "Sandsbots: Robots that sync and swarm," *IEEE Access*, vol. 8, pp. 218 752–218 764, 2020.

[11] X. Xu, W. Xiao, and S. Ceron, "Navigation of robotic swarmalators with dynamics and constraints," *IEEE Robotics and Automation Letters*, vol. 11, no. 2, pp. 1066–1073, 2025.

[12] S. Ceron, W. Xiao, and D. Rus, "Reciprocal and non-reciprocal swarmalators with programmable locomotion and formations for robot swarms," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 12 233–12 239.

[13] W. Xiao, C. G. Cassandras, C. A. Belta, and D. Rus, "Control barrier functions for systems with multiple control inputs," in *2022 American Control Conference (ACC)*, 2022, pp. 2221–2226.

[14] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *Ieee Access*, vol. 6, pp. 28 573–28 593, 2018.

[15] R. Morishita, S. Kawai, and H. Nobuhara, "Downwash reduction drone with adaptive rotors and its 3d aerodynamic analysis and stabilization control," *IEEE Access*, vol. 12, pp. 22 832–22 840, 2024.

[16] 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)*. IEEE, 2017, pp. 37–42.

[17] A. Anand, K. Seel, V. Gjærum, A. Håkansson, H. Robinson, and A. Saad, "Safe learning for control using control lyapunov functions and control barrier functions: A review," *Procedia Computer Science*, vol. 192, pp. 3987–3997, 2021.

[18] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*. Ieee, 2019, pp. 3420–3431.

[19] R. A. Freeman and J. A. Primbs, "Control lyapunov functions: New ideas from an old source," in *Proceedings of 35th IEEE conference on decision and control*, vol. 4. IEEE, 1996, pp. 3926–3931.

[20] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3299–3304.

[21] E. Piña, "Rotations with rodrigues' vector," *European journal of physics*, vol. 32, no. 5, p. 1171, 2011.