Distributed Formation Control of Double-Integrator Systems via Adaptive CBFs with Dynamic Safety Margins

Guettaf Wahid[1]*, Bouderah Brahim[1,2], Zaitri Mohamed Abdelaziz[3]

[1] Laboratory of Pure and Applied Mathematics "LMPA", University of M'Sila, University Pole, M'Sila 28000, Algeria
[2] Computer Science Department, Abdelhamid Ibn Badis Mostaganem University, Mostaganem 27000, Algeria
[3] Centre of Mathematics (CMAT), University of Minho, Campus de Gualtar, Braga 4710-057, Portugal

Corresponding Author Email: wahid.guettaf@univ-msila.dz

ABSTRACT

This paper presents the design of a safe formation control approach for homogeneous multi-robot systems with double-integrator dynamics—a model integrating inertial dynamics, which are essential for accurate control during high-speed manoeuvres. To ensure safe navigation, a real-time control scheme is proposed, in which formation constraints and collision avoidance are formulated as control barrier function (CBF) conditions within a quadratic program (QP). Forward invariance of the safe formation set is achieved by guaranteeing a minimum safety margin, preserving formation geometry, and forcing the system to stay in the safe set. Collision-free transitions between different shapes are achieved through CBF. Numerical simulations in cluttered, dynamic environments demonstrate that our approach preserves formation integrity and prevents collisions without sacrificing agility. These results demonstrate the advantages of combining elastic formation flexibility with safety guarantees, which constitutes a promising approach for agile and scalable coordination.

1. INTRODUCTION

Faced with the inability of a single robot to perform complex tasks, due to its limitations, failures and environmental difficulties, there is a strong motivation to coordinate with other robots to overcome the incapacity in the face of complex tasks, inspiring groups of birds or fish. A multi-robot system contains two or more robots; this type of system is used in many fields, such as surveillance, rescue, and exploration of dangerous or congested areas. The difficulty of controlling a multi-robot system has prompted researchers to develop theoretical and algorithmic avenues for increasing system stability and maintaining a high level of safety.

In contrast to single-integrator models that omit acceleration, double-integrator dynamics offer a more accurate depiction of robotic platforms, where control inputs relate to acceleration instead of velocity. The selection of this model is crucial in scenarios requiring rapid or agile movements, such as drone swarms, autonomous driving, or legged robotics, where inertial effects profoundly impact system behavior and must be explicitly incorporated into the control architecture. By integrating second-order dynamics, control techniques can more efficiently manage the motion and response of agents within physical limitations.

The main objective in general is to guide the robots while respecting the constraints linked to their capacities, the navigation space, and the task to be carried out. All this according to their detection capacities and the topology of communication between them. Different problems of

formation control have been treated in the literature: distributed consensus [1-4]; leader-follower [5, 6]; artificial potential fields [7, 8]; graph rigidity and distance-based shape control [9, 10]; distributed model predictive control [11, 12]; safety via control barrier functions (CBF) [13-16]; and learning-based methods [17-20].

Control Barrier Functions (CBFs) [13, 15] have become a cornerstone for enforcing hard safety constraints such as inter-robot collision avoidance because they translate physical rules into real-time inequalities that are satisfied by solving a Quadratic Program (QP) at every control step. Although a single, centralized CBF-QP guarantees safety, its computational cost escalates rapidly with the number of robots. Recent work therefore introduces distributed CBF formulations in which each robot solves a lightweight local QP that depends only on its own state and occasional neighbors information, yet still preserves global safety certificates through graph-theoretic guarantees [21].

The CBF framework has been expanded in recent years to accommodate increasingly complex systems and task requirements. To enforce safety constraints on systems with relative degrees greater than one, as is frequently the case in double-integrator or underactuated robotic dynamics, Higher-Order Control Barrier Functions (HOCBFs) [22-24] have been proposed. By creating a series of inequality constraints that guarantee higher-order derivatives of safety functions stay bounded, HOCBFs make safe set invariance possible. Applications like multi-vehicle lane keeping, quadrotor navigation, and manipulation tasks where safety involves

acceleration-level constraints have demonstrated the potential of these techniques. Additionally, recent advances have demonstrated that HOBFs expand the class of systems and constraints for which explicit, real-time safety enforcement is possible, accommodating nonuniform and state-dependent relative degrees and relaxing assumptions on system completeness and regularity [25]. Moreover, HOBFs admit robustness properties: the resulting safe sets can be rendered asymptotically stable, ensuring resilience to modelling perturbations and external disturbances. In performance-critical applications, designers can also define regions where nominal control policies are implemented, maximizing efficiency while preserving safety [25].

The local objective often focusses two priorities: (i) preserving closeness to a nominal control input, frequently produced by a formation-tracking proportional-derivative (PD) controller, and (ii) enforcing desired relative offsets that define or adjust the formation. The integration of consensus-based distributed optimization algorithms, particularly the consensus alternating direction method of multipliers (ADMM), enhances scalability and fault tolerance inside the CBF-QP framework. These algorithms partition the global coordination job into low-dimensional subproblems, necessitating only local variable exchanges with immediate neighbors. Convergence is assured under mild convexity conditions. The distributed CBF-QP and consensus optimization layers offer a systematic and computationally efficient approach to ensuring provably safe, real-time coordination in extensive multi-robot systems, independent of centralized control.

The paper is organized as follows: Section 2, presents the proposed methodology for distributed formation control in multi-robot systems with double-integrator dynamics. It covers the system modeling, the formulation of desired and deformable formations, and the application of CBFs and HOCBF, including their higher-order extensions, to ensure safety and stability. Section 3 provides simulation results and discusses the performance of the proposed approach in various scenarios.

We conclude in Section 4, by discussing the implications and limitations of our results, and summarizing the main findings and outlining potential directions for future research.

2. METHODOLOGY

2.1 Multi-robot system with double integrator dynamics

Single-integrator models, with their simplicity and tractability, insufficiently represent the inertial characteristics of physical systems, rendering them unsatisfactory for applications requiring rapid or forceful manoeuvres. The double-integrator model provides a more accurate and physically consistent framework for the control of robotic systems, particularly in situations where control inputs influence acceleration directly rather than velocity.

Consider a team of N robots, each robot $i \in 1, \ldots, N$ is modeled by a double integrator dynamic:

$$\begin{cases} \dot{p}_i = v_i, \\ \dot{v}_i = u_i, \end{cases} \tag{1}$$

where, $p_i \in \mathbb{R}^n, v_i \in \mathbb{R}^n$ and $u_i \in \mathbb{R}^n$ are the position,

velocity and control input (acceleration) respectively of robot i. The state vector is $x_i = [p_i^T, v_i^T]^T \in \mathbb{R}^{2n}$, with dynamics:

$$\dot{x}_i = F x_i + G u_i, \tag{2}$$

where,

$$F = \begin{bmatrix} 0_{n \times n} & I_n \\ 0_{n \times n} & 0_{n \times n} \end{bmatrix} \in \mathbb{R}^{2n \times 2n}, G = \begin{bmatrix} 0_{n \times n} \\ I_n \end{bmatrix} \in \mathbb{R}^{2n \times n}.$$

The global state of the team of N robots is given by:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^{2nN}.$$

The global dynamics of the robot team are then:

$$\dot{X} = \mathcal{F} X + \mathcal{G} U, \tag{3}$$

where,

$$\mathcal{F} = \begin{bmatrix} F & 0_{n \times n} & \cdots & 0_{n \times n} \\ 0_{n \times n} & F & \cdots & 0_{n \times n} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{n \times n} & 0_{n \times n} & \cdots & F \end{bmatrix} \in \mathbb{R}^{2nN \times 2nN},$$

$$\mathcal{G} = \begin{bmatrix} G & 0_{n \times n} & \cdots & 0_{n \times n} \\ 0_{n \times n} & G & \cdots & 0_{n \times n} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{n \times n} & 0_{n \times n} & \cdots & G \end{bmatrix} \in \mathbb{R}^{2nN \times nN}, U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} \in \mathbb{R}^{nN}.$$

2.2 Desired formation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with vertex set $\mathcal{V} = \{1, \ldots, N\}$ (robots) and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges (connections links), the neighbors set of agent i is defined as:

$$\mathcal{N}_i = \{j \mid (j, i) \in \mathcal{E}, i \neq j\}. \tag{4}$$

A desired formation is defined by desired relative positions $\delta_{ij}^d \in \mathbb{R}^n$ between robots i and j by:

$$p_i(t) - p_j(t) = \delta_{ij}^d, \quad \forall (i, j) \in \mathcal{E}. \tag{5}$$

Remark 1. In practical terms, achieving a strictly rigid formation is challenging. The formation should be nearly rigid: the robots must maintain a desired or target formation while allowing for adjustments, when necessary, such as avoiding collisions or modifying their trajectories to reach the goal.

The formation constraint can be reformulated as follows:

$$\| p_i(t) - p_j(t) - \delta_{ij}^d \| \leq \varepsilon \quad \forall (i, j) \in \mathcal{E}, \tag{6}$$

where, ε represents a tolerance that provides flexibility in maintaining the distance between the robots. This constraint

allows the robots to have a small deviation from the desired distance, ensuring the formation remains cohesive while accommodating practical adjustments.

2.3 Deformation and reformation process

When navigating through constrained environments, the formation scales and rotates the offsets to avoid collisions with obstacles and between robots:

$$\delta_{ij}^{adapt}(t) = \Gamma(t)\delta_{ij}^{d}, \tag{7}$$

where, $\Gamma(t) \in \mathbb{R}^{n \times n}$ is the deformation matrix. After avoiding obstacles, the multi-robot system should temporarily adjust its formation shape and then autonomously return to its original configuration with stable and guaranteed recovery. In practice, a rigid formation cannot navigate safely, for example, to pass through a narrow corridor, bypass a large obstacle, or navigate over uneven terrain. To overcome these difficulties, a deformation process (changing the geometric shape) is essential in the above situations to maintain the safety of the system.

The formation must safely navigate from an initial state to a target state by satisfying the following requirements: maintaining the desired formation along the trajectory, controlling the deformation of the geometric shape, ensuring that the robots avoid colliding during the deformation process, and ensuring system stability. Achieving these goals involves precise coordination and communication among the robots, as well as robust algorithms that can adapt to changing situations.

2.4 CBF

In a multirobot system, ensuring the safety of the robots, particularly avoiding collisions, is of paramount importance. CBF provides a robust framework for guaranteeing safety in dynamic systems by enforcing state constraints that must be satisfied at all times [13-15]. In this section, we apply CBF to a system of N robots, each robot is modelled by a double integrator dynamic, to ensure that the robots maintain a minimum distance from each other and avoid collisions during their motion. To ensure that the robots do not collide during their movement, we define a control barrier function that enforces a minimum distance between each pair of robots.

For each pair of robots i and j, defining a function $h_{ij}: \mathbb{R}^{2n} \to \mathbb{R}$ by:

$$h_{ij}(p_i, p_j) = \| p_i - p_j \|^2 - d_{min}^2, \tag{8}$$

where, d_{min} is the desired minimum distance between robots. To ensure collision avoidance, the barrier function must remain be non-negative:

$$h_{ij}(p_i, p_j) \geq 0 \quad \forall i \neq j. \tag{9}$$

Therefore, the control input $u_i(t)$ must be adjusted to guarantee that this condition is maintained over time, taking the time derivative of $\dot{h}_{ij}$, we obtain:

$$\dot{h}_{ij} = 2(p_i - p_j)^T (v_i - v_j). \tag{10}$$

To guarantee the forward invariant of the safe set $\mathcal{C}$ defined

by:

$$\mathcal{C} = \{p_i, p_j \mid h_{ij}(p_i, p_j) \geq 0\}, \tag{11}$$

We enforce the condition [13]:

$$\dot{h}_{ij} \geq -\alpha(h_{ij}), \tag{12}$$

where, $\alpha(\cdot): \mathbb{R} \to \mathbb{R}$ is a strictly increasing function [25], typically chosen as a class $\mathcal{K}$ function such that $\alpha(0) = 0$, which ensures that when the robots get too close, the rate of change of h_{ij} is limited, forcing the robots to move away from each other.

Remark 2. Standard first-order CBFs are insufficient for this system since they are designed for systems where the control input directly influences the first derivative of the state. In this case, the control input doesn't show up directly in the time derivative of the barrier function. This is why HOCBFs are used to enforce safety constraints at the acceleration level.

2.5 HOCBFs

HOCBFs are necessary for systems with a relative degree greater than one, such as double integrator dynamics, to ensure safety by explicitly incorporating higher-order state derivatives (e.g. acceleration inputs) into constraints, thus guaranteeing collision avoidance even when the control input does not directly affect the first derivative of the safety condition.

$$\ddot{h}_{ij} = \frac{d}{dt}\dot{h}_{ij} = 2 \| v_i - v_j \|^2 + 2(p_i - p_j)^T (u_i - u_j). \tag{13}$$

The safety constraint thus becomes:

$$\ddot{h}_{ij} + \alpha_1(\dot{h}_{ij}) + \alpha_2(h_{ij}) \geq 0, \tag{14}$$

where, $\alpha_1(\cdot)$ and $\alpha_2(\cdot)$ are functions that are strictly increasing, typically chosen as class $\mathcal{K}$ functions with $\alpha_1(0) = 0$ and $\alpha_2(0) = 0$.

This guarantees the forward invariance of the safe set, meaning that the robots will maintain the required minimum distance from each other over time.

3. RESULTS AND DISCUSSION

This section provides numerical simulations using MATLAB R2024b to demonstrate the effectiveness of the proposed approach through three representative examples.

3.1 Navigation without deformation

Consider a team of N robots moving from an initial state x^{init} to a final state x^f while maintaining a desired formation along the trajectory, the initial and final states of each robot i are given by:

$$x_i^{init}(0) = \left[p_i(0)^T, v_i(0)^T \right]^T, \quad x_i^f = \left[p_i^f, v_i^f \right]^T. \tag{15}$$

To ensure that the robots maintain the desired formation,

avoid collisions, and minimize the control effort, we formulate the problem as a quadratic programming (QP) problem:

$$\min_u \frac{1}{2} \sum_{i=1}^{N} \| u_i - u_{des} \|^2 + \lambda \sum_{i<j} \| p_i - p_j - \delta_{ij}^d \|^2,$$ (16)

Subject to the following constraints:

$$\| p_i - p_j - \delta_{ij}^d \| \leq \varepsilon \quad \forall i \neq j, \quad \| u_i \| \leq u_{max} \quad \forall i,$$
$$2(p_i - p_j)^T (u_i - u_j) \geq -2 \| v_i - v_j \|^2 - \alpha_1(\dot{h}_{ij}) - \alpha_2(h_{ij}), \forall i \neq j,$$

where u_{des} is the desired control input and u_{max} is the maximum control input, λ is a weighting factor.

Remark 3. As the number of robots increases, the global quadratic optimization problem becomes computationally expensive. To reduce the computational cost, each robot solves its own QP problem in a distributed manner, considering only local interactions with its neighbors.

Each robot minimizes its own local cost function subject to constraints while exchanging information with its neighbors (e.g., relative positions p_j, velocities v_j and desired formation offsets δ_{ij}^d). For each robot i, the distributed optimization problem is formulated as:

$$\min_{u_i} \frac{1}{2} \| u_i - u_{des} \|^2 + \lambda \sum_{j \in \mathcal{N}_i} \| p_i - p_j - \delta_{ij}^d \|^2,$$ (17)

subject to:

$$\| p_i - p_j - \delta_{ij}^d \| \leq \varepsilon \quad \forall j \in \mathcal{N}_i, \quad \| u_i \| \leq u_{max} \quad \forall i,$$
$$2(p_i - p_j)^T (u_i - u_j)$$ (18)
$$\geq -2 \| v_i - v_j \|^2 - \alpha_1(\dot{h}_{ij}) - \alpha_2(h_{ij}), \quad \forall j \in \mathcal{N}_i.$$

Example 1. Consider 5 robots positioned on a circle with center (1,1) and radius r=2. They navigate from an initial state at (1,1) to a final state at (10,10) while maintaining the desired formation. The initial and final positions are illustrated in Figure 1 and Figure 2, respectively.
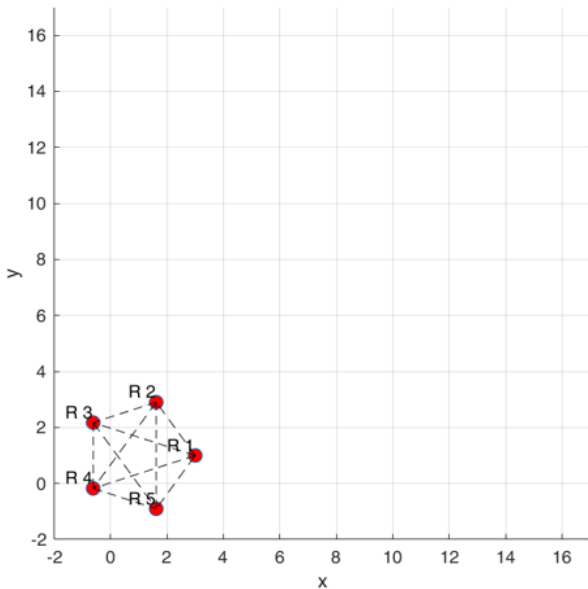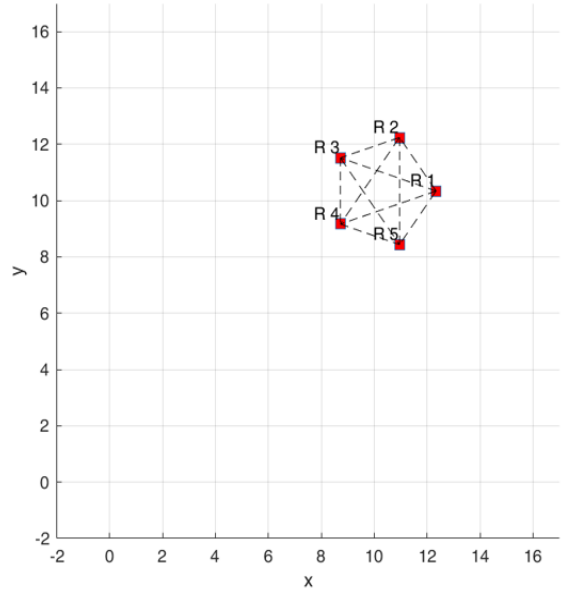

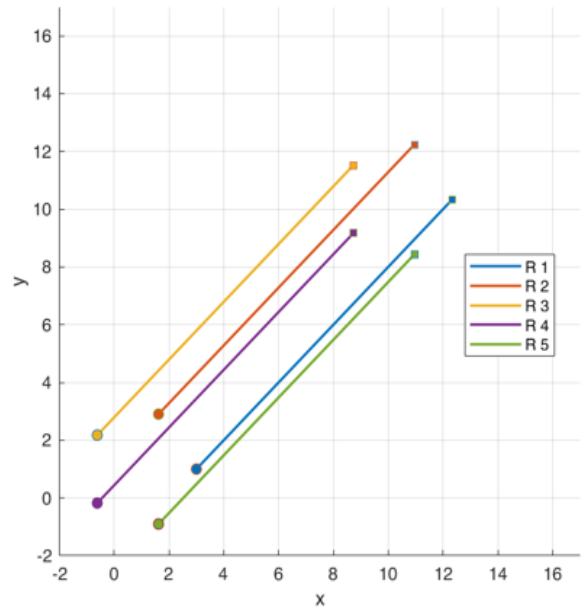**Figure 2.** Final formation

Figure 3. Robot trajectories

The five-robot pentagon moves rigidly toward the goal while preserving its geometry shape, non-intersecting paths verify collision-free motion and near-zero formation error, as shown in Figure 3, where the trajectories of the robots are illustrated.

3.2 Navigation with deformation (scale and rotation)

In practice, it is not feasible to achieve a strictly rigid deformation. Therefore, we introduce a deformation tolerance $\varepsilon^{adapt} > 0$ as follows:

$$\| \delta_{ij}^{adapt}(t) - \Gamma(t)\delta_{ij}^d \| \leq \varepsilon^{adapt}.$$ (19)

Remark 4. The deformation matrix $\Gamma(t)$ remains invertible for all t. This invertibility ensures that the formation can always be recovered to its original shape whenever external constraints or collision-avoidance manoeuvres no longer

Figure 1. Initial formation

apply.

For simplicity, we focus on the two-dimensional setting. However, the same approach can be extended analogously to higher dimensions $n \geq 2$ without fundamental change in the methodology.

The deformation matrix is given as:

$$\Gamma(t) = s(t) \cdot R(\theta(t)), \tag{20}$$

where, $s(t)$ is a scalar function (the scale factor) that allows the formation to expand or contract in the plane, $R(\theta(t))$ is the rotation matrix given by:

$$R(\theta(t)) = \begin{bmatrix} \cos(\theta(t)) & -\sin(\theta(t)) \\ \sin(\theta(t)) & \cos(\theta(t)) \end{bmatrix}. \tag{21}$$

There are numerous ways to specify the time-dependent scale $s(t)$ and rotation $\theta(t)$. Two illustrative cases are:

1. Linear Variations:
   $s(t) = 1 + \beta_1 t$, β_1 is a small constant rate of expansion or contraction, $\theta(t) = \beta_2 t$, β_2 is a constant rotational speed.
2. Sigmoidal Variations:
   $s(t) = \frac{1}{1+e^{-\beta(t-t_0)}}$, $\beta > 0$, t_0 represents the midpoint of the switch, $\theta(t) = \theta_0 + \gamma s(t)$, θ_0 is the initial angle of the formation, γ is a rotation factor (or rotation gain).

Example 2. Consider 5 robots in an initial state arranged in a circle of radius $r=2$ with the formation center at $(1,1)$. In this example, a simple deformation (comprising both scaling and rotation) is applied to adjust the formation as the robots navigate toward a final state with the center at $(10,10)$ while maintaining the desired formation, as illustrated in Figure 4 for the initial position and Figure 5 for the final position.

The multi-robot team translates from its initial coordinates to the target location while simultaneously scaling and rotating the formation, maintaining strict inter-agent separation and achieving collision-free trajectories throughout the maneuver, as shown in Figure 6.
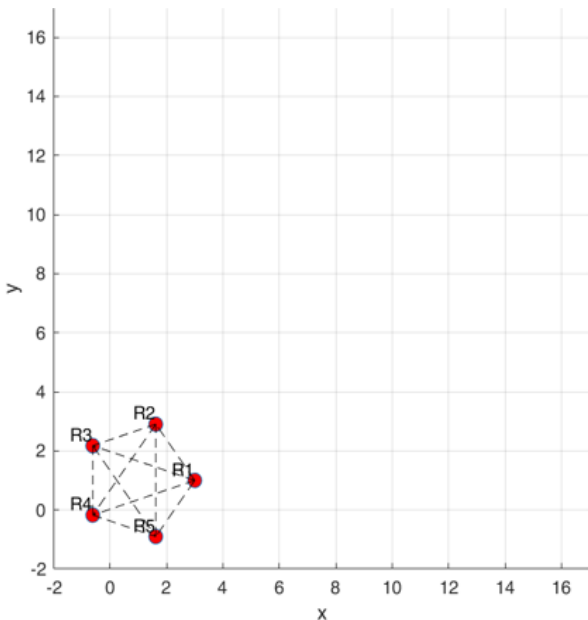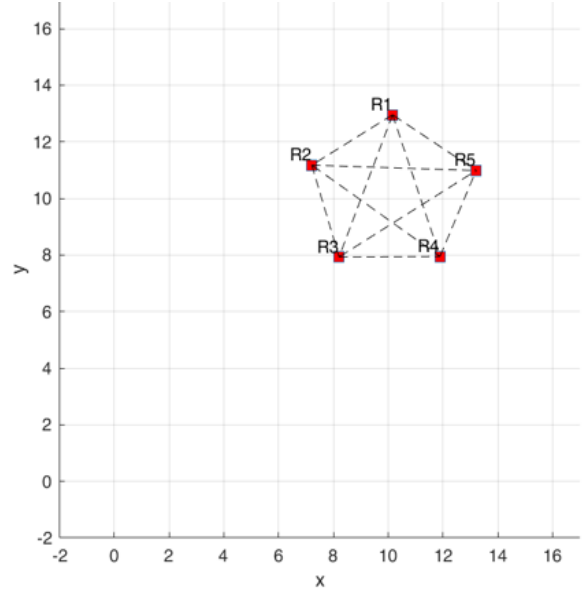

Figure 4. Initial formation

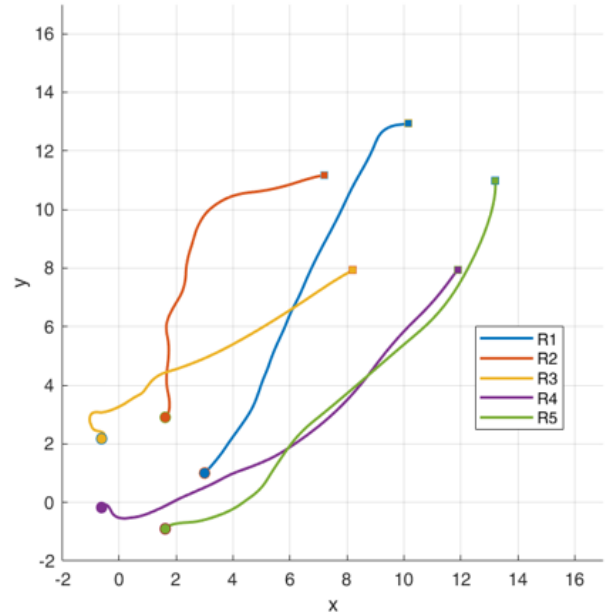**Figure 5.** Final formation

Figure 6. Robot trajectories

3.3 Navigation with deformation (shape change)

Let $\delta_{ij}^{Sh_1}$ and $\delta_{ij}^{Sh_2}$ denote the desired relative positions between robots i and j corresponding to the shapes Sh_1 and Sh_2, respectively. We introduce a continuous transition function $\beta(t) \in [0,1]$ such that:

- $\beta(0) = 0$: Initially, the formation is Sh_1.
- $\beta(t) = 1$, for all $t \geq T_{trans}$: After a certain transition time, the formation becomes Sh_2.

The desired relative offset is defined as:

$$\delta_{ij}^d = \left(1 - \beta(t)\right)\delta_{ij}^{Sh_1} + \beta(t)\delta_{ij}^{Sh_2}. \tag{22}$$

Remark 5. The following functions can be used like a transition functions:

- $\beta(t) = \min\left\{1, \dfrac{1}{T_{trans}}\right\}.$

$$\beta(t) = \begin{cases} 0 & t \leq t_{start}, \\ \dfrac{1}{1 + e^{-k(t-t_0)}} & t_{start} < t < t_{end}, \\ 1 & t \geq t_{end}. \end{cases}$$

where, $k > 0$ is a parameter, t_0 is the midpoint (which means that the transition is halfway complete at $t = t_0$, t_{start}, t_{end} are the start and end transitions, respectively.

The distributed QP problem is solved locally by each robot based on the information from its neighbors:

$$\min_{u_i} \frac{1}{2} \| u_i - u_{des} \|^2 + \lambda \sum_{j \in \mathcal{N}_i} \| p_i - p_j - \delta_{ij}^d \|^2 . \qquad (23)$$

This formulation allows each robot to use local information (via distributed consensus) to adjust its control input and drive the formation from shape Sh_1 to shape Sh_2 over time.

Example 3. Consider 5 robots in an initial state arranged in a circle of radius r=2 with the formation center at $(1,1)$. In this example, the formation transitions from a circle to a triangle while simultaneously navigating toward the target.
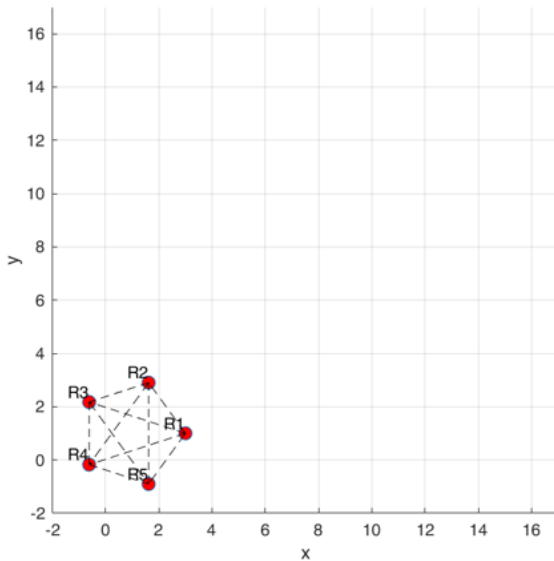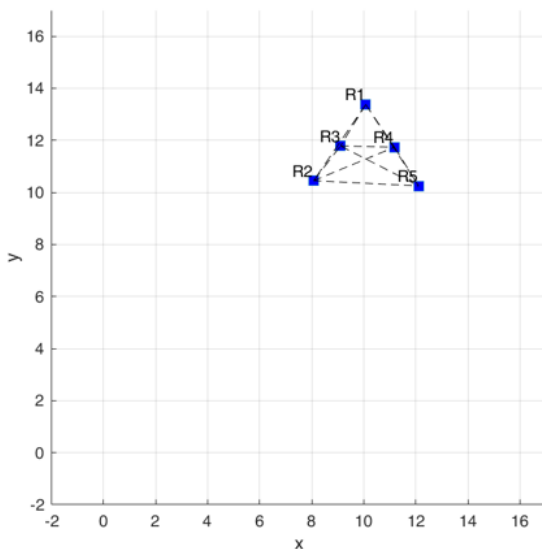

Figure 7. Initial formation
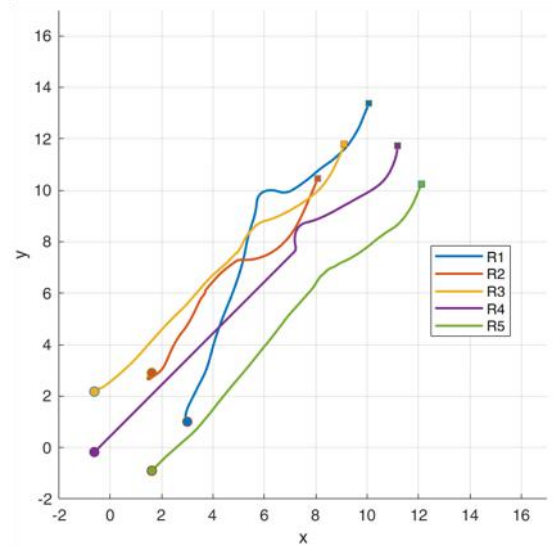

Figure 8. Final formation

Figure 9. Robot trajectories

As shown in Figures 7-9, the five-robot team successfully executes a dual objective translation toward the goal and continuous morphing from an initial circular lattice (radius 2 m, center at $(1,1)$) to a terminal triangular pattern without a single inter-robot collision. The smooth, non-intersecting trajectories confirm that the CBF-QP supervisor enforces the prescribed minimum separations at every instant, even while accommodating the coupled inertial dynamics and the non-convex deformation path. These results underscore the controller's ability to orchestrate complex, real-time formation reshaping while rigorously maintaining safety guarantees.

4. CONCLUSIONS

This work proposes an integrated, real-time control architecture for formation maintenance in multi-robot systems governed by double-integrator dynamics. The method accounts for inertial forces, enables elastic shape morphing, and enforces inter-robot collision avoidance. Both rigid and deformable spacing requirements are encoded via a control barrier function-based quadratic program, which guarantees forward invariance of a formally defined safe formation set. Theoretical analysis establishes a strictly positive lower bound on inter-robot distances, ensuring that robots remain within the prescribed (possibly time-varying) formation geometry. Extensive simulations confirm that the controller preserves formation coherence and achieves collision-free motion without compromising agility.

Scalability is achieved by decentralising the QP via consensus-based optimization, allowing each robot to solve a low-dimensional problem with limited communication while maintaining global safety guarantees. This combination of flexible shape control, certified safety, and distributed computation provides a principled and computationally tractable pathway for large scale robot deployments.

While the proposed method showed promising performance for collision-free formation control under double-integrator dynamics, it has significant drawbacks needing further exploration. First, our present method assumes a well-structured environment with no static or dynamic obstacles. In scenarios including fixed or moving obstacles, creating a single CBF that incorporates both inter-robot and robot-to-

obstacle constraints may result in non-differentiability at the intersection of several safety sets. This constraint restricts the implementation of gradient-based optimization approaches and could impact real-time performance and practicality. Second, neighbor-based distributed optimization is based on a fixed and established communication topology. When robots start from random placements, the cardinality and structure of each robot's neighborhood might change dynamically, complicating real-time updates to formation restrictions and local CBFs. Handling these topological changes while maintaining stability and safety requirements is a remaining challenge. Finally, our method does not explicitly account model uncertainties or external disturbances. Incorporating robust control or adaptive techniques would make the approach more applicable to real-world settings.

Future work will focus on (i) hardware validation with on-board perception and latency-aware communication, (ii) robustness extensions that accommodate model uncertainty and actuation limits, (iii) hierarchical planners that couple the proposed low-level controller with high-level task allocation for heterogeneous teams, and (iv) extending the controller to handle unknown environments with static and moving obstacles while guaranteeing safe and stable transitions. These directions aim to bridge the gap from simulation to real-world, mission-critical applications.

ACKNOWLEDGMENT

REFERENCES

[1] Olfati-Saber, R., Fax, J.A., Murray, R.M. (2007). Consensus and cooperation in networked multi-agent systems. Proceedings of the IEEE, 95(1): 215-233. https://doi.org/10.1109/JPROC.2006.887293

[2] Zhu, J., Lu, C., Li, J., Wang, F.Y. (2025). Secure consensus control on multi-agent systems based on improved PBFT and raft blockchain consensus algorithms. IEEE/CAA Journal of Automatica Sinica, 12(7): 1407-1417. https://doi.org/10.1109/JAS.2025.125300

[3] Peng, J., Xiao, H., Lai, G., Chen, C.P. (2024). Consensus formation control of wheeled mobile robots with mixed disturbances under input constraints. Journal of the Franklin Institute, 361(17): 107300. https://doi.org/10.1016/j.jfranklin.2024.107300

[4] Lim, Y.H., Lee, G.S. (2019). Observer-based distributed consensus algorithm for multi-agent systems with output saturations. Journal of Information and Communication Convergence Engineering, 17(3): 167-173. https://doi.org/10.6109/jicce.2019.17.3.167

[5] Oh-hara, S., Fujimori, A. (2023). A leader-follower formation control of mobile robots by position-based visual servo method using fisheye camera. ROBOMECH Journal, 10(1): 30. https://doi.org/10.1186/s40648-023-00268-6

[6] Liu, Z., Li, J., Shen, J., Wang, X., Chen, P. (2024). Leader–follower UAVs formation control based on a deep Q-network collaborative framework. Scientific Reports, 14(1): 4674. https://doi.org/10.1038/s41598-024-54531-w

[7] Antony, A., Kumar, S.R., Mukherjee, D. (2024). Artificial potential fields based formation control for fixed wing UAVs with obstacle avoidance. IFAC-PapersOnLine, 57: 19-24. https://doi.org/10.1016/j.ifacol.2024.05.004

[8] Chang, X., Jiao, J., Li, Y., Hong, B. (2024). Multi-consensus formation control by artificial potential field based on velocity threshold. Frontiers in Neuroscience, 18: 1367248. https://doi.org/10.3389/fnins.2024.1367248

[9] Wang, C., Sun, Y., Ma, X., Chen, Q., Gao, Q., Liu, X. (2024). Multi-agent dynamic formation interception control based on rigid graph. Complex & Intelligent Systems, 10(4): 5585-5598. https://doi.org/10.1007/s40747-024-01467-3

[10] Cai, X. (2013). Graph rigidity-based formation control of planar multi-agent systems. Louisiana State University and Agricultural & Mechanical College. https://doi.org/10.31390/gradschool_dissertations.1666

[11] Köhler, M., Müller, M.A., Allgöwer, F. (2023). Distributed model predictive control for periodic cooperation of multi-agent systems. IFAC-PapersOnLine, 56(2): 3158-3163. https://doi.org/10.1016/j.ifacol.2023.10.1450

[12] Carron, A., Saccani, D., Fagiano, L., Zeilinger, M.N. (2023). Multi-agent distributed model predictive control with connectivity constraint. IFAC-PapersOnLine, 56(2): 3806-3811. https://doi.org/10.1016/j.ifacol.2023.10.1310

[13] Ames, A.D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., Tabuada, P. (2019). Control barrier functions: Theory and applications. In 2019 18th European control conference (ECC), Naples, Italy, pp. 3420-3431. https://doi.org/10.23919/ECC.2019.8796030

[14] Alan, A., Taylor, A.J., He, C.R., Ames, A.D., Orosz, G. (2023). Control barrier functions and input-to-state safety with application to automated vehicles. IEEE Transactions on Control Systems Technology, 31(6): 2744-2759. https://doi.org/10.1109/TCST.2023.3286090

[15] Reis, M.F., Aguiar, A.P., Tabuada, P. (2020). Control barrier function-based quadratic programs introduce undesirable asymptotically stable equilibria. IEEE Control Systems Letters, 5(2): 731-736. https://doi.org/10.1109/LCSYS.2020.3004797

[16] Kiss, A.K., Molnar, T.G., Ames, A.D., Orosz, G. (2023). Control barrier functionals: Safety-critical control for time delay systems. International Journal of Robust and Nonlinear Control, 33(12): 7282-7309. https://doi.org/10.1002/rnc.6751

[17] Dawson, C., Gao, S., Fan, C. (2023). Safe control with learned certificates: A survey of neural Lyapunov, barrier, and contraction methods for robotics and control.

IEEE Transactions on Robotics, 39(3): 1749-1767. https://doi.org/10.1109/TRO.2022.3232542

[18] Brunke, L., Greeff, M., Hall, A.W., Yuan, Z., Zhou, S., Panerati, J., Schoellig, A.P. (2022). Safe learning in robotics: From learning-based control to safe reinforcement learning. Annual Review of Control, Robotics, and Autonomous Systems, 5(1): 411-444. https://doi.org/10.1146/annurev-control-042920-020211

[19] Robey, A., Lindemann, L., Tu, S., Matni, N. (2021). Learning robust hybrid control barrier functions for uncertain systems. IFAC-PapersOnLine, 54(5): 1-6. https://doi.org/10.1016/j.ifacol.2021.08.465

[20] Jin, W., Wang, Z., Yang, Z., Mou, S. (2020). Neural certificates for safe control policies. arXiv preprint arXiv:2006.08465. https://doi.org/10.48550/arXiv.2006.08465

[21] Wang, H., Peng, J., Xu, J., Zhang, F., Wang, Y. (2023). High-order control barrier functions-based optimization control for time-varying nonlinear systems with full-state constraints: A dynamic sub-safe set approach.

International Journal of Robust and Nonlinear Control, 33(8): 4490-4503. https://doi.org/10.1002/rnc.6624

[22] Xiao, W., Belta, C. (2021). High-order control barrier functions. IEEE Transactions on Automatic Control, 67(7): 3655-3662. https://doi.org/10.1109/TAC.2021.3105491

[23] Xiao, W., Belta, C. (2019). Control barrier functions for systems with high relative degree. In 2019 IEEE 58th Conference on Decision and Control (CDC), Nice, France, pp. 474-479. https://doi.org/10.1109/CDC40024.2019.9029455

[24] Tan, X., Cortez, W.S., Dimarogonas, D.V. (2021). High-order barrier functions: Robustness, safety, and performance-critical control. IEEE Transactions on Automatic Control, 67(6): 3021-3028. https://doi.org/10.1109/TAC.2021.3089639

[25] Xu, X., Tabuada, P., Grizzle, J.W., Ames, A.D. (2015). Robustness of control barrier functions for safety critical control. IFAC-PapersOnLine, 48(27): 54-61. https://doi.org/10.1016/j.ifacol.2015.11.152