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

Multi-Robot Segregation Using Finite-Time MPC With Chernoff Bound-Based Asynchronous Motion Smoothing

Richa Dubey, Shreyash Gupta, Saurabh Chaudhary, Niladri S. Tripathy, and Suril V. Shah, *Member, IEEE*

***Abstract*—For multi-robot systems operating in dynamic environments, collision-free segregation into a desired set of groups in finite time is an essential task requirement in many applications. This work presents a control framework for such systems, utilizing Finite-time Model Predictive Control. The objective is to guide the robots toward a segregated formation while adhering to leader-follower dynamics and effectively avoiding collisions. To ensure finite-time convergence, the concept of a control invariant set is incorporated. Furthermore, the letter derives an upper bound on the required time steps for the robots to achieve the segregated formation. In order to maintain a smooth motion profile in the face of external state perturbations, this work proposes a data-driven Chernoff bound-based triggering method that enables Asynchronous Motion Smoothing for the robots. To validate the effectiveness of the proposed control framework, both simulations and hardware experiments are conducted, focusing on the segregation of five robots into two distinct groups.**

***Index Terms*—Multi-robot system, segregation, finite-time MPC, asynchronous motion smoothing, Chernoff bound.**

I. Introduction

MULTI-ROBOT systems often operate in dynamic environments, where they face challenges such as collisions, sensor noise, and external disturbances like jerks. In real-time applications, tasks are often executed sequentially, with segregation being a key sub-task. During segregation, robots divide into distinct groups that later collaborate to achieve a common goal [1], [2], [3], as seen in applications like surveillance, search and rescue, and foraging. Given its foundational role, ensuring finite-time convergence for segregation is crucial in such scenarios. Furthermore, to achieve this, effective collision avoidance and motion smoothing are essential, particularly in the presence of external perturbations. This work focuses on developing a control framework that ensures finite-time convergence during segregation, while also addressing the challenges of collision avoidance and mitigating the impact of external perturbations.

In the past decade, several works have addressed the segregation problem in multi-robot systems. The artificial potential-based approaches in [1], [3] achieve segregation; however, both of them do not address collisions and [1] lacks convergence analysis too. A differential potential based approach in [2] ensures asymptotic convergence to segregation. In contrast, [4] employs Particle Swarm Optimization with an extension of Optimal Reciprocal Collision Avoidance to segregate robots. Model Predictive Control (MPC) has also been extensively applied for multi-robot systems because of its predictive capability, constraint handling, and adaptability [5]. For instance, MPC has been used for motion planning [6], formation control [7], and collision avoidance via constraints [8]. [9] used MPC to segregate a multi-robot system, however, without convergence analysis. While event- and self-triggered MPC has also been explored for controlling nonholonomic systems with collision avoidance [10] and finite-time convergence [11], a task-specific formula for the maximum number of steps required in multi-robot settings is typically missing.

There is existing literature on multi-robot systems [2], [3], [8], [9], [10], [12], but theoretical guarantees for finite-time convergence in the context of the segregation problem remain relatively limited. The MPC cost function is modified in [13], [14] to achieve finite-time convergence of a single agent to the desired region. However, the discontinuity of such a cost function at the boundary of the desired region poses issues during optimization. To overcome these issues, a desired set-based cost function is proposed in [15] for a single agent. Nevertheless, establishing finite-time convergence for multi-robot segregation problem in dynamic environments presents an additional challenge. While operating in dynamic environments, multi-robot systems are also faced with external perturbations in robot states, which cause abrupt jitters in their motion profiles [16]. To address these jitters, an Asynchronous Path Smoothing method is incorporated within the MPC framework in [9]. This method employs a heuristic threshold to decide when the perturbations in the robot states are substantial enough to necessitate relaying sensor data to the MPC, thereby imparting smooth path profiles to the robots. On similar lines, a heuristically-derived threshold in the MPC framework [8] is used for path smoothing. While using such heuristic thresholds is convenient, they are often tailored to specific scenarios, lacking a consistent basis grounded in theoretical or statistical analysis. Moreover, thresholds chosen by intuition or trial-and-error may lead to non-reproducible outcomes, hindering scalability and standardization in real-world

Received 19 May 2025; accepted 16 September 2025. Date of publication 29 September 2025; date of current version 13 October 2025. This article was recommended for publication by Associate Editor G. Michieletto and Editor C. Della Santina upon evaluation of the reviewers' comments. *(Corresponding author: Richa Dubey.)*

Richa Dubey and Niladri S. Tripathy are with the Department of Electrical Engineering, Indian Institute of Technology, Jodhpur 342037, India (e-mail: richa.1@iitj.ac.in; niladri@iitj.ac.in).

Shreyash Gupta, Saurabh Chaudhary, and Suril V. Shah are with the Department of Mechanical Engineering, Indian Institute of Technology, Jodhpur 342037, India (e-mail: gupta.50@iitj.ac.in; chaudhary.10@iitj.ac.in; surilshah@iitj.ac.in).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3615524, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3615524

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

TABLE I
OVERVIEW OF COMPARISON WITH EXISTING WORKS

Articles	Comparison criterion				
	A	B	C	D	E
Luis et al. [8]	✗	✗	✓	✗	✓
Anderson et al. [15], Hashimoto et al. [11]	✗	✓	✗	✗	✗
Gupta et al. [9]	✓	✗	✓	✗	✓
Schlüter et al. [18]	✗	✗	✗	✓	✓
Eqtami et al. [10]	✗	✗	✓	✗	✗
Proposed work	✓	✓	✓	✓	✓

A: Multi-robot segregation with motion smoothing B: Finite-time convergence
C: Collision avoidance D: Data-driven threshold E: Hardware implementation

applications [17]. Consequently, there is an evident need for systematically derived or data-driven thresholds that consider the multi-robot dynamics and interaction complexities for improved, consistent performance in multi-robot applications.

In light of these research gaps associated with multi-robot segregation, the objective of this work is to address the lack of finite-time convergence guarantee, prevent inter-robot collisions, and formulate a systematic method to derive thresholds. In this work, we propose a Finite-time MPC-based control framework for a multi-robot system with leader-follower error dynamics and an on-demand collision avoidance constraint to achieve finite-time convergence to segregated formation. We also derive an upper bound on the convergence time required for segregation. Furthermore, a data-driven Chernoff bound-based threshold is proposed for Asynchronous Motion Smoothing (AMS) to effectively detect and mitigate the effect of perturbations in robot states. Table I presents the comparison of this work with existing works in terms of key criteria, highlighting the research gaps. Our key contributions are:

- A Finite-time MPC-based control framework using invariant sets for segregating multi-robot systems with leader-follower error dynamics and collision avoidance.
- Derivation of analytical expression of upper bound on the time required to converge to the segregated formation.
- Formulation of a data-driven (Chernoff bound-based) threshold to enable Asynchronous Motion (AMS) Smoothing in the presence of perturbations in robot states.
- Performance validation of the proposed framework via both simulations and hardware experiments in the presence of dynamic obstacles and perturbations.

II. PROBLEM DESCRIPTION AND PROPOSED SOLUTION

Consider[1] a multi-robot system consisting of n robots to be segregated into m groups ($m \leq n$), where each group has one leader and others are followers. To perform segregation, we utilize the following discrete-time error dynamics for i^{th} robot

$$\boldsymbol{e}_i[k+1] = \boldsymbol{A}\boldsymbol{e}_i[k] + \boldsymbol{B}\boldsymbol{u}_{\boldsymbol{e}_i}[k], \text{ for } i \in \{1, 2, \ldots, n\}, \quad (1)$$

where $\boldsymbol{e}_i[k] = \boldsymbol{x}_d[k] - \boldsymbol{x}_i[k]$, $\boldsymbol{u}_{\boldsymbol{e}_i}[k] = \boldsymbol{u}_d[k] - \boldsymbol{u}_i[k]$ and discrete-time index $k \in \mathbb{N}$. Here, $\boldsymbol{x}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^\intercal$, where $x_i, y_i \in \mathbb{R}$ are the position coordinates, $\boldsymbol{u}_i = \begin{bmatrix} \dot{x}_i & \dot{y}_i \end{bmatrix}^\intercal$ denotes the velocity input in x and y directions, and $\boldsymbol{x}_d$ and $\boldsymbol{u}_d$ are the desired state and input, respectively. For brevity, we omit the subscript i subsequently. The error and input constraints are $\boldsymbol{e}[k] \in \mathcal{E} \subset \mathbb{R}^2$ and $\boldsymbol{u}_e[k] \in \mathcal{U} \subset \mathbb{R}^2$, where $\mathcal{E}$ is a closed set and $\mathcal{U}$ is a compact set, both containing the origin in their interior.

The segregated formation of robots is defined such that the leader of each group aims to attain a desired steady-state position $\boldsymbol{x}_s$. For each leader, $\boldsymbol{x}_s$ is set beforehand based on the desired segregated formation of each group. Consequently, for the leader, $\boldsymbol{x}_d[k] = \boldsymbol{x}_s$. Let $\boldsymbol{e}_s$ denote the desired steady-state error. Then, for the leaders, we propose $||\boldsymbol{e}_s|| = 0$ to ensure their convergence to $\boldsymbol{x}_s$ at steady-state. The followers, however, aim to track the positions of their leaders. Therefore, for the followers, we define convergence to segregated formation as arranging themselves at a distance r around their respective leaders at steady-state. Hence, for the followers, we propose $||\boldsymbol{e}_s|| = r$. Next, we utilize $\boldsymbol{e}_s$ and the concept of λ-Control Invariant Set (λ-CIS) [19] to obtain the desired set of error for the robots, denoted by $\Omega_{\boldsymbol{e}} \subset \mathcal{E}$.

For dynamics (1), given $\lambda \in (0, 1]$, $\Omega_{\boldsymbol{e}} \subseteq \mathbb{R}^2$ is a λ-CIS iff $\Omega_{\boldsymbol{e}} \subseteq \mathcal{E}$ and $\boldsymbol{e}[k] \in \Omega_{\boldsymbol{e}} \Rightarrow \exists\, \boldsymbol{u}_e[k] \in \mathcal{U} \mid \mathbf{e}[k+1] \in \lambda\Omega_{\boldsymbol{e}}$. For $\lambda = 1, \Omega_{\boldsymbol{e}}$ is simply a CIS. The input set $\Phi(\Omega_{\boldsymbol{e}}) = \{\boldsymbol{u}_e[k] \in \mathcal{U} : \exists\, \boldsymbol{e}[k] \in \Omega_{\boldsymbol{e}} \mid \boldsymbol{e}[k+1] \in \lambda\Omega_{\boldsymbol{e}}\}$. Then, 1-step controllable set to $\Omega_{\boldsymbol{e}}$ corresponding to $\Phi(\Omega_{\boldsymbol{e}})$ is given as $\Theta_1(\Omega_{\boldsymbol{e}}, \Phi(\Omega_{\boldsymbol{e}})) : \boldsymbol{e}[k] \in \Theta_1(\Omega_{\boldsymbol{e}}, \Phi(\Omega_{\boldsymbol{e}})) \Rightarrow \exists\, \boldsymbol{u}_e[k] \in \Phi(\Omega_{\boldsymbol{e}}) \mid \boldsymbol{e}[k+1] \in \Omega_{\boldsymbol{e}}$ [15]. For $m \in \mathbb{N}$, it can be extended to obtain m-step controllable set. The multi-robot system achieves segregated formation when error converges to $\Omega_{\boldsymbol{e}}$. Once segregated, each group of robots proceeds to perform other tasks following their respective leaders. Therefore, for finite-time convergence of multi-robot system to segregated formation, it is required that

$$\boldsymbol{e}[\tau] \in \Omega_{\boldsymbol{e}}, \text{ in some finite time step } \tau \in \mathbb{N}.$$

Consequently, $\Phi(\Omega_{\boldsymbol{e}})$ serves as the desired set of inputs and it is required that $\boldsymbol{u}_{\boldsymbol{e}}[k] \in \Phi(\Omega_{\boldsymbol{e}})$. Additionally, it is crucial to address the issue of inter-robot collisions. In this work, we assume that for each robot, the other robots serve as dynamic obstacles. Let d_{ij} denote the distance between robots i and j. Then, a collision avoidance strategy is required such that

$$d_{ij} \geq r_{\min}, \quad (2)$$

where $r_{\min}$ is the minimum distance to avoid collision. Furthermore, the robot states may suffer from external perturbations that could deviate them from their desired trajectories. Threshold-based methods like AMS can detect and mitigate deviations to ensure smooth motion, but determining appropriate thresholds for multi-robot systems in dynamic environments remains a challenge. In light of these challenges, the problem statement of this work is summarized below.

Problem statement: Formulating a control framework for a multi-robot system having leader-follower error dynamics to achieve finite-time convergence to segregated formation while avoiding collisions, and deriving a statistically grounded method to obtain a bound that detects perturbations in robot states, thereby enabling Asynchronous Motion Smoothing.

To address the problem statement, we propose a Finite-time MPC framework with data-driven AMS, as shown in Fig. 1. The subsequent sections present the details of this framework.

[1]Notations: $\mathbb{R}$ and $\mathbb{N}$ denote the set of all real and natural numbers, respectively. $P \oplus Q$ denotes the minkowski sum of sets P and Q. $\boldsymbol{I}$ denotes identity matrix of compatible dimensions. For $S \subset \mathbb{R}^n$, S° is a set of all interior points and $\partial(S)$ is the set of all the boundary points of S. $\lambda_{\max}(\boldsymbol{M})$ denotes the maximum eigenvalue of matrix $\boldsymbol{M}$, and $\boldsymbol{M} = \text{diag}(\boldsymbol{P}, \boldsymbol{P}, \ldots, \boldsymbol{P})$ is a diagonal matrix formed of matrix $\boldsymbol{P}$ in each of its diagonal elements. $\mathbb{P}[X = x]$ is the probability that $X = x$, $\mathbb{E}[X]$ is the expectation of X, $\text{Var}(X)$ is the variance of X, and $\text{Cov}(X, Y)$ is the covariance of X and Y.

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

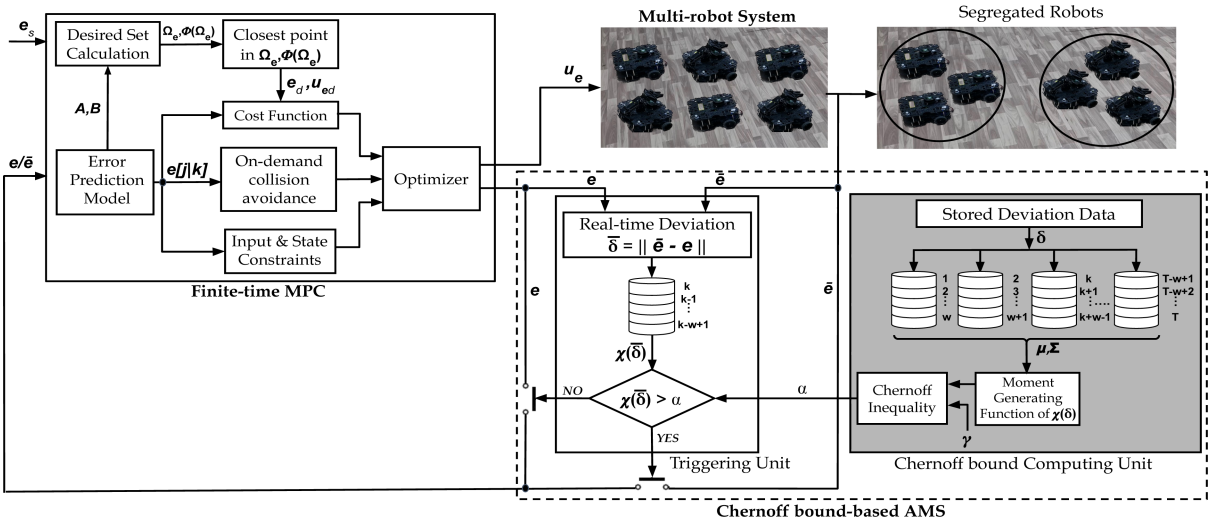

Fig. 1. Schematic of the proposed framework: The Finite-time MPC block computes optimal control inputs using predicted errors, desired set information and on-demand collision avoidance constraint, yielding multi-robot segregation. The AMS unit monitors real-time deviations and uses Chernoff bound-based triggering to relay measured states to MPC only when necessary, and relays predicted states otherwise. Here, the Chernoff bound (α) is computed offline using stored deviation data.

III. Finite-Time MPC for Segregation

A. Prediction Model and Cost Function

The Finite-time MPC calculates the optimal control input using an error prediction model for every robot, given by

$$\boldsymbol{e}[j+1|k] = \boldsymbol{A}\boldsymbol{e}[j|k] + \boldsymbol{B}\boldsymbol{u_e}[j|k],\ j \in \{0,1,\ldots,(N-1)\} \tag{3}$$

where $\boldsymbol{e}[j|k]$ and $\boldsymbol{u_e}[j|k]$ are the predicted error and input at the $(j+k)^{th}$ instant, and N is the prediction horizon. Then, (3) can also be expressed in stacked form as

$$\boldsymbol{E}_k = \Upsilon \boldsymbol{e}[0|k] + \Psi \boldsymbol{U}_k, \tag{4}$$

where $\boldsymbol{e}[0|k]$ is the error at k^{th} instant, $\boldsymbol{E}_k = \big[e[\mathbf{1}|\mathbf{k}], \boldsymbol{e}[\mathbf{2}|\mathbf{k}], \ldots \boldsymbol{e}[\mathbf{N}|\mathbf{k}]\big]^\intercal \in \mathbb{R}^{2\mathbf{N}\times\mathbf{1}}$ and $\boldsymbol{U}_k = \big[u_e[\mathbf{0}|\mathbf{k}], \boldsymbol{u_e}[\mathbf{1}|\mathbf{k}], \ldots \boldsymbol{u_e}[\mathbf{N}-\mathbf{1}|\mathbf{k}]\big]^\intercal \in \mathbb{R}^{2\mathbf{N}\times\mathbf{1}}$ are stacked predicted errors and inputs, respectively, and $\Upsilon \in \mathbb{R}^{2N\times 2}$ and $\Psi \in \mathbb{R}^{2N\times 2N}$ are as (7) in [9]. Then, we propose the desired error at j^{th} prediction $\boldsymbol{e}_{kd}[j]$ as the point closest to $\boldsymbol{e}[0|k] \in \Omega_{\boldsymbol{e}}$,

$$\boldsymbol{e}_{kd}[j] = \boldsymbol{e}_y : \min_{\boldsymbol{e}_y \in \Omega_{\boldsymbol{e}}} ||\boldsymbol{e}[0|k] - \boldsymbol{e}_y||. \tag{5}$$

Similarly, the desired input at j^{th} prediction $\boldsymbol{u}_{kd}[j]$ is the closest point to $\boldsymbol{u_e}[0|k]$ in $\Phi(\Omega_{\boldsymbol{e}})$, i.e.,

$$\boldsymbol{u}_{kd}[j] = \boldsymbol{u_v} : \min_{\boldsymbol{u_v} \in \Phi(\Omega_{\boldsymbol{e}})} ||\boldsymbol{u_e}[0|k] - \boldsymbol{u_v}||. \tag{6}$$

Here, we assume $\boldsymbol{u_e}[0|k] \approx \boldsymbol{u_e}[1|k-1]$ in order to find $\boldsymbol{u}_{kd}$ as $\boldsymbol{u_e}[1|k-1]$ is the previously estimated optimal input corresponding to $(k-1)^{th}$ instant. Using (3), (5) and (6), the proposed cost function can be stated as

$$\mathbf{V}(\boldsymbol{e}[k], \Omega_{\boldsymbol{e}}; \boldsymbol{U}_k) = \sum_{j=1}^{N} \big((\boldsymbol{e}_{kd}[j] - \boldsymbol{e}[j|k])^\intercal \boldsymbol{Q}(\boldsymbol{e}_{kd}[j] - \boldsymbol{e}[j|k]) + (\boldsymbol{u}_{kd}[j-1] - \boldsymbol{u_e}[j-1|k])^\intercal \boldsymbol{R}(\boldsymbol{u}_{kd}[j-1] - \boldsymbol{u_e}[j-1|k])\big), \tag{7}$$

where $\boldsymbol{e}[k] = \boldsymbol{e}[0|k]$ and $\Omega_{\boldsymbol{e}}$ are the optimization parameters, $\boldsymbol{U}_k$ is the optimization variable, $\boldsymbol{Q} \geq 0$ and $\boldsymbol{R} > 0$ are the weight matrices. Then, (7) can be stated in stacked form as $\mathbf{V}(\boldsymbol{e}[k], \Omega_{\boldsymbol{e}}; \boldsymbol{U}_k) = (\boldsymbol{E}_{kd} - \boldsymbol{E}_k)^\intercal \boldsymbol{Q}_s (\boldsymbol{E}_{kd} - \boldsymbol{E}_k) + (\boldsymbol{U}_{kd} - \boldsymbol{U}_k)^\intercal \boldsymbol{R}_s (\boldsymbol{U}_{kd} - \boldsymbol{U}_k)$, where $\boldsymbol{Q}_s = \mathrm{diag}(\boldsymbol{Q}, \boldsymbol{Q}, \ldots, \boldsymbol{Q})$ and $\boldsymbol{R}_s = \mathrm{diag}(\boldsymbol{R}, \boldsymbol{R}, \ldots, \boldsymbol{R})$, and $\boldsymbol{E}_{kd} = \big[e_{\mathbf{kd}}[\mathbf{1}], \quad \boldsymbol{e}_{kd}[2], \ldots \boldsymbol{e}_{kd}[N]\big]^\intercal$ and $\boldsymbol{U}_{kd} = [\boldsymbol{u}_{kd}[0], \quad \boldsymbol{u}_{kd}[1], \ldots \boldsymbol{u}_{kd}[N-1]]^\intercal$. At k^{th} instant, let $\mathscr{U}_N(k)$ denote the set of all admissible control input sequences that satisfy the error and control constraints, and the terminal constraint $\boldsymbol{e}[N|k] \in \Omega_{\boldsymbol{e}}$, i.e.,

$$\mathscr{U}_N(k) = \{\boldsymbol{U}_k \in \mathcal{U}^N \mid \boldsymbol{e}[j|k] \in \mathcal{E}, \boldsymbol{u_e}[j|k] \in \mathcal{U}, \boldsymbol{e}[N|k] \in \Omega_{\boldsymbol{e}}\}. \tag{8}$$

Then, the optimal control input is derived by solving the following optimization problem

$$\mathscr{P}(\boldsymbol{e}[k], \Omega_{\boldsymbol{e}}) : \mathbf{V}^*(\boldsymbol{e}[k], \Omega_{\boldsymbol{e}}) \doteq \min_{\boldsymbol{U}_k} \{\mathbf{V}(\boldsymbol{e}[k], \Omega_{\boldsymbol{e}}; \boldsymbol{U}_k) | \boldsymbol{U}_k \in \mathscr{U}_N(k)\}$$

where $\mathbf{V}^*(\boldsymbol{e}[k], \Omega_{\boldsymbol{e}})$ denotes the optimal cost at k^{th} instant.

Remark 1: $\Omega_{\boldsymbol{e}}$ is approximated as $\Omega_{\boldsymbol{e}} = \boldsymbol{e}_s \oplus \tilde{\Omega}_{\boldsymbol{e}}$, where $\tilde{\Omega}_{\boldsymbol{e}}$ is an arbitrarily small λ-CIS around $\boldsymbol{e}_s$, approximately occupying a region of radius $0 < \rho < 1$ in $\mathcal{E}$, such that $\tilde{\Omega}_{\boldsymbol{e}} \subseteq \Theta_1(\tilde{\Omega}_{\boldsymbol{e}}, \Phi(\tilde{\Omega}_{\boldsymbol{e}}))^{\circ}$ [20]. Precisely, $\Omega_{\boldsymbol{e}}$ can be obtained by substituting $||\boldsymbol{e}_s|| = 0$ for the leaders and $||\boldsymbol{e}_s|| = r$ for the followers.

B. Finite-Time Convergence Analysis

We now present a theorem that derives a closed-form expression of upper bound on time required by robots to segregate. In addition to offering a formal convergence guarantee, it also serves as a metric for performance evaluation later.

Theorem 1: Consider a multi-robot system with error dynamics (1) and initial error in the N-step controllable set $(\boldsymbol{e}[0] \in \Theta_N(\Omega_{\boldsymbol{e}}, \mathcal{U}))$. Let $\mathbf{V}^*(\mathbf{e}[0], \Omega_{\boldsymbol{e}})$ be the optimal cost at $\mathbf{e}[0]$, and d_m be the minimum distance between the boundary of

$\Theta_1(\Omega_{\boldsymbol{e}}, \Phi(\Omega_{\boldsymbol{e}}))$ and $\Omega_{\boldsymbol{e}}$,

$$d_m = \min_{\substack{\boldsymbol{e}_b \in \partial(\Theta_1(\Omega_{\boldsymbol{e}}, \Phi(\Omega_{\boldsymbol{e}}))) \\ \boldsymbol{e}_y \in \Omega_{\boldsymbol{e}}}} ||\boldsymbol{e}_b - \boldsymbol{e}_y||. \tag{9}$$

Then, using the proposed cost function for Finite-time MPC (7), the upper bound on the time steps required by error for each robot to converge to $\Omega_{\boldsymbol{e}}$ is given by $\frac{\mathbf{V}^*(\mathbf{e}[0], \Omega_{\boldsymbol{e}})}{d_m^2} + 1$.

Proof: Let the error at $(k+1)^{th}$ instant be $\boldsymbol{e}[k+1] = \boldsymbol{e}[0|k+1] \in \Theta_N(\Omega_{\boldsymbol{e}}, \mathcal{U})$ and the feasible cost at $(k+1)^{th}$ instant $\mathbf{V}(\boldsymbol{e}[k+1], \Omega_{\boldsymbol{e}}; \boldsymbol{U}_{k+1})$ be obtained using (7). Then, one feasible solution to the optimization problem at $(k+1)^{th}$ instant is $\boldsymbol{U}_{k+1} = \left[u_{\boldsymbol{e}}^*[\mathbf{1}|\mathbf{k}],\ \boldsymbol{u}_{\boldsymbol{e}}^*[\mathbf{2};\mathbf{k}],\ \ldots,\ \boldsymbol{u}_{\boldsymbol{e}}^*[\mathbf{N}-\mathbf{1}|\mathbf{k}],\ \boldsymbol{u}_{\boldsymbol{e}}[\mathbf{N}-\mathbf{1}|\mathbf{k}+\mathbf{1}]\right]^\intercal$, $\boldsymbol{E}_{k+1} = [\boldsymbol{e}^*[2|k],\ \boldsymbol{e}^*[3;k],\ \ldots, \boldsymbol{e}^*[N|k], \boldsymbol{e}[N|k+1]]^\intercal$. Then, $\mathbf{V}(\boldsymbol{e}[k+1], \Omega_{\boldsymbol{e}}; \boldsymbol{U}_{k+1}) = \mathbf{V}^*(\mathbf{e}[k], \Omega_{\boldsymbol{e}}) - ((\boldsymbol{e}_{kd}[1] - \boldsymbol{e}^*[1|k])^\intercal \boldsymbol{Q}(\boldsymbol{e}_{kd}[1] - \boldsymbol{e}^*[1|k]) + (\boldsymbol{u}_{kd}[0] - \boldsymbol{u}_{\boldsymbol{e}}^*[0|k])^\intercal \boldsymbol{R}(\boldsymbol{u}_{kd}[0] - \boldsymbol{u}_{\boldsymbol{e}}^*[0|k])) + ((\boldsymbol{e}_{kd}[N] - \boldsymbol{e}[N|k+1])^\intercal \boldsymbol{Q}(\boldsymbol{e}_{kd}[N] - \boldsymbol{e}[N|k+1]) + (\boldsymbol{u}_{kd}[N-1] - \boldsymbol{u}_{\boldsymbol{e}}[N-1|k+1])^\intercal \boldsymbol{R}(\boldsymbol{u}_{kd}[N-1] - \boldsymbol{u}_{\boldsymbol{e}}[N-1|k+1]))$.

Due to the terminal constraint in (8) and $\Omega_{\boldsymbol{e}}$ being a λ-CIS, $\boldsymbol{u}_{\boldsymbol{e}}[N-1|k+1] \in \Phi(\Omega_{\boldsymbol{e}})$, and $\boldsymbol{e}[N|k+1] \doteq (\boldsymbol{A}\boldsymbol{e}^*[N|k] + \boldsymbol{B}\boldsymbol{u}_{\boldsymbol{e}}[N-1|k+1]) \in \Omega_{\boldsymbol{e}}$, yielding $(\boldsymbol{e}_{kd}[N] - \boldsymbol{e}[N|k+1])^\intercal \boldsymbol{Q}(\boldsymbol{e}_{kd}[N] - \boldsymbol{e}[N|k+1]) = 0$ and $(\boldsymbol{u}_{kd}[N-1] - \boldsymbol{u}_{\boldsymbol{e}}[N-1|k+1])^\intercal \boldsymbol{R}(\boldsymbol{u}_{kd}[N-1] - \boldsymbol{u}_{\boldsymbol{e}}[N-1|k+1]) = 0$, using (5) and (6), respectively. Also, since $\boldsymbol{R} > 0$, $(\boldsymbol{u}_{kd}[0] - \boldsymbol{u}_{\boldsymbol{e}}^*[0|k])^\intercal \boldsymbol{R}(\boldsymbol{u}_{kd}[0] - \boldsymbol{u}_{\boldsymbol{e}}^*[0|k]) \geq 0$. Thus, by the principle of optimality [5], the decrement in cost can be expressed as $\mathbf{V}^*(\mathbf{e}[k+1], \Omega_{\boldsymbol{e}}) - \mathbf{V}^*(\mathbf{e}[k], \Omega_{\boldsymbol{e}}) \leq -((\boldsymbol{e}_{kd}[1] - \boldsymbol{e}^*[1|k])^\intercal \boldsymbol{Q}(\boldsymbol{e}_{kd}[1] - \boldsymbol{e}^*[1|k]))$.

Next, assuming $\boldsymbol{e}[0] \in \Theta_N(\Omega_{\boldsymbol{e}}, \mathcal{U}) \backslash \Theta_1(\Omega_{\boldsymbol{e}}, \Phi(\Omega_{\boldsymbol{e}}))$, let $\tau \in \mathbb{N}$ be a time-step such that for some $k \leq \tau$, $\boldsymbol{e}[1|k] \in \Omega_{\boldsymbol{e}}$, and hence, for $k \leq (\tau - 1)$, $\boldsymbol{e}[1|k] \in \Theta_1(\Omega_{\boldsymbol{e}}, \Phi(\Omega_{\boldsymbol{e}}))$. Then, using the definition of d_m from (9), $\forall k \leq (\tau - 2)$, $0 < d_m^2 \leq ((\boldsymbol{e}_{kd}[1] - \boldsymbol{e}[1|k])^\intercal \boldsymbol{Q}(\boldsymbol{e}_{kd}[1] - \boldsymbol{e}[1|k]))$, implying the cost decrement $\mathbf{V}^*(\mathbf{e}[k+1], \Omega_{\boldsymbol{e}}) - \mathbf{V}^*(\mathbf{e}[k], \Omega_{\boldsymbol{e}}) \leq -d_m^2$. Here, $\boldsymbol{Q} \geq 0$ is chosen to have sufficiently high eigenvalues to ensure the validity of this decrement while appropriately penalizing the deviation of error from desired. Next, summing the decrement from $k = 0$ to $(\tau - 1)$ for the worst-case scenario that $\boldsymbol{e}[1|\tau - 1] \in \partial(\Theta_1(\Omega_{\boldsymbol{e}}, \Phi(\Omega_{\boldsymbol{e}})))$, one obtains $\mathbf{V}^*(\mathbf{e}[\tau], \Omega_{\boldsymbol{e}}) - \mathbf{V}^*(\mathbf{e}[0], \Omega_{\boldsymbol{e}}) \leq -\tau d_m^2$. Since $\boldsymbol{e}^*[1|\tau] \in \Omega_{\boldsymbol{e}}$, it can be inferred from (5) and (6) that $\mathbf{V}^*(\mathbf{e}[\tau], \Omega_{\boldsymbol{e}}) = 0$. Then, we obtain $\tau \leq \frac{\mathbf{V}^*(\mathbf{e}[0], \Omega_{\boldsymbol{e}})}{d_m^2}$. This expression signifies the upper bound on time-steps required by the predicted states $\boldsymbol{e}[1|k]$ to reach $\Omega_{\boldsymbol{e}}$. Hence, for $\mathbf{e}[\tau] \in \Omega_{\boldsymbol{e}}$, $\tau \leq \frac{\mathbf{V}^*(\mathbf{e}[0], \Omega_{\boldsymbol{e}})}{d_m^2} + 1$. ■

Remark 2: τ increases with larger $\boldsymbol{Q}, \boldsymbol{R}$ as these weights amplify the contributions of the state error and control effort terms in the cost function, respectively.

C. Collision Avoidance

We build on the on-demand collision avoidance strategy from [21] to proactively identify and avoid potential collisions among robots. Unlike [21], which imposes distance-based constraints directly, we derive a linear collision avoidance constraint by leveraging the error dynamics in (1) and the error prediction model in (3), described as follows.

Let $k_{c,i}$ denote the first predicted time step at which robot i predicts a collision with robot j. Given $\boldsymbol{e}_i[j|k]$ and $\boldsymbol{x}_{di}$, the current predicted position of the robot $\boldsymbol{x}_i[j|k] = \boldsymbol{x}_{di} - \boldsymbol{e}_i[j|k]$. Then, from (2), the constraint gets activated for robot i iff $d_{ij} = \|(\boldsymbol{x}_{di} - \boldsymbol{e}_i[k_{c,i}|k-1]) - (\boldsymbol{x}_{dj} - \boldsymbol{e}_j[k_{c,i}|k-1])\| < r_{\min}$. Let us assume that robot i detects collision using this inequality at k. Then, for collision avoidance, the predicted state at $(k_{c,i} - 1)$ step of the prediction horizon must attain a position such that $\|(\boldsymbol{x}_{di} - \boldsymbol{e}_i[k_{c,i} - 1|k]) - (\boldsymbol{x}_{dj} - \boldsymbol{e}_j[k_{c,i}|k-1])\| \geq r_{\min}$, which can be linearized using the Taylor series expansion [22] assuming $f(\boldsymbol{x}_{di} - \boldsymbol{e}_i[k_{c,i} - 1|k]) = ||(\boldsymbol{x}_{di} - \boldsymbol{e}_i[k_{c,i} - 1|k]) - (\boldsymbol{x}_{dj} - \boldsymbol{e}_j[k_{c,i}|k-1])||$, about the previous predicted position of robot i at $(k_{c,i} + k - 1)$, denoted by $(\boldsymbol{x}_{di} - \boldsymbol{e}_i[k_{c,i}|k-1])$, as follows $f(\boldsymbol{x}_{di} - \boldsymbol{e}_i[k_{c,i} - 1|k]) = f(\boldsymbol{x}_{di} - \boldsymbol{e}_i[k_{c,i}|k-1]) + ((\boldsymbol{x}_{di} - \boldsymbol{e}_i[k_{c,i} - 1|k]) - (\boldsymbol{x}_{dj} - \boldsymbol{e}_j[k_{c,i}|k-1]))^\intercal \boldsymbol{J}(\boldsymbol{x}_{di} - \boldsymbol{e}_i[k_{c,i}|k-1])$. Here, higher order terms of the expansion are neglected and $\boldsymbol{J}(\boldsymbol{x}_{di} - \boldsymbol{e}_i[k_{c,i}|k-1])$ is the corresponding Jacobian matrix. On substituting this into LHS of d_{ij} for collision avoidance and rearranging, we get $\boldsymbol{b}_{i,j}^\intercal \boldsymbol{e}_i[k_{c,i} - 1|k] \geq ((r_{\min} d_{ij} - d_{ij}^2) + \boldsymbol{b}_{i,j}^\intercal \boldsymbol{e}_i[k_{c,i}|k-1])$, where $\boldsymbol{b}_{i,j} = (\boldsymbol{x}_{di} - \boldsymbol{e}_i[k_{c,i}|k-1]) - (\boldsymbol{x}_{dj} - \boldsymbol{e}_j[k_{c,i}|k-1])$. According to the results reported in [21], further improvements in collision avoidance are achieved by imposing the constraint one time step after collision detection, i.e., on $\boldsymbol{e}_i[k_{c,i}|k]$ rather than $\boldsymbol{e}_i[k_{c,i} - 1|k]$. Subsequently, using the stacked predicted errors (4), the following constraint is obtained for collision avoidance

$$\boldsymbol{g}_{i,j}^\intercal \left(\Upsilon_i \bar{\boldsymbol{e}}_i[0|k] + \boldsymbol{\Psi}_i \boldsymbol{U}_{\boldsymbol{ki}}\right) \geq c_{ij}, \tag{10}$$

where $\boldsymbol{g}_{i,j} = \left[0_{\mathbf{2(k_{c,i}-1)\times 1}}^\intercal \quad \boldsymbol{b}_{\mathbf{i,j}}^\intercal \quad \mathbf{0}_{\mathbf{2(N-k_{c,i})\times 1}}^\intercal\right]^\intercal$ and $c_{ij} = ((r_{\min} d_{ij} - d_{ij}^2) + \boldsymbol{b}_{i,j}^\intercal \boldsymbol{e}_i[k_{c,i}|k-1])$.

Remark 3: At the occurrence of obstacles, the robot states may get redefined. As long as the relocated state $\boldsymbol{e}_i[k] \in \Theta_N(\Omega_{\boldsymbol{e}}, \mathbb{U})$, finite-time convergence to $\Omega_{\boldsymbol{e}}$ is preserved. Moreover, in the case of bounded disturbances and model uncertainties, τ can be re-derived using a robust λ-CIS [19], [20].

Remark 4: The framework can handle non-agent obstacles with known dynamics by predicting their future states. For obstacles with unknown dynamics, future positions may be estimated using measured states [12] for collision avoidance.

Next, the Quadratic Programming (QP) problem for segregating multiple robots using the proposed cost-function (7) and on-demand collision avoidance constraint (10) to guarantee finite-time convergence (Theorem 1) is given as

$$\begin{aligned} \boldsymbol{U}_{\boldsymbol{k}_i}^* \quad = \quad & \text{argmin}_{\boldsymbol{U}_{\boldsymbol{k}_i}} \quad \mathbf{V}(\boldsymbol{e}_i, \Omega_{\boldsymbol{e}}; \boldsymbol{U}_{\boldsymbol{k}_i}), \\ & \text{subjected to} \quad \boldsymbol{A}_i^{coll} \boldsymbol{U}_{\boldsymbol{k}_i} \leq \boldsymbol{b}_i^{coll}, \end{aligned} \tag{11}$$

where $\boldsymbol{A}_i^{coll} = -g_{ij}^T \Psi_i$ and $\boldsymbol{b}_i^{coll} = g_{ij}^T \Upsilon_i \bar{\boldsymbol{e}}_i[0|k] - c_{ij}$.

IV. Chernoff Bound-Based AMS

Next, we derive a threshold to enable AMS using Chernoff Inequality [23]. Firstly, we formulate an event that triggers AMS. For i^{th} robot, let the real-time deviation between measured error $\bar{\boldsymbol{e}}_i[k]$ and predicted error $\boldsymbol{e}_i[1|k-1]$ be defined as

$$\bar{\delta}_i[k] = \|\bar{\boldsymbol{e}}_i[k] - \boldsymbol{e}_i[1|k-1]\|.$$

Let the real-time trigger value $\chi(\bar{\delta}_i[k])$ be a function of $\bar{\delta}_i[k]$ (Fig. 1). Then, we update the Finite-time MPC unit as

$$\bar{\boldsymbol{e}}_i[0|k] = \begin{cases} \bar{\boldsymbol{e}}_i[k], & \text{if } \chi(\bar{\delta}_i[k]) \geq \alpha_i \\ \boldsymbol{e}_i[1|k-1], & \text{otherwise} \end{cases}, \tag{12}$$

where α_i is the Chernoff bound derived using Chernoff Inequality [23] on the probability distribution of the deviation metric $\chi(\delta_i[k])$, which is computed using the stored deviation (δ_i) data of the nominal multi-robot operation. Chernoff bound was selected because it offers tighter exponential decay on tail probabilities [24], and leverages the Moment Generating Function (MGF) [25] to accurately capture both the mean and covariance structure of deviations, which enables more reliable event detection. Therefore, Chernoff bound-based AMS distinguishes nominal system behavior from instances requiring controller action due to external perturbations, i.e., it quantifies how far the actual measurements can deviate from the predictions before being statistically significant to indicate an external perturbation. Consequently, the controller is updated with measured state only for significant deviations, whereas the controller relies on predicted states during the intervals between consecutive triggers. As a result, this approach eliminates the jitters that occur when the controller is continuously updated with measured states, ensuring smooth robot motion profiles. Additionally, AMS can reduce the communication load in bandwidth-limited settings [26] by minimizing unnecessary sensor data transmission. However, due to finite size of stored sensor data and its inherent randomness, it is impossible to avoid false triggers completely. We capture this behavior by defining a confidence level γ as

$$\gamma = \mathbb{P}[\chi(\delta_i) \geq \alpha_i], \tag{13}$$

representing the tail probability of false triggers. The significance of γ is discussed later in Remark 5. Let Ξ denote the stored deviation dataset of size $T \times n$ having $\delta_i[k]$ stacked for robot $i \in \{1, 2, \ldots, n\}$ and $\forall k \in \{1, 2, \ldots, T\}$.

Assumption 1: Ξ contains the stored deviation information corresponding to the multi-robot system operation when its states have converged to a stationary Gaussian distribution.

This assumption ensures that the mean and variance of deviation distribution are constant. Since the states of a stable system eventually converge into a stationary Gaussian distribution [27], this is not too restrictive. Next, two methods of deriving bounds for a multi-robot system are proposed.

A. Decentralized Trigger

When a separate bound is derived for each robot and they are triggered individually on detection of an event, we term the method as a decentralized trigger. Here, the deviation metric $\chi(\delta_i[k])$ is defined as the sum of deviations ($\delta_i[k]$) over a time-window w for robot i to analyze the cumulative impact of deviations as a jointly normally distributed random variable. Then, the corresponding bound α_i is computed.

Firstly, we define the sequence of deviations over w as

$$\delta_{ik} = \left[\delta_i[k-w+1], \delta_i[k-w+2], \ldots, \delta_i[k-1], \delta_i[k]\right]^\intercal. \tag{14}$$

Here, δ_{ik} is considered as a jointly normally distributed random vector with a $w \times 1$ mean vector $\mu_i = \mathbb{E}[\delta_{ik}] = \left[\mathbb{E}[\delta_i[Y+1]], \quad \mathbb{E}[\delta_i[Y+2]], \quad \ldots \quad \mathbb{E}[\delta_i[Y+w]]\right]^\intercal$ and a $w \times w$ covariance matrix Σ_i[2].

Then, we define

$$\chi(\delta_i) = \delta_{ik}^\intercal\, \delta_{ik} = \sum_{j=k-w+1}^{k} \delta_i[j]^\intercal \delta_i[j]. \tag{15}$$

For computing α_i using the Chernoff Inequality [23], we first need to derive the expression for MGF of $\chi(\delta_i)$. Here, MGF of $\chi(\delta_i)$ is given as $M_\chi(\zeta_i) = \mathbb{E}[\exp(\zeta_i \chi(\delta_i))]$ [25], which can be expressed as a $w-$fold integral over $\mathbb{R}^w$ to obtain $M_\chi(\zeta_i) = \frac{1}{\sqrt{\det(2\pi\Sigma_i)}} \int_{\mathbb{R}^w} \exp(\zeta_i \delta_{ik}^\intercal \delta_{ik} - \frac{1}{2}(\delta_{ik} - \mu_i)^\intercal {\Sigma_i}^{-1}(\delta_{ik} - \mu_i))\, \mathrm{d}\delta_{ik}$. We now simplify the expression for $M_\chi(\zeta_i)$. Let $\boldsymbol{s}_i$ be a $w \times 1$ random vector following standard normal distribution. Then, $\exists$ a transformation matrix $\boldsymbol{P}$: $\delta_{ik} = \boldsymbol{P}\boldsymbol{s}_i + \mu_i$, where $\boldsymbol{P}^T \Sigma_i^{-1} \boldsymbol{P} = \boldsymbol{I}, \boldsymbol{P}^T \boldsymbol{P} = \Lambda$ and $\boldsymbol{P}\Lambda\boldsymbol{P}^{-1} = \Sigma_i$ [25]. Here, $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_w)$ containing the eigenvalues (λ_q) of Σ_i, and $\boldsymbol{g}_i$ is a constant vector. Using these transformations, $M_\chi(\zeta_i)$ simplifies to $M_\chi(\zeta_i) = \prod_{q=1}^{w} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp\left(\zeta_q \lambda_q (s_q + g_q)^2 - \frac{1}{2} s_q^2\right) \mathrm{d}s_q$. Using the Gaussian integral formula [28], and then expressing the terms as matrix products, $M_\chi(\zeta_i)$ can be simplified to $M_\chi(\zeta_i) = \frac{\exp(\frac{1}{2}\mu_i{}^\intercal((\boldsymbol{I}-2\zeta_i\Sigma_i)^{-1}-\boldsymbol{I})\Sigma_i{}^{-1}\mu_i)}{\sqrt{\det(\boldsymbol{I}-2\zeta_i\Sigma_i)}}$. Next, this expression of MGF and the confidence level (13) are substituted into the Chernoff Inequality[3] to obtain the bound $\alpha_i = \inf_{\zeta_i \in (0, \frac{1}{2\lambda_{\max}(\Sigma_i)})} [\frac{1}{\zeta_i}(\frac{1}{2}\mu_i{}^\intercal((\boldsymbol{I} - 2\zeta_i\Sigma_i)^{-1} - \boldsymbol{I})\Sigma_i{}^{-1}\mu_i - \ln(\gamma\sqrt{\det(\boldsymbol{I} - 2\zeta_i\Sigma_i)}))]$. During actual operation, $\chi(\bar{\delta}_i)$ is computed using real-time deviations $\bar{\delta}_i$ and compared with α_i to detect an event for robot i (Fig. 1).

B. Centralized Trigger

Here, the deviation metric $\chi(\delta[k])$ is defined as the sum of deviations of all the robots over w such that a single bound α is obtained. Firstly, the sequence of deviations for n robots at k^{th} instant is defined as $\omega_k = \left[\delta_{1k}, \delta_{2k}, \ldots, \delta_{ik}, \ldots, \delta_{nk}\right]^\intercal$, where δ_{ik} is as (14). Here, ω_k is considered as a jointly normally distributed random vector with $wn \times 1$ mean vector $\mu = \mathbb{E}[\delta_{ik}] = \left[\mathbb{E}[\delta_{1k}], \quad \mathbb{E}[\delta_{2k}], \ldots \mathbb{E}[\delta_{nk}]\right]^\intercal$ and $wn \times wn$ covariance matrix Σ[2]. Then, $\chi(\delta)$ is defined as $\chi(\delta) = \omega_k^\intercal \omega_k = \sum_{i=1}^{n} \delta_{ik}^\intercal \delta_{ik} = \sum_{i=1}^{n} (\sum_{j=k-w+1}^{k} \delta_i[j]^\intercal \delta_i[j])$. The subsequent derivation of α is similar to Section IV-A by first deriving the MGF for $\chi(\delta)$ and then simplifying its expression using the Gaussian integral formula [28]. Next, substituting (13) and MGF of $\chi(\delta)$ into Chernoff Inequality[3], the bound $\alpha = \inf_{\zeta \in (0, \frac{1}{2\lambda_{\max}(\Sigma)})} [\frac{1}{\zeta}(\frac{1}{2}\mu^\intercal((\boldsymbol{I} - 2\zeta\Sigma)^{-1} - \boldsymbol{I})\Sigma^{-1}\mu -$

[2] $\Sigma_i = \begin{bmatrix} \mathrm{Var}(\delta_i[Y+1]) & \mathrm{Cov}(\delta_i[Y+1], \delta_i[Y+2]) & \cdots & \mathrm{Cov}(\delta_i[Y+1], \delta_i[Y+w]) \\ \mathrm{Cov}(\delta_i[Y+2], \delta_i[Y+1]) & \mathrm{Var}(\delta_i[Y+2]) & \cdots & \mathrm{Cov}(\delta_i[Y+2], \delta_i[Y+w]) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(\delta_i[Y+w], \delta_i[Y+1]) & \mathrm{Cov}(\delta_i[Y+w], \delta_i[Y+2]) & \cdots & \mathrm{Var}(\delta_i[Y+w]) \end{bmatrix}$, where $Y = k - w$, and $\Sigma = \begin{bmatrix} \mathrm{Var}(\delta_{1k}) & \mathrm{Cov}(\delta_{1k}, \delta_{2k}) & \cdots & \mathrm{Cov}(\delta_{1k}, \delta_{nk}) \\ \mathrm{Cov}(\delta_{2k}, \delta_{1k}) & \mathrm{Var}(\delta_{2k}) & \cdots & \mathrm{Cov}(\delta_{2k}, \delta_{nk}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(\delta_{nk}, \delta_{1k}) & \mathrm{Cov}(\delta_{nk}, \delta_{2k}) & \cdots & \mathrm{Var}(\delta_i[k]) \end{bmatrix}$.

[3] Whenever MGF $M_X(\zeta)$ exists for X, Chernoff bound α is obtained using the Chernoff Inequality $\mathbb{P}[X \leq \alpha] \leq \inf_{\zeta < 0} \frac{M_X(\zeta)}{e^{\zeta\alpha}}$ [23].

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

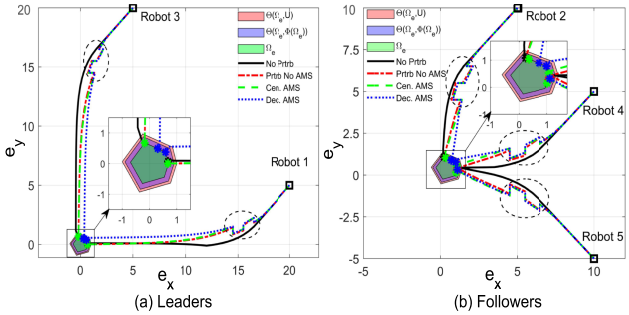

Fig. 2. Error plot depicting convergence.

TABLE II
CONVERGENCE TIME STEPS WHEN WEIGHT MATRICES WERE EMPIRICALLY TUNED TO (1) SIMULATIONS $\boldsymbol{Q}_1, \boldsymbol{Q}_3, \boldsymbol{Q}_5 = 15\boldsymbol{I}$, $\boldsymbol{Q}_2, \boldsymbol{Q}_4 = 25\boldsymbol{I}$, AND $\boldsymbol{R}_1 = \boldsymbol{R}_2 = \boldsymbol{R}_3 = \boldsymbol{R}_4 = \boldsymbol{R}_5 = 10\boldsymbol{I}$, (2) EXPERIMENTS $\boldsymbol{Q}_1 = 35\boldsymbol{I}$, $\boldsymbol{Q}_2 = 20\boldsymbol{I}$, $\boldsymbol{Q}_3 = 35\boldsymbol{I}$, $\boldsymbol{Q}_4 = 20\boldsymbol{I}$, $\boldsymbol{Q}_5 = 35\boldsymbol{I}$, AND $\boldsymbol{R}_1 = 8\boldsymbol{I}$, $\boldsymbol{R}_2 = 5\boldsymbol{I}$, $\boldsymbol{R}_3 = 8\boldsymbol{I}$, $\boldsymbol{R}_4 = 5\boldsymbol{I}$, $\boldsymbol{R}_5 = 5\boldsymbol{I}$, WHERE OBST.: OBSTACLES, PERTB.: PERTURBATIONS

	Multi-robot Segregation	Leaders		Followers	
		$\Theta_1(\Omega_e, \Phi(\Omega_e))$	Ω_e	$\Theta_1(\Omega_e, \Phi(\Omega_e))$	Ω_e
Simulation	No Obst., No Prtb.	27	28	28	29
	Obst., Prtb., No AMS	30	31	32	33
Experiment	Obst., Prtb., No AMS	15	16	16	17
	Obst., Prtb. and AMS	12	13	13	14

$\ln(\gamma\sqrt{\det(\boldsymbol{I} - 2\zeta\Sigma)}))]$. While actual operation, $\chi(\bar{\delta})$ is computed using real-time deviations $\bar{\delta}$ and compared with α to trigger all the robots simultaneously.

Remark 5: $\gamma \in (0, 1)$ is the tail probability that quantifies outlier likelihood when computing α. Higher γ implies higher chance of false triggers, and its selection, typically $\gamma \in [0.01, 0.001]$ [29], is at designer discretion.

V. RESULTS AND DISCUSSIONS

A. Simulation Results

A multi-robot system having 5 robots is to be segregated into 2 groups of 3 and 2 robots each. Group 1 has the leader Robot 1 and follower Robot 2, and Group 2 has leader Robot 3 and follower Robots 4, 5. For error dynamics shown in (1), we consider $\mathbf{A} = \mathbf{I}$, $\mathbf{B} = h\mathbf{I}$, where sampling time $h = 0.2$ sec. The initial errors of Robots 1, 2, 3, 4 and 5 are (20,5), (5,10), (5,20), (10,5), and $(10, -5)$, respectively. The desired positions of Leaders 1 and 3 are (20,10) and (10,20), respectively. Control input $\boldsymbol{u_e} \in \mathbb{R}^2 : ||\boldsymbol{u_e}||_\infty \leq 2.5$, prediction horizon $N = 10$, $r_{\min} = 0.4$ for collision avoidance and $r = 0.4$ is to be maintained between leader-follower. $\Omega_{\boldsymbol{e}}$ is then obtained using $||\boldsymbol{e}_s|| = 0$ for the leaders and $||\boldsymbol{e}_s|| = r$ for the followers, with $\lambda = 0.65$. Then, (11) is solved using CasADi [30] in MATLAB. We first validate finite-time convergence by disabling on-demand collision avoidance and with no perturbations. The resulting robot trajectories are shown by black solid lines in Fig. 2(a) and (b). Table II presents the maximum convergence time steps for leaders and followers, categorized by robot type. The following two scenarios were then simulated.

Scenario 1: Segregation in the presence of obstacles and perturbations: Collision avoidance is enabled, and robots are subjected to perturbations $\boldsymbol{p} \in \mathbb{R}^2 : ||\boldsymbol{p}||_\infty \leq 0.6$ at $k = 8, 10, 12$. Obstacles and perturbations comply with Remark 3. Robot trajectories for this scenario are shown as red-dash-dotted lines in Fig. 2(a) and (b), with perturbations depicted as black-dashed circles. Despite perturbations, leaders and followers successfully converge to $\Omega_{\boldsymbol{e}}$, as shown in Fig. 2(a), (b), and Table II. However, perturbations cause jittery velocity and acceleration profiles, as shown in Fig. 3.

Scenario 2: Segregation in the presence of obstacles, perturbations and Chernoff bound-based AMS: Collision avoidance is enabled with perturbations identical to Scenario 1, and the Chernoff bound-based AMS is activated. To derive the bounds, a deviation δ dataset was generated during simulations of nominal multi-robot operation under random measurement noise, starting from varied initial positions without perturbations, and with $\gamma = 0.01$ and window length $w = 3$. The decentralized trigger bounds for 5 robots are $\alpha_1 = 1.1405$, $\alpha_2 = 1.2277$, $\alpha_3 = 1.3173$, $\alpha_4 = 1.4932$, and $\alpha_5 = 0.7834$, whereas $\alpha = 2.7395$ for centralized trigger. Fig. 2(a) and (b) illustrate robot trajectories for this scenario with blue-dotted and green-dashed lines for decentralized and centralized triggering, respectively, where robots converge to their $\Omega_{\boldsymbol{e}}$. To test scalability, additional simulations with larger number of robots were performed, which yielded consistent results.

Effect of Chernoff bound-based AMS: Under identical perturbations, the effect of Chernoff bound-based AMS can be observed from Fig. 3(a). The robots exhibit the smoothest velocity and acceleration profiles for AMS via decentralized triggering, followed by centralized triggering, and the most jittery in the absence of AMS. Furthermore, the triggering behavior of AMS can be observed from Fig. 4(a) and (b), respectively. Here, $\chi(\bar{\delta}) < \alpha$ (green region) generates no trigger and $\chi(\bar{\delta}) \geq \alpha$ (red region) generates triggers. 2 triggers are generated for AMS via centralized triggering due to perturbations driving $\chi(\bar{\delta}) > \alpha$, as shown Fig. 4(a). Despite these triggers, the input and acceleration profiles are much smoother than the no AMS case, as depicted in Fig. 3(a)(i) and (a)(ii). Furthermore, it can be observed from Fig. 4(a) that the perturbation at $k = 8$ is not high enough to drive $\chi(\bar{\delta}) \geq \alpha$. However, the cumulative effect of perturbations at $k = 8, 10, 12$ finally drives $\chi(\bar{\delta}) > \alpha$, leading to triggers. Similarly, 1 trigger is generated for AMS via decentralized triggering in robot 5, as shown in Fig. 4(b)(v), where the perturbations drive $\chi(\bar{\delta}_5) > \alpha_5$, and there are no triggers for the other robots (Fig. 4(b)(i)–(iv)).

B. Experimental Results

The proposed framework was verified experimentally by segregating a multi-robot system of 5 TurtleBot3 Waffle Pi robots into 2 groups of 3 and 2 robots. An 8-camera Vicon Motion Capture system was used for tracking robot positions, which were fed back to a central computing unit to calculate the control inputs using the proposed framework. In the experiment arena of size 3m × 3 m, the initial positions of the 5 robots were (0.63,1.15), $(-0.22, 1.60)$, $(-0.71, -0.98)$, $(0.40, -1.11)$, and $(-1.01, 1.46)$, respectively. Group configuration is identical to Section V-A, where desired positions of leader Robots 1 and 3 are $(1.70, -1.20)$ and $(-1.30, 0.70)$, respectively. Control input $\boldsymbol{u_e} \in \mathbb{R}^2 : ||\boldsymbol{u_e}||_\infty \leq 0.22$ m/sec due to hardware constraints of the robot, $N = 10$, $r_{\min} = 0.3$m and $r = 0.3$ m. Figs. 5 and 6 depict the experimental results, where perturbations $\boldsymbol{p} \in \mathbb{R}^2 : ||\boldsymbol{p}||_\infty \leq 0.5$ were introduced in Robots 2 and 4 as random displacement from their current position and orientation at random instances.

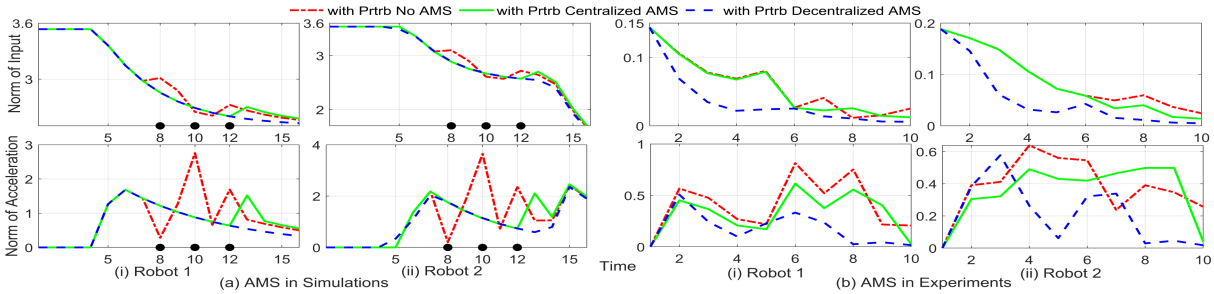

Fig. 3. Input (velocity) and acceleration profiles of Robot 1 (leader) and 2 (follower) in simulations and experiments.

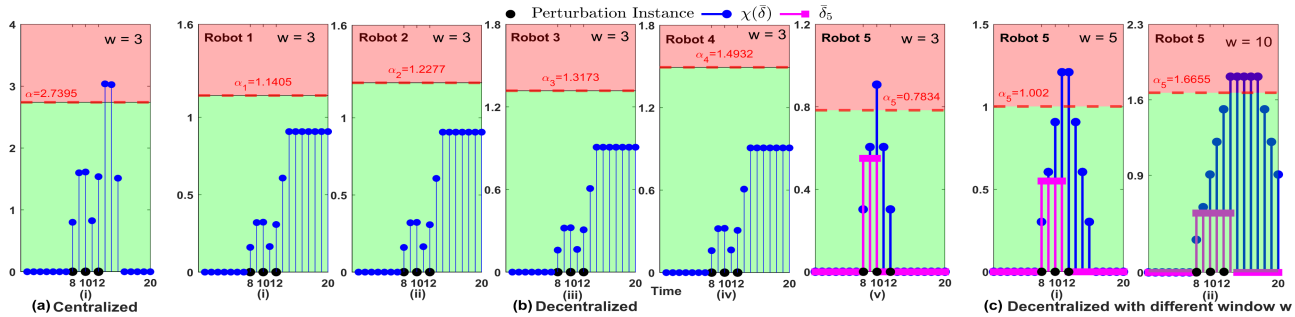

Fig. 4. $\chi(\bar{\delta})$, perturbation instances (black dots) and triggers for AMS via Centralized (a) and Decentralized triggering (b)–(c).

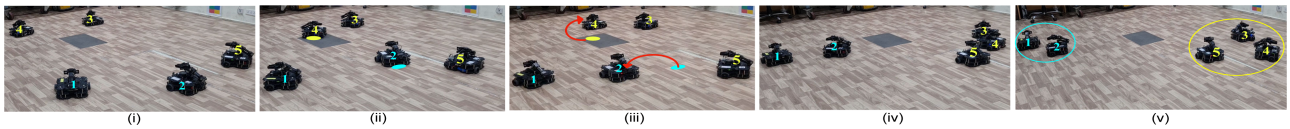

Fig. 5. Snapshots of segregation experiment. (iii) Robots 2 and 4 are perturbed, as shown using red arrow.

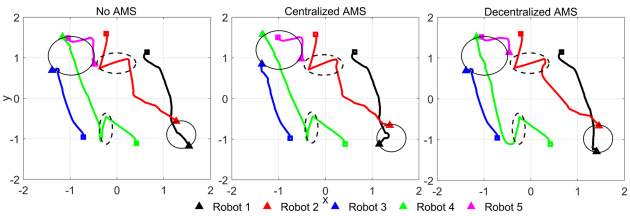

Fig. 6. Robot trajectories (Vicon data) during experiments. Dashed circles: perturbations, solid circles: segregated groups.

TABLE III
VARIATION OF CHERNOFF BOUND α WITH γ AND w

γ	α (for $w = 3$)	α (for $w = 6$)
0.0001	3.6531	5.5787
0.001	3.1310	4.8639
0.01	2.7395	4.0963

We attempted to introduce identical perturbations to clearly distinguish between AMS via centralized and decentralized triggering. The Chernoff bounds obtained in Scenario 2, Section V-A were used in the experiments too and collision avoidance was always enabled. Fig. 3(b)(i) and (b)(ii) illustrate the input and acceleration profiles of the robots obtained during experiments. Similar to simulations, the profiles are the smoothest for AMS via decentralized triggering, followed by centralized triggering, and the most jittery in the absence of AMS.

C. Discussions and Comparison With Existing Works

Fig. 4(b)(v) and (c) illustrate the behavior of the Chernoff bound-based AMS method via decentralized triggering for robot 5 in terms of $\bar{\delta}$ and $\chi(\bar{\delta})$ when $w = 3, 5, 10$, under identical perturbation $\boldsymbol{p} \in \mathbb{R}^2 : ||\boldsymbol{p}|| = 0.55$ at $k = 8$. Though the bounds for $w = 5, 10$ are higher than $w = 3$, a perturbation high enough to drive $\chi(\bar{\delta}_5) \geq \alpha_5$ generates more triggers in $w = 5, 10$, as shown in Fig. 4(c), because the effect of 1 perturbation persists for a duration proportional to w. Similarly, multiple triggers are generated for $w = 5, 10$ (Fig. 4(c)) when only 1 trigger is generated for $w = 3$ (Fig. 4(b)(v)).

Also, Table III depicts the variation in α with γ and w for centralized trigger. Given w, lower γ leads to higher α; however, given γ, higher w leads to higher α. Therefore, if α is obtained using the nominal system data with appropriate γ and w, communication between sensor and controller can be reduced by minimizing total triggers.

Table IV compares the proposed framework with existing works, namely the heuristic method [9] and mean-based Hoeffding method [31] under identical conditions and perturbations for Robot 5. The heuristic method exhibits longer delays in detection (D_t) due to lack of window w, and therefore, leads to

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

TABLE IV
PERFORMANCE COMPARISON WITH EXISTING WORKS

Method	Bound	N_T	$\boldsymbol{v}_{\text{rms}}$	$\boldsymbol{a}_{\text{rms}}$	D_t	N_c
Heuristic [9]	3	2	2.8753	3.1159	6	3
Hoeffding bound [31]	2.1833	2	2.5186	2.8046	3	2
Cen. Chernoff bound (Sec. IV-B)	2.7395	2	2.1715	2.1184	1	2
Dec. Chernoff bound (Sec. IV-A)	0.7834	1	1.2628	1.4664	1	2

N_T: No. of Triggers, $\boldsymbol{v}_{\text{rms}}$: rms velocity, $\boldsymbol{a}_{\text{rms}}$: rms acceleration

more collision-prone paths (N_c). A similar yet slightly improved behavior is observed for Hoeffding method. In contrast, the proposed Chernoff bound method, particularly the decentralized trigger, exhibits lower detection delay D_t by using w, and by leveraging MGF to obtain tighter bounds. This enables faster and more reliable detection without the downsampling limitations of Hoeffding method, and therefore, yields lower $\boldsymbol{v}_{\text{rms}}$ and $\boldsymbol{a}_{\text{rms}}$.

VI. CONCLUSION

This work presented a Finite-Time MPC framework that uses invariant sets to achieve finite-time convergence of a multi-robot system having leader-follower error dynamics into segregated formation with on-demand collision avoidance. An upper bound on the convergence time required by the robots is also derived. Additionally, a Chernoff bound-based Asynchronous Motion Smoothing method via Centralized and Decentralized triggering is also proposed to enable smoother robot motion profiles, offering a statistically-grounded approach for deriving bounds in multi-robot systems. The framework is validated through extensive simulations and hardware experiments, along with a comparison with existing works. Future work will consider higher-order robot models with uncertainties, and collision avoidance with obstacles having unknown dynamics. Moreover, learning the optimal window length for Chernoff bound calculation, and derivation of tighter upper bound while accounting for robot relocations constitute prominent future research directions.

REFERENCES

[1] M. Pan et al., "Physical interactions segregate robot swarms," *IEEE Robot. Autom. Lett.*, vol. 9, no. 8, pp. 7333–7340, Aug. 2024.

[2] V. G. Santos, L. C. A. Pimenta, and L. Chaimowicz, "Segregation of multiple heterogeneous units in a robotic swarm," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 1112–1117.

[3] M. Kumar, D. P. Garg, and V. Kumar, "Segregation of heterogeneous units in a swarm of robotic agents," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 743–748, Mar. 2010.

[4] F. R. Inácio, D. G. Macharet, and L. Chaimowicz, "PSO-based strategy for the segregation of heterogeneous robotic swarms," *J. Comput. Sci.*, vol. 31, pp. 86–94, 2019.

[5] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[6] R. Tallamraju, D. H. Salunkhe, S. Rajappa, A. Ahmad, K. Karlapalem, and S. V. Shah, "Motion planning for multi-mobile-manipulator payload transport systems," in *Proc. IEEE Int. Conf. Automat. Sci. Eng.*, 2019, pp. 1469–1474.

[7] C. Jiang, "Distributed sampling-based model predictive control via belief propagation for multi-robot formation navigation," *IEEE Robot. Autom. Lett.*, vol. 9, no. 4, pp. 3467–3474, Apr. 2024.

[8] C. E. Luis, M. Vukosavljev, and A. P. Schoellig, "Online trajectory generation with distributed model predictive control for multi-robot motion planning," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 604–611, Apr. 2020.

[9] S. Gupta, S. Chaudhary, D. Maurya, S. K. Joshi, N. S. Tripathy, and S. V. Shah, "Segregation of multiple robots using model predictive control with asynchronous path smoothing," in *Proc. IEEE Conf. Control Technol. Appl.*, 2022, pp. 1378–1383.

[10] A. Eqtami, S. H. Alamdari, D. V. Dimarogonas, and K. J. Kyriakopoulos, "A self-triggered model predictive control framework for the cooperation of distributed nonholonomic agents," in *Proc. IEEE Conf. Decis. Control*, 2013, pp. 7384–7389.

[11] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Event-triggered intermittent sampling for nonlinear model predictive control," *Automatica*, vol. 81, pp. 148–155, 2017.

[12] B. Guo, N. Guo, and Z. Cen, "Obstacle avoidance with dynamic avoidance risk region for mobile robots in dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 5850–5857, Jul. 2022.

[13] P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings, "Suboptimal model predictive control (feasibility implies stability)," *IEEE Trans. Autom. Control*, vol. 44, no. 3, pp. 648–654, Mar. 1999.

[14] P. O. M. Scokaert and D. Q. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Trans. Autom. Control*, vol. 43, no. 8, pp. 1136–1142, Aug. 1998.

[15] A. Anderson, A. H. González, A. Ferramosca, and E. Kofman, "Finite-time convergence results in model predictive control," in *Proc. Eur. Control Conf.*, 2018, pp. 3203–3208.

[16] D. Shah and L. Vachhani, "Swarm aggregation without communication and global positioning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 886–893, Apr. 2019.

[17] M. H. Romanycia and F. J. Pelletier, "What is a heuristic?," *Comput. Intell.*, vol. 1, no. 1, pp. 47–58, 1985.

[18] H. Schlüter, F. Solowjow, and S. Trimpe, "Event-triggered learning for linear quadratic control," *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4485–4498, Oct. 2021.

[19] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*, vol. 78. Berlin, Germany: Springer, 2008.

[20] A. Anderson, A. H. González, A. Ferramosca, and E. Kofman, "Finite-time convergence results in robust model predictive control," *Optim. Control Appl. Methods*, vol. 39, no. 5, pp. 1627–1637, 2018.

[21] C. E. Luis and A. P. Schoellig, "Trajectory generation for multiagent point-to-point transitions via distributed model predictive control," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 375–382, Apr. 2019.

[22] C. H. Edwards, *Advanced Calculus of Several Variables*. North Chelmsford, MA, USA: Courier Corporation, 2012.

[23] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," *Ann. Math. Stat.*, vol. 23, pp. 493–507, 1952.

[24] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[25] A. Gut, "Probability: A graduate course," in *Statistics*. Berlin, Germany: Springer, 2005.

[26] N. S. Tripathy, I. N. Kar, and K. Paul, "Stabilization of uncertain discrete-time linear system with limited communication," *IEEE Trans. Autom. Control*, vol. 62, no. 9, pp. 4727–4733, Sep. 2017.

[27] P. R. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control*. Philadelphia, PA, USA: SIAM, 2015.

[28] A. Jeffrey and H. H. Dai, *Handbook of Mathematical Formulas and Integrals*. Amsterdam, The Netherlands: Elsevier, 2008.

[29] M. G. Natrella, *Experimental Statistics*, vol. 91. North Chelmsford, MA, USA: Courier Corporation, 2005.

[30] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi–A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, 2019.

[31] K. Gatsis and G. J. Pappas, "Sample complexity of networked control systems over unknown channels," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 6067–6072.