Invariant Set Distributed Explicit Reference Governors for Provably Safe On-Board Control of Nano-Quadrotor Swarms

Bryan Convens[1,2]*, *Kelly Merckaert*[1,3], *Bram Vanderborght*[1,2] and *Marco M. Nicotra*[4]

[1]Robotics and Multibody Mechanics (R&MM), Department of Mechanical Engineering, Vrije Universiteit Brussel, Brussels, Belgium, [2]Imec, Leuven, Belgium, [3]Flanders Make, Leuven, Belgium, [4]Robotics, Optimization, and Constrained Control (ROCC), Department of Electrical, Computer, and Energy Engineering, University of Colorado Boulder, Boulder, CO, United States

This article provides a theory for provably safe and computationally efficient distributed constrained control, and describes an application to a swarm of nano-quadrotors with limited on-board hardware and subject to multiple state and input constraints. We provide a formal extension of the explicit reference governor framework to address the case of distributed systems. The efficacy, robustness, and scalability of the proposed theory is demonstrated by an extensive experimental validation campaign and a comparative simulation study on single and multiple nano-quadrotors. The control strategy is implemented in real-time on-board palm-sized unmanned erial vehicles, and achieves safe swarm coordination without relying on any offline trajectory computations.

Keywords: aerial robotics control, multi-robot systems, actuator saturation, distributed collision avoidance, guaranteed safety, human-swarm interaction, invariant set control, nano-quadrotor swarm

OPEN ACCESS

Edited by:
Holger Voos,
University of Luxembourg,
Luxembourg

Reviewed by:
Ning Sun,
Nankai University, China
Navid Razmjooy,
Independent researcher, Ghent,
Belgium

****Correspondence:***
Bryan Convens
bryan.convens@vub.be

Specialty section:
This article was submitted to
Robotic Control Systems,
a section of the journal
Frontiers in Robotics and AI

Received: *03 February 2021*
Accepted: *21 April 2021*
Published: *22 June 2021*

Citation:
Convens B, Merckaert K,
Vanderborght B and Nicotra MM
(2021) Invariant Set Distributed Explicit
Reference Governors for Provably Safe
On-Board Control of Nano-
Quadrotor Swarms.
Front. Robot. AI 8:663809.
doi: 10.3389/frobt.2021.663809

1 INTRODUCTION

Swarms of aerial robots or Unmanned Aerial Vehicles (UAVs) are emerging as a disruptive technology that enables highly re-configurable, on-demand, distributed intelligent autonomous systems with high impact on many areas of science, technology, and society (Chung et al., 2018).

These swarms can be employed to solve real-world tasks where the environment is to be explored (Marconi et al., 2012; Bayram et al., 2017), and to be traversed or exploited (Vásárhelyi et al., 2018) with a prescribed goal state or a desired formation. To operate effectively in uncertain real-world environments, each agent in the swarm must be capable of safely navigating to its target along a-priori unknown paths. Not only does each robot need to respect its operational constraints (e.g. actuator saturation, speed limits, allowed flight zones), it must also avoid collisions with environmental hazards and other agents (Franchi et al., 2012; Alonso-Mora et al., 2015; Franchi et al., 2016; Zhou et al., 2018) in the presence of imperfect dynamic models, measurement noise, and communication delays. Most importantly, to ensure a high level of safety and robustness, the robots should use their on-board computational resources rather than relying on off-board resources (e.g. a ground control station). The latter provide a central point of failure, and are susceptible to time delays, communication overhead, and information loss. This calls for reactive and distributed control algorithms that can be implemented in real-time on-board UAVs and only rely on local information to solve the global navigation task safely.

Achieving goal satisfaction and safety certificates for a swarm of autonomous Micro Aerial Vehicles (MAVs) presenting limited resources for on-board computation, power, communication, sensing, and actuation is considerably challenging (Chung et al., 2018). Moreover, even for large platforms with more advanced capabilities, the computational power available to implement control algorithms is typically limited in favor of running mission-dependent algorithms related to localization and sensing systems (Brockers et al., 2014). Hence, computationally efficient and provably safe on-board algorithms for multi-robot systems are of paramount importance for achieving safety-critical tasks in complex environments.

In this work, we develop a provably safe and robust constrained control methodology that is fully distributed and can be implemented on the individual agents of a swarm of Vertical Take-Off and Landing (VTOL) vehicles. The algorithm is validated using the smallest open-source available nano-quadrotor platform, i.e. Bitcraze's Crazyflie 2.1. An accompanying video can be found at https://youtu.be/le6WSeyTXNU.

2 RELATED WORK

As discussed in (Murray, 2007; Brambilla et al., 2013; Parker et al., 2016; Chamanbaz et al., 2017; Chung et al., 2018; Coppola et al., 2020), swarm robotics has become an active area of research covering a broad spectrum of topics within the robotics and control communities. The problem of safely controlling the motion of aerial robot swarms can be classified based on approaches for which the main portion of the algorithm, and especially the part that ensures safety and goal satisfaction, is running either *off-board* or *on-board* the UAVs. This classification is motivated because most existing works provide algorithmic contributions which belong to the off-board category (see **Section 2.1**), but as explained in **Section 1**, on-board navigation algorithms (see **Section 2.2**) are preferred from a safety and autonomy perspective.

Unfortunately, there does not exist one safe navigation strategy that suits all UAV applications. For each strategy there is an inherent trade-off between computational efficiency, performance, safety guarantees, simplicity, generality, and scalability to swarms. To provide a fair point of comparison, it is worth noting that VTOLs can vary significantly in terms of the available on-board computational power. For instance, a 35 g Crazyflie quadrotor carries an STM32F4 microprocessor with a clock speed of 168MHz and 192kB RAM. For comparison, larger platforms with a mass above ±700 g can use processors like the Odroid-XU4 (Liu et al., 2018) or the NVIDIA TX2 (Jung et al., 2018; Sanket et al., 2018; Ding et al., 2019; Carrio et al., 2020). The latter has a six-core CPU, each with a clock speed of 2GHz, a 256-core NVIDIA GPU, and 8 GB RAM. Since very limited battery power for computation, memory, and communication available to tiny MAVs intrinsically calls for different kinds of navigation and control strategies (Purohit et al., 2014), the literature review is mainly limited to off-board and on-board navigation strategies applied to nano-quadrotors.

2.1 Off-Board Navigation Strategies for Nano-Quadrotors

Most approaches, such as (Campos-Macías et al., 2017; Chen et al., 2017; Herbert et al., 2017; Preiss et al., 2017a; Wang et al., 2017; Fridovich-Keil et al., 2018; Honig et al., 2018; Kolaric et al., 2018; Cappo et al., 2018a; Cappo et al., 2018b; Xu and Sreenath, 2018; Bajcsy et al., 2019; Du et al., 2019; Fathian et al., 2019; Liu et al., 2019; Luis and Schoellig, 2019; Rubies-Royo et al., 2019; Vukosavljev et al., 2019), try to ensure a particular level of safety and robustness, by running the core search-based or optimization-based algorithms *off-board* the UAVs, and thus outsource the high computational cost to ground control stations that send the trajectories to the UAV's on-board position or attitude controller. Frameworks such as (Preiss et al., 2017a; Honig et al., 2018) combine graph-based planning and continuous trajectory optimization to compute safe and smooth trajectories, but take several minutes for a swarm of hundreds of quadrotors in obstacle-rich environments. In (Luis and Schoellig, 2019), a scalable distributed model predictive control algorithm with on-demand collision avoidance is proposed to perform point-to-point transitions with labeled agents. This strategy reduces the computation time to the order of seconds. (Campos-Macías et al., 2017) introduces a hybrid approach to trajectory planning, fusing sampling-based planning techniques and model-based optimization via quadratic programming (QP). For a single nano-quadrotor in obstacle-dense environments, a provably safe trajectory can be computed online every 0.1–1s, depending on the scenario. Frameworks such as (Du et al., 2019; Vukosavljev et al., 2019) are based on designing off-board libraries of safe motion primitives for a swarm of tiny MAVs, but typically require too much memory for on-board implementation. (Du et al., 2019) relies on combinatorial and nonlinear optimization techniques that are executed on a central computer, requires iterative procedures to resolve collisions between agents in a sequential manner, and does not guarantee to find a feasible solution. A modular, robust, and hierarchical framework for safe planning of robot teams is proposed in (Vukosavljev et al., 2019). Although the run-time components, executed off-board, require only a small computing time, this approach is centralized, requires a-priori known environments and is conservative due to the restriction to a discretization, i.e. a gridded workspace partitioned into rectangular boxes. Works based on the online FaSTrack motion planner (Herbert et al., 2017) provide strong safety guarantees under the assumption of a single near-hover quadrotor with a decoupled structure (Fridovich-Keil et al., 2018) or obtain weaker safety guarantees using neural network classifiers to consider control-affine dynamics (Rubies-Royo et al., 2019). Hamilton-Jacobi reachability analysis was applied to multi-agent swarms using sequential priority ordering (Bajcsy et al., 2019) or the selection of air highways (Chen et al., 2017). A centralized multi-robot system planner for enabling theatrical performance is designed in (Cappo et al., 2018a; Cappo et al., 2018b) using time-aware trajectory formulation for validation, verification, and trajectory refinement. The human intent is translated online into non-colliding and dynamically feasible trajectories for multiple nano-quadrotors. Safety barrier certificates based on exponential control barrier functions are used in (Wang et al., 2017) to ensure in a minimally invasive way collision-free maneuvers for teams of small quadrotors flying through formations and in (Xu and Sreenath, 2018) for the safe teleoperation of nano-quadrotor swarms via a remote joystick in a set of static constraints. In (Wang et al., 2017) this requires a centralized QP to be solved at 50 Hz on a ground PC to minimize the difference between the actual and nominal control. Distributed formation control approaches that have been demonstrated on small quadrotors, but are computed off-board have shown robustness to bounded measurement noise (Kolaric et al., 2018), to communication delays, nonlinearities, parametric perturbations, and external disturbances (Liu et al., 2019). Input feasibility and collision avoidance is guaranteed in (Fathian et al., 2019) for single-integrator dynamics, and is claimed to be extendable to agents with higher-order dynamics in (Fathian et al., 2018).

2.2 On-Board Navigation Strategies for Nano-Quadrotors

Only few works such as (Preiss et al., 2017b; Desaraju and Michael, 2018; McGuire et al., 2019) achieved to run computationally efficient navigation algorithms *on-board* the small embedded flight controllers of nano-quadrotors, but mostly with limited safety guarantees. These strategies typically can only handle first order dynamics, can only deal with a small set of constraints and a small number of agents, or require too much on-board memory. In (McGuire et al., 2019), a swarm gradient bug algorithm reacts to static obstacles on the fly, but collisions still occur. In (Preiss et al., 2017b), single piece polynomial planners can follow predefined paths uploaded offline for a single quadrotor, but are not suitable for dynamically changing environments. They use artificial potential fields on a swarm of these UAVs hovering in formation and show avoidance of an obstacle with a known position in a distributed fashion, but without providing theoretical safety certificates on collision avoidance or actuator saturation. A promising approach to the computationally efficient robust constrained control of nonlinear systems is proposed in (Desaraju et al., 2018) and uses an experience driven Explicit MPC (EMPC). This method was implemented in (Desaraju and Michael, 2018) and reliably ran at 100 Hz on board the tiny MAV's firmware in the presence of control input and velocity constraints. Due to the nature of EMPC, however, the introduction of collision avoidance constraints between multiple robots would make the EMPC database grow exponentially in size, thus becoming prohibitive for fast online queries.

2.3 Contributions

To the best of our knowledge, the literature does not provide any provably safe control techniques that achieve on-board real-time control of large nano-quadrotor swarms with higher-order dynamics in the presence of actuator, obstacle, and agent collision avoidance constraints.

This work is based on the Explicit Reference Governor (ERG), which is a novel framework for the closed-form feedback control of nonlinear systems subject to constraints on the state and input variables (Nicotra and Garone, 2018). This approach does not rely on online optimization and is particularly promising for control applications with fast dynamics, limited on-board computational capabilities, or strict regulations on code reliability. This article extends the centralized ERG framework (Nicotra and Garone, 2018) and a distributed ERG (D-ERG) (Nicotra et al., 2015) formulation, and encapsulates these two core contributions:

1. The ERG theory is extended to distributed multi-agent systems with fourth-order dynamics and subject to constraints on states and actuator inputs. This work supplies all theoretical details of a general and scalable D-ERG framework along with a formal proof on correctness, the formulation of different offline design strategies for computing safe threshold values of Lyapunov and invariance-based level sets. Moreover we formulated two swarm collision avoidance control policies, a decentralized and a distributed version, that require a different information exchange.
2. The effectiveness, robustness, and computational efficiency of our control and navigation layers, running on-board the Crazyflie nano-quadrotor at 500 Hz, is validated extensively in several scenarios with single or multiple quadrotors subject to state and input constraints. All proposed formulations are validated and quantitatively compared. These are the first published experimental results on the use of ERG and D-ERG on quadrotors, and (to the best of our knowledge) is the only work in the literature that achieves provably safe constrained control at such high frequencies on-board nano-quadrotors for such a broad set of state and input constraints. The D-ERG's goal satisfaction and safety certificates are put in sharp contrast with those of a Navigation Field method that suffers from instabilities and collisions when the agents posses higher-order dynamics.

The rest of this article is organized as follows. **Section 3** introduces the used notation. The problem is formulated in **Section 4**. The proposed strategy is outlined in **Section 5**, and constitutes the control layer and the navigation layer which are described in **Section 6** and in **Section 7**, respectively. The results of extensive hardware validations and a comparative simulation study with single and swarms of nano-quadrotors are presented in **Section 8**, and discussed in **Section 9**. Finally, some concluding remarks are given in **Section 10**.

3 NOTATION

In this work, all vectors are column vectors. Unit vectors are denoted using the hat symbol $\hat{\boldsymbol{a}}$. Unit vectors aligned with the axes of a right-handed Cartesian reference frame are denoted as $\hat{\boldsymbol{e}}_1$, $\hat{\boldsymbol{e}}_2$, $\hat{\boldsymbol{e}}_3$. $\mathbf{0}_{m\times n}$ and $\mathbf{1}_{m\times n}$ represent $m \times n$ matrices of zeros and ones, respectively. $\boldsymbol{I}_n$ represents an identity matrix of dimension $n \times n$. The concatenation of vectors $\boldsymbol{v}_i$ to $\boldsymbol{v}_k$ is denoted by the vector $\boldsymbol{v}_{i:k} = [\boldsymbol{v}_i^T, \ldots, \boldsymbol{v}_k^T]^T$. Given a vector in $\mathbb{R}^3$, $\|\|_{xy}$ denotes the following norm $\|\boldsymbol{v}\|_{xy} = \sqrt{v_1^2 + v_2^2}$. The hat operator $\wedge : \mathbb{R}^3 \mapsto \mathrm{SO}(3)$ denotes the skew-symmetric matrix transformation

$$\boldsymbol{v}^{\wedge} = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}, \tag{1}$$

whereas the vee operator $\vee : \mathrm{SO}(3) \mapsto \mathbb{R}^3$ denotes the vector extraction of the skew-symmetric terms

$$\boldsymbol{R}^{\vee} = \frac{1}{2}\begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix}. \tag{2}$$

4 PROBLEM FORMULATION

The system and parts of the problem are stated first. **Section 4.1** presents the dynamic model of a generic quadrotor. Nevertheless, the proposed method can be readily extended to any VTOL vehicle. The state and input constraints, which each agent should always satisfy, are defined in **Section 4.2** and illustrated in this video https://youtu.be/le6WSeyTXNU.

4.1 Dynamic Model

As depicted in **Figure 1**, each agent of the robotic swarm is modeled as a quadrotor with mass $m \in \mathbb{R}_{>0}$ and moment of inertia $\boldsymbol{J} \in \mathbb{R}^{3\times 3}_{>0}$, $\boldsymbol{J} = \boldsymbol{J}^T$ defined with respect to the body reference frame $\mathcal{B}$. Let $\boldsymbol{p} = [x, y, z]^T \in \mathbb{R}^3$ and $\dot{\boldsymbol{p}} = [\dot{x}, \dot{y}, \dot{z}]^T \in \mathbb{R}^3$ denote the position and the velocity of the body reference frame $\mathcal{B}$ with respect to the inertial reference frame $\mathcal{W}$. The attitude of each agent is represented by either the rotation matrix $\boldsymbol{R}$ or by the roll, pitch, and yaw angles $\boldsymbol{\Theta} = [\phi, \theta, \psi]^T \in \mathbb{R}^3$ that realign the axes of $\mathcal{B}$ with the axes of $\mathcal{W}$. Finally, $\boldsymbol{\omega} = \omega_x \hat{\boldsymbol{x}}_{\mathcal{B}} + \omega_y \hat{\boldsymbol{y}}_{\mathcal{B}} + \omega_z \hat{\boldsymbol{z}}_{\mathcal{B}} \in \mathbb{R}^3$ denotes the angular velocity of the vehicle expressed in the frame $\mathcal{B}$.

As detailed in (Hua et al., 2013), the dynamic model of a generic VTOL subject to a gravitational force in the $-\hat{\boldsymbol{z}}_{\mathcal{W}}$ direction, a unidirectional thrust force $T \in \mathbb{R}_{\geq 0}$ in the $\hat{\boldsymbol{z}}_{\mathcal{B}}$ direction, and a torque vector $\boldsymbol{\tau} \in \mathbb{R}^3$ about the axes of $\mathcal{B}$ is

$$\begin{cases} \text{Position dynamics :} & m\ddot{\boldsymbol{p}} = T\hat{\boldsymbol{z}}_{\mathcal{B}} + m\boldsymbol{g}, \\ \text{Attitude dynamics :} & \boldsymbol{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega}^{\wedge}\boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{\tau}, \\ & \dot{\boldsymbol{R}} = \boldsymbol{\omega}^{\wedge}\boldsymbol{R}, \end{cases} \tag{3}$$

where $\hat{\boldsymbol{z}}_{\mathcal{B}} = \boldsymbol{R}\hat{\boldsymbol{e}}_3$, $\boldsymbol{g} = -g\hat{\boldsymbol{e}}_3$, and $g \approx 9.81\ \mathrm{m/s^2}$ is the gravitational acceleration. System **(3)** possesses fourth-order dynamics and can be entirely described by the state vector

$$\boldsymbol{x} = \left[\boldsymbol{p}^T, \boldsymbol{\Theta}^T, \dot{\boldsymbol{p}}^T, \boldsymbol{\omega}^T\right]^T \in \mathbb{R}^{12} \tag{4}$$

subject to the control input vector

$$\boldsymbol{u} = \left[T, \boldsymbol{\tau}^T\right]^T \in \mathbb{R}^4. \tag{5}$$

For the specific case of a quadrotor, it is possible to rewrite the control input **(5)** as a function of the motor voltage commands $\boldsymbol{U} = [U_1, \ldots, U_4]^T \in \mathbb{R}^4$, leading to

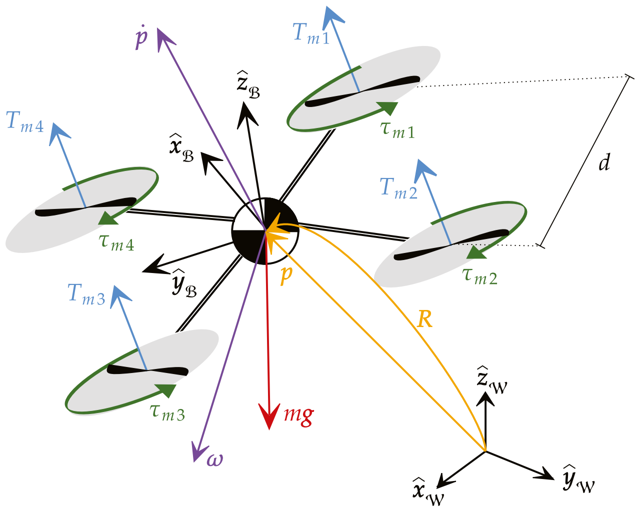


FIGURE 1 | Schematic representation of a quadrotor agent.

$$\boldsymbol{u} = \begin{bmatrix} K_T & K_T & K_T & K_T \\ -K_T\dfrac{d}{\sqrt{2}} & -K_T\dfrac{d}{\sqrt{2}} & K_T\dfrac{d}{\sqrt{2}} & K_T\dfrac{d}{\sqrt{2}} \\ -K_T\dfrac{d}{\sqrt{2}} & K_T\dfrac{d}{\sqrt{2}} & K_T\dfrac{d}{\sqrt{2}} & -K_T\dfrac{d}{\sqrt{2}} \\ -K_\tau & K_\tau & -K_\tau & K_\tau \end{bmatrix} \begin{bmatrix} U_1^2 \\ U_2^2 \\ U_3^2 \\ U_4^2 \end{bmatrix} \tag{6}$$

where d is the nominal distance between the motor axis and the center of mass of the aircraft, and $K_T, K_\tau \in \mathbb{R}_{>0}$ denote the actuator's thrust and torque constant respectively.

4.2 State and Input Constraints

To ensure safety of a swarm of N_{a} agents, every agent $i \in \{1, \ldots, N_{\mathrm{a}}\}$ is subject to the following constraints.

4.2.1 Saturation (Static Box Input Constraints)

Actuator saturation has been observed as the primary cause of instability for quadrotors in free flight. Indeed, whenever one of the motors is subject to saturation, the control law is unable to generate an arbitrary torque vector. This can lead to undesired attitude oscillations that quickly devolve into catastrophic failures. To prevent this scenario, each motor voltage U_j is required to stay within its lower and upper saturation limits,

$$U_{\min} \leq U_j \leq U_{\max}, \quad \forall j \in \{1, 2, 3, 4\}, \tag{7}$$

with $U_{\min} < U_h = \sqrt{mg/(4K_T)} < U_{\max} \in \mathbb{R}_{>0}$ and U_h defines the motor voltages required for static hovering in place.

4.2.2 Walls (Static Polytopic State Constraints)

All agents have collision radius $R_{\mathrm{a}} \in \mathbb{R}_{>0}$ and are required to operate in a confined environment defined by a convex polytope of N_{w} oriented faces (i.e. planar walls). To enforce this requirement, each agent i must satisfy the following convex constraint

$$\hat{\boldsymbol{c}}_{\mathrm{w}_j}^T \boldsymbol{p}_i \leq d_{\mathrm{w}_j} - R_{\mathrm{a}}, \quad \forall j \in \{1, \ldots, N_{\mathrm{w}}\}, \tag{8}$$

with $\hat{\boldsymbol{c}}_{\mathrm{w}_j} \in \mathbb{R}^3$ denoting the normal vector on the wall pointing in the inadmissible direction and $d_{\mathrm{w}_j} \in \mathbb{R}$ describing the shortest distance between the origin of $\mathcal{W}$ and the wall.

4.2.3 Obstacles (Static Cylindrical/Spherical State Constraints)

In addition to planar walls, all agents must also avoid collision with N_o cylindrical obstacles. To enforce this requirement, each agent i must satisfy the following non-convex constraints

$$\left\| \boldsymbol{p}_i - \boldsymbol{o}_j \right\|_{xy} \geq R_{\mathrm{o}_j} + R_{\mathrm{a}}, \quad \forall j \in \{1, \ldots, N_{\mathrm{o}}\}, \tag{9}$$

with cylinder radius $R_{\mathrm{o_j}} \in \mathbb{R}_{>0}$ and center $\boldsymbol{o}_j \in \mathbb{R}^3$. Note that the cylindrical obstacles can be replaced with spheres by replacing $\|\|_{xy}$ with the Eucledian norm.

4.2.4 Agent Collisions (Collaborative Cylindrical/Spherical State Constraints)

To prevent undesirable interactions between agents (e.g. collision, propeller downwash, sonar jamming), each pair of agents is tasked with satisfying the following dynamic cylindrical exclusion constraints

$$\left\| \boldsymbol{p}_i - \boldsymbol{p}_k \right\|_{xy} \geq 2R_{\mathrm{a}}, \quad \forall k \in \{1, \ldots, N_{\mathrm{a}}\} : k \neq i. \tag{10}$$

As per the previous case, it is trivial to replace the cylindrical constraint with a spherical constraint if vertical agent interactions are not deemed problematic.

4.3 Control Objectives

The aim of this paper is to develop a guaranteed safe distributed constrained control strategy for an homogeneous swarm of quadrotors with very limited on-board resources for computation, memory, and communication. It is assumed that all agents are collaborative and that the locations of all nearby obstacles are known within the MAV's limited sensing range. Let each agent be subject to an a priori unknown and arbitrary reference $\boldsymbol{r}_i(t) = [\boldsymbol{p}_i^r(t)^T, \psi_i^r(t)]^T \in \mathbb{R}^4$, where $\boldsymbol{p}_i^r$ and ψ_i^r are the target position and yaw of agent i. The aggregate reference for the swarm, denoted by $\boldsymbol{r}_{1:N_{\mathrm{a}}}(t)$, is steady-state admissible at time t if $\boldsymbol{p}_{1:N_{\mathrm{a}}}^r(t)$ satisfies constraints **(8)–(10)**.

The purpose of this paper is to design a feedback control law in the form $\boldsymbol{U}_{1:N_{\mathrm{a}}}(\boldsymbol{r}_{1:N_{\mathrm{a}}}(t), \boldsymbol{x}_{1:N_{\mathrm{a}}}(t))$ such that the following objectives are achieved for a suitably large set of initial conditions $\boldsymbol{x}_{1:N_{\mathrm{a}}}(0)$:

- Safety: For any piecewise continuous reference $\boldsymbol{r}_{1:N_{\mathrm{a}}}(t)$, the control law is able to guarantee constraint satisfaction, i.e. the set of constraints **(7)–(10)** on the state and input variables of all agents $\boldsymbol{c}(\boldsymbol{x}_{1:N_{\mathrm{a}}}(t), \boldsymbol{U}_{1:N_{\mathrm{a}}}(t)) \geq 0, \forall t \geq 0$;
- Asymptotic Stability: If the reference $\boldsymbol{r}_{1:N_{\mathrm{a}}}$ is constant and steady-state admissible, the closed-loop system satisfies $\lim_{t \to \infty}([\boldsymbol{p}_{1:N_{\mathrm{a}}}(t)^T, \psi_{1:N_{\mathrm{a}}}(t)]^T) = \boldsymbol{r}_{1:N_{\mathrm{a}}}$;
- Robustness: The control law must ensure safety and stability in the presence of model uncertainty, sensor noise, and external disturbances;
- Reactiveness: The control law must run in real-time on-board the nano-quadrotor's hardware, without relying on off-board pre-generated trajectories;
- Scalability: Each agent must be capable of generating its own control input based on local information. To this end, inter-agent communication is limited to a given radius.

5 PROPOSED STRATEGY

The main challenge that arises from the control problem stated in **Section 4.3** is that it combines the nonlinear dynamics of the individual agent with the nonconvex constraints of the aggregated swarm. The higher-order nonlinear agent dynamics **(3)** would be significantly easier to stabilize in the absence of constraints, whereas the position constraints **(8)–(10)** would be easier to enforce if the agent dynamics were a first-order linear system $\dot{\boldsymbol{p}}_i = \boldsymbol{\rho}_i$ as in (Fathian et al., 2019). We propose a multi-layer control architecture that relies on the ERG framework (Nicotra and Garone, 2018) and decouples the control problem into more tractable sub-tasks to facilitate on-board implementation.

The first task, which is handled by the **Control Layer**, consists in pre-stabilizing the dynamics of each agent to a locally defined reference $\boldsymbol{v}_i(t) = [\boldsymbol{p}_i^v(t)^T, \psi_i^v(t)]^T \in \mathbb{R}^4$. This will be done using a classical inner-outer loop controller that does not account for system constraints and does not require any form of inter-agent coordination. The second task, which is handled by the **Navigation Layer**, consists in manipulating the aggregate auxiliary references $\boldsymbol{v}_{1:N_{\mathrm{a}}}(t)$ so that the constraints are always satisfied. This layer is also responsible for coordinating the overall swarm and reaching the target configuration $\boldsymbol{r}_{1:N_{\mathrm{a}}}(t)$. The proposed control architecture is illustrated in **Figure 2**. The detailed design of the control and navigation layers will be addressed in **Sections 6** and **7**, respectively.

6 CONTROL LAYER

The goal of the control layer is to pre-stabilize the individual quadrotors using a classical nonlinear inner-outer loop control law (Mellinger and Kumar, 2011; Hua et al., 2013). This is done without accounting for the state or input constraints, which will instead be handled by the navigation layer. The proposed architecture of the control layer is illustrated in **Figure 3**.

6.1 Inner-Outer Loop Control Law

The objective of the outer loop is to control the position of the quadrotor under the assumption that the attitude dynamics are instantaneous. To this end, we define the auxiliary control input $\boldsymbol{R}^d \in \mathrm{SO}(3)$ and assume that $\boldsymbol{R} \approx \boldsymbol{R}^d$. The position dynamics in the dynamic model **(3)** then become

$$m\ddot{\boldsymbol{p}} = T\boldsymbol{R}^d\hat{\boldsymbol{e}}_3 - mg\hat{\boldsymbol{e}}_3, \tag{11}$$

where $T\boldsymbol{R}^d\hat{\boldsymbol{e}}_3$ is the desired thrust vector expressed in $\mathcal{W}$. Using a PD control law with gravity compensation, the outer loop control inputs T and $\boldsymbol{R}^d$ are chosen so that

$$\boldsymbol{T}^d = T\boldsymbol{R}^d\hat{\boldsymbol{e}}_3 = m\left(\boldsymbol{K}_P(\boldsymbol{p}^v - \boldsymbol{p}) - \boldsymbol{K}_D\dot{\boldsymbol{p}} + g\hat{\boldsymbol{e}}_3\right), \tag{12}$$

where $\boldsymbol{K}_P, \boldsymbol{K}_D > 0$ are diagonal gain matrices. The total thrust can thus be obtained as

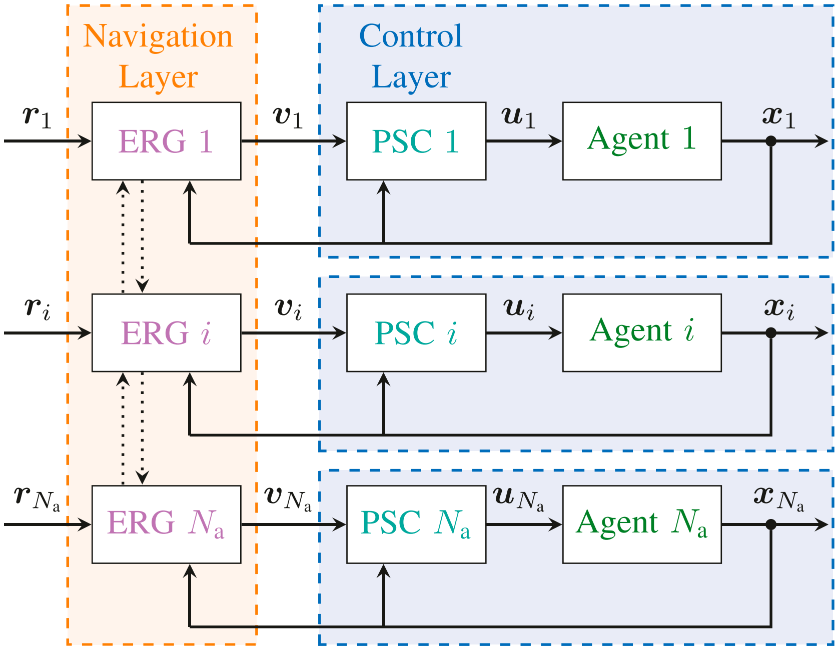


FIGURE 2 | Distributed Constrained Control Architecture – The higher-order dynamics of each agent in the multi-robot system are stabilized by a Pre-Stabilizing Control (PSC) unit that computes the control inputs $\boldsymbol{u}_i$ using only $\boldsymbol{x}_i$ for state feedback and without accounting for constraints. An Explicit Reference Governor (ERG) block is placed in a distributed fashion before each pre-stabilized agent and only relies on information $\boldsymbol{v}_{\mathcal{N}_i}$ available in its local one-hop spherical neighborhood $\mathcal{N}_i$ to enforce state and input constraints and achieve asymptotic convergence to $\boldsymbol{r}_i$. In this article $\boldsymbol{v}_{\mathcal{N}_i}$ represents the set of applied references $\boldsymbol{v}_k$ in the distributed policy or the set of states $\boldsymbol{x}_k$ in the decentralized policy (such that a worst-case approximation of $\boldsymbol{v}_k$ can be locally computed) for all agents k in the one-hop local neighborhood of agent i. We assume each agent can communicate in parallel with its neighbors.

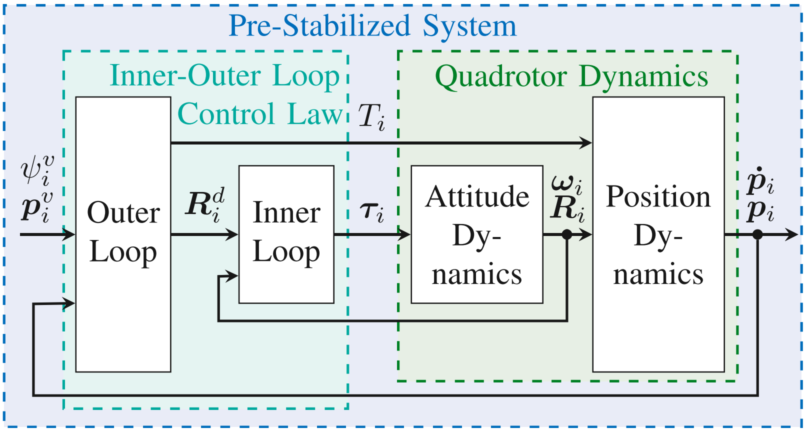


FIGURE 3 | Pre-Stabilizing Control Scheme – In the traditional inner-outer loop control paradigm, it is assumed that the inner loop control law stabilizes the attitude dynamics an order of magnitude faster than the outer loop control law stabilizes the position dynamics.

$$u_1 = T = m\left\|\boldsymbol{K}_P\left(\boldsymbol{p}^v - \boldsymbol{p}\right) - \boldsymbol{K}_D\dot{\boldsymbol{p}} + g\hat{\boldsymbol{e}}_3\right\|. \tag{13}$$

The target attitude is $\boldsymbol{R}^d = \boldsymbol{R}_{\psi^v}\boldsymbol{R}_{\alpha^d}$, where $\boldsymbol{R}_{\psi^v}$ is a standard rotation of ψ^v around the third axis, whereas $\boldsymbol{R}_{\alpha^d}$ is the minimum rotation α^d that aligns $\hat{z}_{\mathcal{W}}$ with the desired $\hat{z}_{\mathcal{B}}^d = \boldsymbol{T}^d/T$ and one can obtain it using the Rodrigues formula with $\alpha^d = \arctan\left(\sqrt{(T_x^d)^2 + (T_y^d)^2}/T_z^d\right)$.

The objective of the inner loop is to control the attitude dynamics of the UAV such that the rotation matrix $\boldsymbol{R}$ asymptotically tends to a constant $\boldsymbol{R}^d$. As detailed in Lee (2011), a possible strategy to compute the torque vector is to define the attitude error as

$$\boldsymbol{e}_R = \frac{1}{2}\left(\boldsymbol{R}^{d^T}\boldsymbol{R} - \boldsymbol{R}^T\boldsymbol{R}^d\right)^{\vee}, \tag{14}$$

and compute the control torques as follows,

$$\boldsymbol{\tau} = -\boldsymbol{K}_R\boldsymbol{e}_R - \boldsymbol{K}_\omega\boldsymbol{\omega}, \quad (15)$$

where $\boldsymbol{K}_R, \boldsymbol{K}_\omega > 0$ are diagonal gain matrices.

6.2 Robust Closed Loop Dynamics

The following Lemma states the robustness of the outer loop dynamics to attitude errors.

LEMMA 1. Let system **(3)** be subject to the outer loop controller (12), with $\boldsymbol{K}_P, \boldsymbol{K}_D > 0$, and the inner loop controller (15), with $\boldsymbol{K}_R, \boldsymbol{K}_\omega > 0$. Assume that the inner loop dynamics are sufficiently fast with respect to the outer loop dynamics. Given a constant applied position reference $\boldsymbol{p}^v$ and a constant applied yaw reference ψ^v, then

$$V(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{p}^v) = \begin{bmatrix} \boldsymbol{p} - \boldsymbol{p}^v \\ \dot{\boldsymbol{p}} \end{bmatrix}^T \boldsymbol{P} \begin{bmatrix} \boldsymbol{p} - \boldsymbol{p}^v \\ \dot{\boldsymbol{p}} \end{bmatrix}, \quad (16)$$

with

$$\boldsymbol{P} = \frac{1}{2}\begin{bmatrix} \boldsymbol{K}_P + \varepsilon\boldsymbol{K}_D^2 & \varepsilon\boldsymbol{K}_D \\ \varepsilon\boldsymbol{K}_D & \boldsymbol{I}_3 \end{bmatrix}, \quad (17)$$

is a Lyapunov function of the outer loop dynamics $\forall\varepsilon \in (0,1)$. Moreover, the outer loop is Input-to-State Stable (ISS) with restrictions on the attitude error.

Proof: Given $\forall\varepsilon \in (0,1)$, **(16)** is an ISS-Lyapunov candidate function for the outer loop dynamics. Noting that for a non-ideal inner loop $\boldsymbol{R}\hat{\boldsymbol{e}}_3 = \boldsymbol{R}\boldsymbol{R}^{d^T}\boldsymbol{R}^d\hat{\boldsymbol{e}}_3$, the closed loop position dynamics, obtained by combining **(3)** and **(12)**, without assuming $\boldsymbol{R}^d \approx \boldsymbol{R}$, have the form

$$\ddot{\boldsymbol{p}} = \tilde{\boldsymbol{R}}\boldsymbol{K}_P(\boldsymbol{p}^v - \boldsymbol{p}) - \tilde{\boldsymbol{R}}\boldsymbol{K}_D\dot{\boldsymbol{p}} + \left(\tilde{\boldsymbol{R}} - \boldsymbol{I}_3\right)g\hat{\boldsymbol{e}}_3, \quad (18)$$

where $\tilde{\boldsymbol{R}} = \boldsymbol{R}\boldsymbol{R}^{d^T}$ represents the attitude error. Equation (18) is a Linear Parameter Varying (LPV) system that can be written in state-space form

$$\begin{bmatrix} \dot{\boldsymbol{p}} \\ \ddot{\boldsymbol{p}} \end{bmatrix} = \boldsymbol{f}(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{p}^v) = \boldsymbol{A}\left(\tilde{\boldsymbol{R}}\right)\begin{bmatrix} \boldsymbol{p} \\ \dot{\boldsymbol{p}} \end{bmatrix} + \boldsymbol{B}\left(\tilde{\boldsymbol{R}}\right)\begin{bmatrix} \boldsymbol{p}^v \\ \boldsymbol{0}_{3\times1} \end{bmatrix} + d\left(\tilde{\boldsymbol{R}}\right), \quad (19)$$

with

$$\boldsymbol{A}\left(\tilde{\boldsymbol{R}}\right) = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 \\ -\tilde{\boldsymbol{R}}\boldsymbol{K}_P & -\tilde{\boldsymbol{R}}\boldsymbol{K}_D \end{bmatrix},$$

$$\boldsymbol{B}\left(\tilde{\boldsymbol{R}}\right) = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \tilde{\boldsymbol{R}}\boldsymbol{K}_P & \boldsymbol{0}_{3\times3} \end{bmatrix}.$$

Noting that $\boldsymbol{A}(\boldsymbol{I}_3)^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}(\boldsymbol{I}_3) < 0$ as detailed in (Khalil, 2001, Example 4.5, pp. 121–122), it follows that $\boldsymbol{A}(\tilde{\boldsymbol{R}})^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}(\tilde{\boldsymbol{R}}) \leq 0$ for $\tilde{\boldsymbol{R}}$ sufficiently close to $\boldsymbol{I}_3$ (i.e. for a sufficiently small attitude error). This shows that **(18)** is Input to State Stable (ISS) with respect to sufficiently small attitude errors. ■

7 NAVIGATION LAYER

7.1 Distributed Explicit Reference Governor

The ERG is a general framework for the constrained control of nonlinear systems introduced in (Garone and Nicotra, 2016; Nicotra and Garone, 2018). Consider a pre-stabilized system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{v})$ such that, if the applied reference $\boldsymbol{v}$ remains constant, the closed-loop equilibrium point $\overline{\boldsymbol{x}}_v$ is asymptotically stable. Given a continuous steady-state admissible path $\boldsymbol{\Phi} : [0,1] \rightarrow \mathbb{R}^3$ between an initial reference $\boldsymbol{\Phi}(0) = \boldsymbol{v}(0)$ and a target reference $\boldsymbol{\Phi}(1) = \boldsymbol{r}$, the principle behind the ERG is to generate a reference $\boldsymbol{v}(t) \in \{\boldsymbol{\Phi}(s) | s \in [0,1]\}$ such that

- the transient dynamics of the closed-loop system cannot cause a constraint violation;
- $\lim_{t\rightarrow\infty}\boldsymbol{v}(t) = \boldsymbol{\Phi}(1)$.

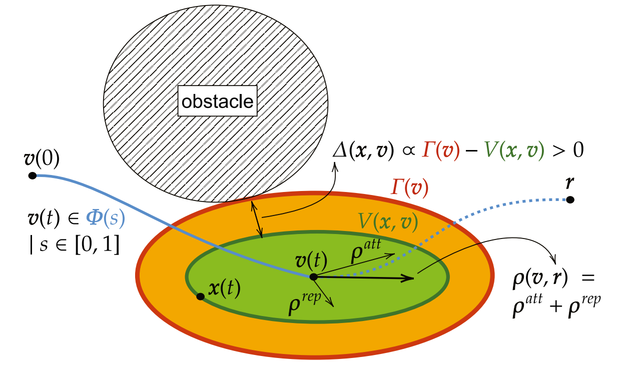


FIGURE 4 | Basic Idea of the Invariant Level Set Explicit Reference Governor – The spherical obstacle is avoided by moving the applied reference $\boldsymbol{v}(t)$ over the a priori unknown (i.e. non pre-computed) path $\boldsymbol{\Phi}(s)$ of steady-state admissible equilibria. The green ellipsoid represents the invariant level set value $V(\boldsymbol{x}, \boldsymbol{v})$ which embeds the future trajectory of $\boldsymbol{x}(t)$ if the current $\boldsymbol{v}(t)$ were to remain constant. The orange ellipsoid represents the threshold value $\Gamma(\boldsymbol{v})$ of the invariant level set that touches the obstacle constraint. The Dynamic Safety Margin (DSM) $\Delta(\boldsymbol{x}, \boldsymbol{v})$ is proportional to the difference between these level-set values and represents how safe it is to change $\boldsymbol{v}(t)$ in the direction of the Navigation Field (NF) $\boldsymbol{\rho}(\boldsymbol{v}, \boldsymbol{r})$, with attraction toward the desired reference $\boldsymbol{r}$ and repulsion away from obstacles.

However, rather than pre-computing a suitable trajectory $\boldsymbol{v}(t)$, the ERG achieves these objectives by continuously manipulating the derivative of the applied reference as follows

$$\dot{\boldsymbol{v}} = \boldsymbol{\rho}(\boldsymbol{v}, \boldsymbol{r})\Delta(\boldsymbol{x}, \boldsymbol{v}), \quad (20)$$

where $\boldsymbol{\rho}(\boldsymbol{v}, \boldsymbol{r})$ is the **Navigation Field** (NF), i.e. a vector field that generates the desired steady-state admissible path $\boldsymbol{\Phi}(s)$, and $\Delta(\boldsymbol{x}, \boldsymbol{v})$ is the **Dynamic Safety Margin** (DSM), i.e. a scalar that quantifies the "distance" between the transient dynamics of the pre-stabilized system and the constraint boundaries if the current $\boldsymbol{v}(t)$ were to remain constant. The principle behind the ERG framework is illustrated in **Figure 4**.

This section extends the ERG framework to handle the case of multi-agent systems. The main challenge is given by the fact that the Distributed ERG (D-ERG) solution must ensure the satisfaction of multi-agent coordination constraints $\boldsymbol{g}(\boldsymbol{x}_i, \boldsymbol{x}_k) \geq 0$, such as the collision avoidance constraints (10). These constraints are not only dependent on agent's i own dynamics, but also on the dynamics of agents k with $k \neq i$. Hence, the original ERG framework, presented in (Nicotra and

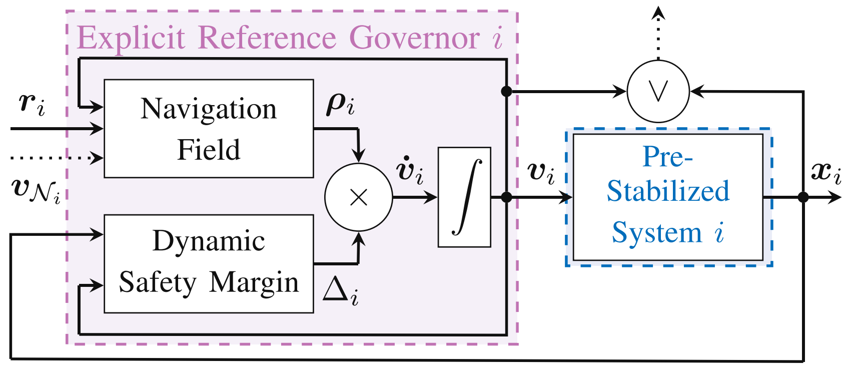


FIGURE 5 | Distributed Explicit Reference Governor (D-ERG) Scheme for Agent i. To ensure that the constraints are satisfied for any desired reference configuration $\boldsymbol{r}_i$, the ERG manipulates the rate of change of the applied reference $\dot{\boldsymbol{v}}_i$ by computing a suitable Navigation Field (NF) $\boldsymbol{\rho}_i$ and a Dynamic Safety Margin (DSM) Δ_i. The NF determines the current direction of $\dot{\boldsymbol{v}}_i$ and the DSM regulates the modulus of $\dot{\boldsymbol{v}}_i$ such that dynamic transients do not cause constraint violations. Agent i relies on the signal $\boldsymbol{v}_{\mathcal{N}_i}$, as defined in **Figure 2**, that is available in its local one-hop neighborhood and communicates its own signals $\boldsymbol{v}_i$ or $\boldsymbol{x}_i$ (but not both) with its neighboring agents, making the ERG distributed.

Garone, 2018, Theorem 1), would require a single, centralized ERG scheme to enforce the full set of constraints $\boldsymbol{c}(\boldsymbol{x}_{1:N_a}, \boldsymbol{v}_{1:N_a}) \geq 0$ on the aggregated states and references. Computing a single, non-conservative DSM would be challenging. Moreover, this scheme would inherently limit the velocity of the aggregate reference $\dot{\boldsymbol{v}}_{1:N_a}$ based on the agent that is closest to constraint violation, resulting in poor performance.

Here, the objective is to show that it is possible to ensure convergence and constraint satisfaction for the overall swarm by manipulating the reference of each agent in a distributed fashion as follows

$$\dot{\boldsymbol{v}}_i = \boldsymbol{\rho}(\boldsymbol{v}_{\mathcal{N}_i}, \boldsymbol{r}_i)\Delta(\boldsymbol{x}_i, \boldsymbol{v}_i), \tag{21}$$

with $\boldsymbol{v}_{\mathcal{N}_i}$ defined in **Figure 2**. The proposed solution computes a DSM for each agent and is based on decomposing the multi-agent coordination constraints $\boldsymbol{g}(\boldsymbol{x}_i, \boldsymbol{x}_k) \geq 0$ into an auxiliary constraint on the references, i.e. $\boldsymbol{\gamma}_1(\bar{\boldsymbol{x}}_{\boldsymbol{v}_i}, \bar{\boldsymbol{x}}_{\boldsymbol{v}_k}) \geq \delta$, and an auxiliary constraint on the dynamics of the individual agents, i.e. $\boldsymbol{\gamma}_2(\boldsymbol{x}_i, \boldsymbol{v}_i) \geq 0$, which can be accounted for in the NF and the DSM, respectively. In what follows $\boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{v}_i) \geq 0$ denotes the set of agent independent constraints, such as constraints **(7)–(9)**. The rest of this section provides the updated definitions of the NF $\boldsymbol{\rho}(\boldsymbol{v}_{\mathcal{N}_i}, \boldsymbol{r}_i)$ and the DSM $\Delta(\boldsymbol{x}_i, \boldsymbol{v}_i)$ used in **(21)** by identifying sufficient conditions for the correct behavior of the D-ERG, as proven in Theorem 1. The schematic representation of the D-ERG is illustrated in **Figure 5**.

Definition 1 (Navigation Field). Let the NF $\boldsymbol{\rho}(\boldsymbol{v}_{\mathcal{N}_i}, \boldsymbol{r}_i)$ be such that, for any possibly time-varying piecewise continuous reference $\boldsymbol{r}_{1:N}$, the initial value problem

$$\begin{cases} \dot{\boldsymbol{v}}_i(\tau) = \boldsymbol{\rho}(\boldsymbol{v}_{\mathcal{N}_i}(\tau), \boldsymbol{r}_i), \\ \boldsymbol{v}_i(0) = \boldsymbol{v}_i, \end{cases} \tag{22}$$

satisfies the following.

1. $\|\boldsymbol{\rho}(\boldsymbol{v}_{\mathcal{N}_i}, \boldsymbol{r}_i)\|$ is finite for all possible $(\boldsymbol{v}_{\mathcal{N}_i}, \boldsymbol{r}_i)$;
2. $\boldsymbol{h}(\bar{\boldsymbol{x}}_{\boldsymbol{v}_i}, \boldsymbol{v}_i) \geq \delta \Rightarrow \boldsymbol{h}(\bar{\boldsymbol{x}}_{\boldsymbol{v}_i(\tau)}, \boldsymbol{v}_i(\tau)) \geq \delta, \forall \tau \geq 0$;
3. $\boldsymbol{\gamma}_1(\bar{\boldsymbol{x}}_{\boldsymbol{v}_i}, \bar{\boldsymbol{x}}_{\boldsymbol{v}_k}) \geq \delta \Rightarrow \boldsymbol{\gamma}_1(\bar{\boldsymbol{x}}_{\boldsymbol{v}_i(\tau)}, \bar{\boldsymbol{x}}_{\boldsymbol{v}_k(\tau)}) \geq \delta, \forall \tau \geq 0$;
4. For any constant reference $\boldsymbol{r}_{1:N}$, there exists a non-empty set of initial conditions $\mathcal{V}$ such that $\forall \boldsymbol{v}_{1:N} \in \mathcal{V}$, then

$$\boldsymbol{h}(\bar{\boldsymbol{x}}_{\boldsymbol{r}_{1:N}}, \boldsymbol{r}_{1:N}) \geq \delta \wedge \boldsymbol{\gamma}_1(\bar{\boldsymbol{x}}_{\boldsymbol{r}_{1:N}}) \geq \delta \Rightarrow \lim_{\tau \to \infty} \boldsymbol{v}_{1:N}(\tau) = \boldsymbol{r}_{1:N}.$$

The key takeaway from Definition 1 is that it only considers the first-order dynamics **(22)**. Thus, the NF is only responsible for generating a steady-state admissible path that connects the current references $\boldsymbol{v}_{1:N}$ to the target references $\boldsymbol{r}_{1:N}$. Since the NF does not account for the system dynamics, we refer to $\delta > 0$ as the "static safety margin".

Definition 2 (Dynamic Safety Margin). Let the DSM $\Delta(\boldsymbol{x}_i, \boldsymbol{v}_i)$ be such that the solution of the initial value problem

$$\begin{cases} \dot{\boldsymbol{\xi}}_i(\tau) = \boldsymbol{f}(\boldsymbol{\xi}_i(\tau), \boldsymbol{v}_i), \\ \boldsymbol{\xi}_i(0) = \boldsymbol{x}_i, \end{cases} \tag{23}$$

satisfies the following.

1. $\Delta(\boldsymbol{x}_i, \boldsymbol{v}_i) > 0 \Rightarrow \boldsymbol{h}(\boldsymbol{\xi}_i(\tau), \boldsymbol{v}_i) > 0, \forall \tau \geq 0$;
2. $\Delta(\boldsymbol{x}_i, \boldsymbol{v}_i) > 0 \Rightarrow \boldsymbol{\gamma}_2(\boldsymbol{\xi}_i(\tau), \boldsymbol{v}_i) > 0, \forall \tau \geq 0$;
3. $\Delta(\boldsymbol{x}_i, \boldsymbol{v}_i) \geq 0 \Rightarrow \boldsymbol{h}(\boldsymbol{\xi}_i(\tau), \boldsymbol{v}_i) \geq 0, \forall \tau \geq 0$;
4. $\Delta(\boldsymbol{x}_i, \boldsymbol{v}_i) \geq 0 \Rightarrow \boldsymbol{\gamma}_2(\boldsymbol{\xi}_i(\tau), \boldsymbol{v}_i) \geq 0, \forall \tau \geq 0$;
5. $\Delta(\boldsymbol{x}_i, \boldsymbol{v}_i) = 0 \Rightarrow \Delta(\boldsymbol{\xi}_i(\tau), \boldsymbol{v}_i) \geq 0, \forall \tau \geq 0$;
6. $\forall \delta > 0$, $\exists \varepsilon > 0$ such that

$$\boldsymbol{h}(\bar{\boldsymbol{x}}_{\boldsymbol{v}_i}, \boldsymbol{v}_i) \geq \delta \wedge \boldsymbol{\gamma}_1\left(\bar{\boldsymbol{x}}_{\boldsymbol{v}_i}, \bar{\boldsymbol{x}}_{\boldsymbol{v}_{k \neq i}}\right) \geq \delta \Rightarrow \Delta(\bar{\boldsymbol{x}}_{\boldsymbol{v}_i}, \boldsymbol{v}_i) \geq \varepsilon.$$

The intuition behind the DSM is that it quantifies the distance between the constraints and the transient dynamics of the individual closed-loop system.

Theorem 1. Consider N identical pre-stabilized systems $\dot{\boldsymbol{x}}_i = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{v}_i)$ such that, if the applied reference $\boldsymbol{v}_i$ remains constant, the closed-loop equilibrium point $\bar{\boldsymbol{x}}_{\boldsymbol{v}_i}$ is asymptotically stable. Let each agent be subject to a set of agent-independent constraints $\boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{v}_i) \geq 0$ and a set of multi-agent coordination constraints $\boldsymbol{g}(\boldsymbol{x}_i, \boldsymbol{x}_k) \geq 0$ with $i \neq k$. Moreover,

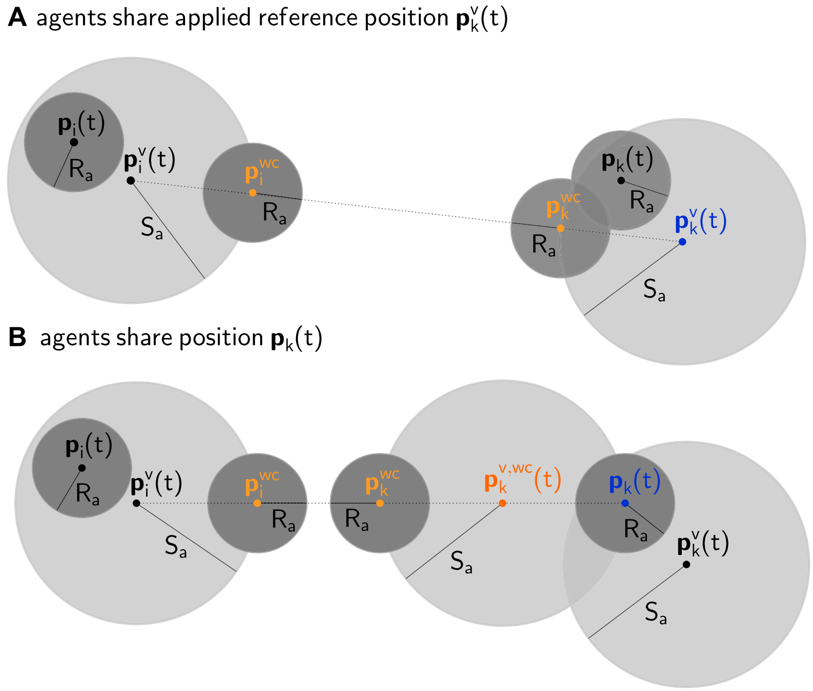


FIGURE 6 | Geometric 2D representation of distributed collision avoidance between two pre-stabilized agents i **(left)** and k **(right)** with safety radii R_a (dark gray disks), drawn from the perspective of agent i. The current position of each agent is $\boldsymbol{p}_i(t)$, $\boldsymbol{p}_k(t)$, whereas their current reference is $\boldsymbol{p}_i^v(t)$,$\boldsymbol{p}_k^v(t)$. Due to the auxiliary constraint (38) (in light gray), accounted for in the DSM, the smallest possible distance between the two agents is equal to the distance between their worst-case future positions $\boldsymbol{p}_i^{wc}$, $\boldsymbol{p}_k^{wc}$. Together with the auxiliary constraint **(37)**, which is enforced by the NF, this ensures the collision avoidance constraint **(10)**. If the agents share their references (**case A**), agent i can compute the worst-case future position of agent k based on its current reference $\boldsymbol{p}_k^v(t)$. If agent i only knows the position of agent k (**case B**), it must use $\boldsymbol{p}_k(t)$ to compute the worst-case current reference $\boldsymbol{p}_k^{v,wc}(t)$ and must then compute the worst-case future position based on $\boldsymbol{p}_k^{v,wc}(t)$.

let the auxiliary constraints $\gamma_1(\bar{\boldsymbol{x}}_{v_i}, \bar{\boldsymbol{x}}_{v_k}) \geq \delta$ and $\gamma_2(\boldsymbol{x}_i, \boldsymbol{v}_i) \geq 0$ be defined so that

$$\left.\begin{array}{l}\gamma_1(\bar{\boldsymbol{x}}_{v_i}, \bar{\boldsymbol{x}}_{v_k}) \geq \delta \wedge \gamma_2(\boldsymbol{x}_i, \boldsymbol{v}_i) \geq 0 \\ \gamma_1(\bar{\boldsymbol{x}}_{v_k}, \bar{\boldsymbol{x}}_{v_i}) \geq \delta \wedge \gamma_2(\boldsymbol{x}_k, \boldsymbol{v}_k) \geq 0\end{array}\right\} \Rightarrow \boldsymbol{g}(\boldsymbol{x}_i, \boldsymbol{x}_k) \geq 0. \tag{24}$$

Given the navigation field $\boldsymbol{\rho}(\boldsymbol{v}_{\mathcal{N}_i}, \boldsymbol{r}_i)$ and the dynamic safety margin $\Delta(\boldsymbol{x}_i, \boldsymbol{v}_i)$, let the initial conditions $\boldsymbol{x}_{1:N}(0), \boldsymbol{v}_{1:N}(0)$ be such that $\Delta(\boldsymbol{x}_{1:N}(0), \boldsymbol{v}_{1:N}(0)) \geq 0$. Then, the D-ERG formulation **(21)** ensures constraint satisfaction, i.e.

- $\boldsymbol{h}(\boldsymbol{x}_{1:N}(t), \boldsymbol{v}_{1:N}(t)) \geq 0, \quad \forall t \geq 0;$
- $\boldsymbol{g}(\boldsymbol{x}_i(t), \boldsymbol{x}_k(t)) \geq 0, \quad \forall t \geq 0, \forall i \in \{1:N\}, \forall k \neq i,$

for any piecewise continuous reference $\boldsymbol{r}_{1:N}(t)$.

Moreover, given a constant aggregate reference $\boldsymbol{r}_{1:N}$ satisfying $\boldsymbol{h}(\bar{\boldsymbol{x}}_{r_i}, \boldsymbol{r}_i) \geq \delta$ and $\boldsymbol{g}(\bar{\boldsymbol{x}}_{r_i}, \bar{\boldsymbol{x}}_{r_k}) \geq \delta$, with $i \in \{1:N\}, k \neq i$, the D-ERG formulation **(21)** also ensures convergence, i.e.

- $\lim_{t\to\infty} \boldsymbol{x}_{1:N}(t) = \bar{\boldsymbol{x}}_{r_{1:N}},$

as long as $\boldsymbol{v}_{1:N}(t) \in \mathcal{V}, \quad \forall t \geq 0$.

Proof: As detailed in the proof of (Nicotra and Garone, 2018), Theorem 1, it can be shown that **(21)** ensures $\Delta(\boldsymbol{x}_i(t), \boldsymbol{v}_i(t)) \geq 0$, $\forall t \geq 0$. As a result, it follows by definition of the DSM that $\boldsymbol{h}(\boldsymbol{x}_{1:N}(t), \boldsymbol{v}_{1:N}(t)) \geq 0$ and $\gamma_2(\boldsymbol{x}_{1:N}(t), \boldsymbol{v}_{1:N}(t)) \geq 0$, $\forall t \geq 0$. Moreover, it follows by definition of the NF that, for any piecewise continuous and non-negative signal $\Delta(t)$, the solution to $\dot{\boldsymbol{v}}_i = \Delta(t)\boldsymbol{\rho}(\boldsymbol{v}_{\mathcal{N}_i}, \boldsymbol{r}_i)$ satisfies $\gamma_1(\bar{\boldsymbol{x}}_{v_i(t)}, \bar{\boldsymbol{x}}_{v_k(t)}) \geq \delta$, $\forall t \geq 0$, $\forall i \in \{1:N\}$, $\forall k \neq i$. As a result it follows from **(24)** that $\boldsymbol{g}(\boldsymbol{x}_i(t), \boldsymbol{x}_k(t)) \geq 0$, $\forall t \geq 0$, $\forall i \in \{1:N\}$, $\forall k \neq i$. Finally, the convergence result $\lim_{t\to\infty} \boldsymbol{x}_{1:N}(t) = \bar{\boldsymbol{x}}_{r_{1:N}}$ follows from the property $\Delta(\bar{\boldsymbol{x}}_{v_{1:N}}, \boldsymbol{v}_{1:N}) \geq \varepsilon$, as detailed in the proof of (Nicotra and Garone, 2018, Theorem 1). ■

It is worth noting that, if $\mathcal{V}$ is equal to the entire set of steady-state admissible constraints, Theorem 1 implies convergence $\forall \boldsymbol{v}_{1:N}(0) \in \mathcal{V}$. However, if the NF admits deadlock configurations, the D-ERG will inherit the same limitations. The following subsections specialize the proposed D-ERG theory to the constrained control of a swarm of quadrotors. The choice of the auxiliary constraints that ensure multi-agent collision avoidance, as stated in **(24)**, is illustrated in **Figure 6**. The pseudocode of the D-ERG is given in **Algorithm 1**, and the accompanying **Table 1**, which lists the type and amount of instructions to be executed, shows that the proposed D-ERG approach is computationally efficient and scalable.

7.2 Navigation Field

As detailed in (Nicotra and Garone, 2018), the NF of agent i can be designed using a traditional attraction and repulsion field[1]

$$\boldsymbol{\rho}(\boldsymbol{v}_{\mathcal{N}_i}, \boldsymbol{r}_i) = \boldsymbol{\rho}_i^{\text{att}} + \boldsymbol{\rho}_i^{\text{rep}}, \tag{25}$$

[1] dependency of $\boldsymbol{\rho}$ on $(\boldsymbol{v}_{\mathcal{N}_i}, \boldsymbol{r}_i)$ is omitted for simplicity of notation.

Algorithm 1: Pseudocode of the Distributed Explicit Reference Governor (D-ERG) Algorithm for Agent *i*.

```
   Input  : desired position $p_i^r(t)$ and yaw $\psi_i^r(t)$
   Output : safe to apply position $p_i^v(t)$ and yaw $\psi_i^v(t)$
1  Offline pre-computations: begin
2      $P$, $\hat{P}_T$ ← traditional and optimally aligned total thrust Lyapunov matrix using (17) and (50);
3      $\Gamma_{T_{\max}}$, $\Gamma_{T_{\min}}$ ← saturation threshold values using Trad Lyap (46), Opt Lyap (52) or Opt Inv
        (54);
4      $\Gamma_a$ ← position error threshold value as in (63);
5  Online and on-board operations: begin
6      $p_i(t)$, $\dot{p}_i(t)$, $\Theta_i(t)$, $\omega_i(t)$ ← state feedback;
7      $p_i^r(t)$, $\psi_i^r(t)$ ← desired commands;
8      $p_{ik}^v(t)$ ← relative agent info;
9      Navigation Field (NF): begin
10         $\rho_i^{att}$ ← attraction field using (26);
11         $0_{4\times1}$ ← initialize $\rho_i^w$, $\rho_i^o$, $\rho_i^a$;
12         foreach wall $w_j$ within influence $\zeta_w$ do
13         add wall repulsion field to $\rho_i^w$ as in (29);
14         foreach obstacle $o_j$ within influence $\zeta_o$ do
15         add obstacle repulsion field to $\rho_i^o$ as in (31);
16         foreach agent $k \neq i$ within influence $\zeta_a$ do
17         add agent repulsion field to $\rho_i^a$ as in (34);
18         $\rho_i^{rep} \leftarrow \rho_i^w + \rho_i^o + \rho_i^a$ as in (28);
19         $\rho_i \leftarrow \rho_i^{att} + \rho_i^{rep}$ as in (25);
20         if $\rho_i$ penetrates a margin $\delta_{w/o/a}$ then
21             $\rho_i$ ← project $\rho_i$ tangentially on constraint;
22     Dynamic Safety Margin (DSM): begin
23         $V(p_i, \dot{p}_i, p_i^v)$, $V_T(p_i, \dot{p}_i, p_i^v)$ ← Trad Lyap (40) and Opt Lyap (48) level-set values;
24         $\Delta_i^s$, $\Delta_i^{s,\psi}$ ← input saturation DSM using Trad Lyap (47), Opt Lyap (53), or Opt Inv (55) for
            position and (56) for yaw;
25         $\Delta_i^a$ ← position error DSM using (64);
26         foreach wall $w_j$ within influence $\zeta_w$ do
27         $\Gamma_{w_j}$ ← threshold value as in (57);
28         $\Delta_i^w$ ← minimal DSM as in (58);
29         foreach obstacle $o_j$ within influence $\zeta_o$ do
30         $\Gamma_{o_j}$ ← threshold value similar as in (57);
31         $\Delta_i^o$ ← minimal DSM as in (61);
32         $\Delta_i \leftarrow \max(\min(\Delta_i^s, \Delta_i^w, \Delta_i^o, \Delta_i^a), 0)$ as (39);
33     Scale NF by DSM's magnitude: begin
34         $\begin{bmatrix}\dot{p}_i^v(t)\\ \dot{\psi}_i^v(t)\end{bmatrix} \leftarrow \begin{bmatrix}\Delta_i I_3 & 0_{3\times1}\\ 0_{1\times3} & \Delta_i^{s,\psi}\end{bmatrix} \cdot \rho(v_{\mathcal{N}_i}, r_i)$
35     ; Forward Euler integration: begin
36         $\begin{bmatrix}p_i^v(t)\\ \psi_i^v(t)\end{bmatrix} \leftarrow \begin{bmatrix}p_i^v(t-1)\\ \psi_i^v(t-1)\end{bmatrix} + \begin{bmatrix}\dot{p}_i^v(t)\\ \dot{\psi}_i^v(t)\end{bmatrix} \cdot dt$;
37         save this result for one sample $dt$;
38     apply new $p_i^v(t)$ and $\psi_i^v(t)$ to pre-stabilized UAV;
39     communicate new $p_i(t)$ or $p_i^v(t)$ with neighbours;
```

where the attraction field is

$$\boldsymbol{\rho}_i^{\mathrm{att}} = \left[l\left(\boldsymbol{p}_i^r - \boldsymbol{p}_i^v, \eta\right)^T, l\left(\psi_i^r - \psi_i^v, \eta_\psi\right)\right]^T, \tag{26}$$

$\eta, \eta_\psi > 0$ are small smoothing radii chosen to avoid numerical problems when $\|\boldsymbol{r}_i - \boldsymbol{v}_i\| \to 0$, and

$$l(\boldsymbol{x}, \eta) = \frac{\boldsymbol{x}}{\max(\|\boldsymbol{x}\|, \eta)}. \tag{27}$$

The repulsion field is the sum of linear repulsion fields pushing away from walls (w), obstacles (o), and nearby agents (a), i.e.

$$\boldsymbol{\rho}_i^{\mathrm{rep}} = \boldsymbol{\rho}_i^{\mathrm{w}} + \boldsymbol{\rho}_i^{\mathrm{o}} + \boldsymbol{\rho}_i^{\mathrm{a}}. \tag{28}$$

The repulsion field of all wall constraints is

$$\boldsymbol{\rho}_i^{\mathrm{w}} = -\sum_{j=1}^{N_{\mathrm{w}}} \max\left(\frac{\zeta_{\mathrm{w}} - \left(d_{\mathrm{w}_j} - R_{\mathrm{a}} - \boldsymbol{c}_{\mathrm{w}_j}^T \boldsymbol{p}_i^v\right)}{\zeta_{\mathrm{w}} - \delta_{\mathrm{w}}}, 0\right)\begin{bmatrix}\widehat{\boldsymbol{c}}_{\mathrm{w}_j}\\ 0\end{bmatrix}, \tag{29}$$

where $\zeta_{\mathrm{w}} > 0$ is the influence margin outside of which the repulsion field has no effect and $\delta_{\mathrm{w}} \in (0, \zeta_{\mathrm{w}})$ is the static safety margin which guarantees that the reference is strictly steady-state admissible. The repulsion field of all static cylindrical obstacles includes the conservative (co) term

$$\boldsymbol{\rho}_i^{\mathrm{o,co}} = -\sum_{j=1}^{N_{\mathrm{o}}} \max\left(\frac{\zeta_{\mathrm{o}_j} - C_j\left(\boldsymbol{p}_i^v\right)}{\zeta_{\mathrm{o}_j} - \delta_{\mathrm{o}_j}}, 0\right)\begin{bmatrix}\widehat{\left(\boldsymbol{o}_j - \boldsymbol{p}_i^v\right)}_{xy}\\ 0\end{bmatrix}, \tag{30}$$

with an influence margin $\zeta_{\mathrm{o}_j} > 0$, a static safety margin $\delta_{\mathrm{o}_j} \in (0, \zeta_{\mathrm{o}_j})$ and $C_j(\boldsymbol{p}_i^v) = \|\boldsymbol{p}_i^v - \boldsymbol{o}_j\|_{xy} - (R_{\mathrm{o}_j} + R_{\mathrm{a}})$. For spherical

TABLE 1 | Computational Requirements of the D-ERG Algorithm – Type and amount of operations to be executed on-board an agent having in its one-hop local neighborhood N_{w} walls, N_{o} static obstacles, and $N_{\mathrm{a}} - 1$ dynamic agents. The required computations are basic arithmetic scalar, vector and matrix operations, scalar and vector min/max operations that scale approximately linear with the number of state constraints. Note that there is no iterative solver or matrix inversion required.

		Navigation Field							Dynamic Safety Margin									
Line # in pseudocode		10	11	12 – 13	14 – 15	16 – 17	18 – 19	20 – 21	23	24	25	26 – 28	29 – 31	32	34	36
Type and amount of operations executed	$a \pm b$	1		$4N_{\mathrm{w}}$	$13N_{\mathrm{o}}$	$13N_{\mathrm{a}}$				5	1	$2N_{\mathrm{w}}+1$	$4N_{\mathrm{o}}+1$			1
	$\boldsymbol{v}_3 \pm \boldsymbol{w}_3$	1	3	N_{w}	$3N_{\mathrm{o}}-1$	$3N_{\mathrm{a}}-1$	3	1					N_{o}			1
	$\boldsymbol{v}_6 \pm \boldsymbol{w}_6$								1							
	$a * \div b$	1		N_{w}	N_{o}	N_{a}		2		7	1	$N_{\mathrm{w}}+2$	$2N_{\mathrm{o}}+1$		1	1
	$\boldsymbol{v}_3 * \div a$	1		N_{w}	$4N_{\mathrm{o}}$	$4N_{\mathrm{a}}$		1					N_{o}		1	1
	$\boldsymbol{v}_3^T \cdot \boldsymbol{w}_3$			N_{w}								N_{w}	N_{o}			
	$\boldsymbol{M}_{3\times3} * \boldsymbol{v}_3$															
	$\boldsymbol{v}_6^T \cdot \boldsymbol{w}_6$								2				$2N_{\mathrm{o}}$			
	$\boldsymbol{M}_{6\times6} * \boldsymbol{v}_6$								2				N_{o}			
	$\lvert a \rvert$	1														
	$\lvert\lvert \boldsymbol{v}_3 \rvert\rvert$	1			$2N_{\mathrm{o}}$	$2N_{\mathrm{a}}$							N_{o}			
	$\max / \min(a, b)$	2		N_{w}	N_{o}	N_{a}		1		2				1		
	$\max / \min(\boldsymbol{v}_4)$									2				1		
	$\max / \min(\boldsymbol{v}_{N_{\mathrm{w}}})$											1				
	$\max / \min(\boldsymbol{v}_{N_{\mathrm{o}}})$												1			

constraints, one can just use the full Euclidean norm and not project $(\boldsymbol{o}_j - \boldsymbol{p}_i^{\nu})$ on the xy-plane. As detailed in (Koditschek and Rimon, 1990), however, conservative vector fields cannot achieve global stability in the presence of obstacle constraints. Therefore, the repulsion field also includes a non-conservative (n-co) term that destabilizes local saddle points

$$\boldsymbol{\rho}_i^{\mathrm{o}} = \boldsymbol{\rho}_i^{\mathrm{o,co}} + \boldsymbol{\rho}_i^{\mathrm{o,n-co}}, \tag{31}$$

where

$$\boldsymbol{\rho}_i^{\mathrm{o,\ n-co}} = \begin{cases} \alpha_{\mathrm{o}_j} \sum_{j=1}^{N_{\mathrm{o}}} \begin{bmatrix} \boldsymbol{o}_j\widehat{(2) - \boldsymbol{p}}_i^{\nu}(2) \\ -\boldsymbol{o}_j(1) + \boldsymbol{p}_i^{\nu}(1) \\ 0 \\ 0 \end{bmatrix} & \text{if } \zeta_{\mathrm{o}_j} \geq C_j(\boldsymbol{p}_i^{\nu}), \\ \boldsymbol{0}_{4\times1} & \text{if } \zeta_{\mathrm{o}_j} < C_j(\boldsymbol{p}_i^{\nu}) \end{cases} \tag{32}$$

with circulation gain $\alpha_{\mathrm{o}_j} > 0$. For the case of a sphere, the term within brackets can be replaced by

$$\begin{bmatrix} -\boldsymbol{o}_j(2) + \boldsymbol{p}_i^{\nu}\widehat{(2) +} \boldsymbol{o}_j(3) - \boldsymbol{p}_i^{\nu}(3) \\ \boldsymbol{o}_j(1) - \boldsymbol{p}_i^{\nu}(1) - \boldsymbol{o}_j(3) + \boldsymbol{p}_i^{\nu}(3) \\ -\boldsymbol{o}_j(1) + \boldsymbol{p}_i^{\nu}(1) + \boldsymbol{o}_j(2) - \boldsymbol{p}_i^{\nu}(2) \\ 0 \end{bmatrix}. \tag{33}$$

In a similar way, one can define the repulsion field that acts on agent i caused by the other agents k as

$$\boldsymbol{\rho}_{\mathrm{i}}^{\mathrm{a}} = \boldsymbol{\rho}_i^{\mathrm{a,co}} + \boldsymbol{\rho}_i^{\mathrm{a,n-co}}, \tag{34}$$

where

$$\boldsymbol{\rho}_i^{\mathrm{a,co}} = -\sum_{\substack{k=1 \\ k\neq i}}^{N_{\mathrm{a}}} \max\left(\frac{\zeta_{\mathrm{a}} - C_{ik}(\boldsymbol{p}_{ik}^{\nu})}{\zeta_{\mathrm{a}} - \delta_{\mathrm{a}}}, 0\right) \begin{bmatrix} \widehat{\boldsymbol{p}}_{ik}^{\nu} \\ 0 \end{bmatrix}, \tag{35}$$

with $C_{ik}(\boldsymbol{p}_{ik}^{\nu}) = \|\boldsymbol{p}_{ik}^{\nu}\|_{xy} - 2R_{\mathrm{a}} - 2S_{\mathrm{a}}$, and

$$\boldsymbol{\rho}_i^{\mathrm{a,n-co}} = \begin{cases} \alpha_{\mathrm{a}} \sum_{\substack{k=1 \\ k\neq i}}^{N_{\mathrm{a}}} \begin{bmatrix} \widehat{\boldsymbol{p}_{ik}^{\nu}(2)} \\ -\boldsymbol{p}_{ik}^{\nu}(1) \\ 0 \\ 0 \end{bmatrix} & \text{if } \zeta_{\mathrm{a}} \geq C_{ik}(\boldsymbol{p}_{ik}^{\nu}), \\ \boldsymbol{0}_{4\times1} & \text{if } \zeta_{\mathrm{a}} < C_{ik}(\boldsymbol{p}_{ik}^{\nu}), \end{cases} \tag{36}$$

with $\zeta_{\mathrm{a}} > 0$, $\delta_{\mathrm{a}} \in (0, \zeta_{\mathrm{a}})$, $C_{ik}(\boldsymbol{p}_{ik}^{\nu}) = \|\boldsymbol{p}_{ik}^{\nu}\|_{xy} - 2R_{\mathrm{a}} - 2S_{\mathrm{a}}$, and $\alpha_{\mathrm{a}} > 0$. This is sufficient to ensure the auxiliary constraint

$$\gamma_1(\boldsymbol{p}_{ik}^{\nu}) : \|\boldsymbol{p}_{ik}^{\nu}\|_{xy} - 2S_{\mathrm{a}} - 2R_{\mathrm{a}} \geq \delta_{\mathrm{a}}. \tag{37}$$

Following from Theorem 1, **(24)**, agent collision can now be avoided by introducing the auxiliary constraint

$$\gamma_2(\boldsymbol{p}_i, \boldsymbol{p}_i^{\nu}) : S_{\mathrm{a}} - \|\boldsymbol{p}_i^{\nu} - \boldsymbol{p}_i\| \geq 0. \tag{38}$$

As shown in **Figure 6**, the combination of **(37)** and **(38)** satisfies **(10)**.

Remark 1. Equations **(35)** and **(36)** assume that agent i knows the difference between its own reference and the reference of agent k. However, the contribution of agent k becomes zero if $\|\boldsymbol{p}_{ik}^{\nu}\|_{xy} \geq \zeta_{\mathrm{a}} + 2R_{\mathrm{a}} + 2S_{\mathrm{a}}$. As a result, it is assumed that agents only share their reference with other agents within an inter-agent distance of $\zeta_{\mathrm{a}} + 2R_{\mathrm{a}} + 4S_{\mathrm{a}}$. A possible option to eliminate communication entirely (i.e. a decentralized approach) is to have each agent measure the position of its neighbors (instead of communicating the applied references) and compute the worst-case references of the neighbors that would still ensure that **(37)** and **(38)** imply **(10)**. This leads to two possible options

$$\boldsymbol{p}_{ik}^{\nu} = \begin{cases} \boldsymbol{p}_k^{\nu} - \boldsymbol{p}_i^{\nu} & i \text{ knows } \boldsymbol{p}_k^{\nu} \\ \boldsymbol{p}_k^{\nu,WC} - \boldsymbol{p}_i^{\nu} = \boldsymbol{p}_k - S_{\mathrm{a}} \dfrac{\boldsymbol{p}_k - \boldsymbol{p}_i^{\nu}}{\|\boldsymbol{p}_k - \boldsymbol{p}_i^{\nu}\|} - \boldsymbol{p}_i^{\nu} & i \text{ knows } \boldsymbol{p}_k, \end{cases}$$

where the latter has the advantage of not requiring inter-agent communication but also leads to a more conservative coordination strategy, as illustrated in **Figure 6**.

7.3 Dynamic Safety Margin

For each agent i its DSM, used in **(21)**, can be obtained by taking the worst case DSM (i.e. the smallest one) of all active saturation (s), wall (w), obstacle (o), and agent collision (a) constraints[2],

$$\Delta_i = \max\left(\min\left(\Delta_i^{\mathrm{s}}, \Delta_i^{\mathrm{w}}, \Delta_i^{\mathrm{o}}, \Delta_i^{\mathrm{a}}\right), 0\right) \geq 0. \tag{39}$$

For the offline design of the DSM we do not rely on explicit trajectory predictions, but use Lyapunov theory and optimization to design the DSM. As such, the following lemma is an important result used throughout this work to compute offline safe threshold values of Lyapunov level sets. As was visualized in **Figure 4**, it guarantees constraint satisfaction if the system dynamics never make its Lyapunov level set value $V(\boldsymbol{x}(t), \boldsymbol{v}(t))$ exceed that threshold value $\Gamma(\boldsymbol{v}(t))$.

Lemma 2. Given a nonlinear pre-stabilized system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{v})$ with state vector $\boldsymbol{x}$, applied reference $\mathbf{v}$, equilibrium point $\bar{\boldsymbol{x}}_{v}$, let

$$V(\boldsymbol{x}, \boldsymbol{v}) = (\boldsymbol{x} - \bar{\boldsymbol{x}}_v)^T \boldsymbol{P} (\boldsymbol{x} - \bar{\boldsymbol{x}}_v), \text{with } \boldsymbol{P} > 0, \tag{40}$$

be a Lyapunov function and let

$$\boldsymbol{c}^T \boldsymbol{x} \leq d(\boldsymbol{v}) \tag{41}$$

be a linear constraint. Then, the Lyapunov treshold value

$$\Gamma(\boldsymbol{v}) = \frac{\left(-\boldsymbol{c}^T \bar{\boldsymbol{x}}_v + d(\boldsymbol{v})\right)^2}{\boldsymbol{c}^T \boldsymbol{P}^{-1} \boldsymbol{c}}, \tag{42}$$

is such that $V(\boldsymbol{x}, \boldsymbol{v}) \leq \Gamma(\boldsymbol{v}) \Rightarrow$ **(41)**.

Proof: See (Nicotra et al., 2019). ■

Since the DSM is computed on a per-agent basis, the agent index i will be omitted for the sake of notational simplicity.

[2] dependency of Δ on $(\boldsymbol{x}_i, \boldsymbol{v}_i)$ is omitted for simplicity of notation.

The following paragraphs address each constraint separately.

7.3.1 Saturation Constraints

In this section we show three strategies to compute a safe threshold value that ensure constraints on at least a subset of the inputs **(5)** are satisfied. The quantitative effects of these three strategies for an input constrained double integrator system are depicted in **Figure 7**.

Traditional Lyapunov Level Set Strategy (Trad Lyap): One practical approach is to consider the outer loop control law and ensure the box constraints on the total thrust are satisfied,

$$T_{\min} \leq T = m\left\|\boldsymbol{K}_P\left(\boldsymbol{p}^v - \boldsymbol{p}\right) - \boldsymbol{K}_D\dot{\boldsymbol{p}} + g\widehat{\boldsymbol{e}}_3\right\| \leq T_{\max}. \tag{43}$$

Since the inequality constraint **(43)** is nonlinear in the outer loop state variables, it is necessary to find a linear constraint that implies **(43)**, in order to apply Lemma 2. A possible approach to provide a linear constraint is to make a distinction between the steady-state thrust $mg\widehat{\boldsymbol{e}}_3$ and the dynamic feedback $m\left(\boldsymbol{K}_P\left(\boldsymbol{p}^v - \boldsymbol{p}\right) - \boldsymbol{K}_D\dot{\boldsymbol{p}}\right)$. For the upper limit of the thrust constraint, this can be done by using the triangular inequality, and we obtain $T \leq m\left\|\boldsymbol{K}_P\left(\boldsymbol{p}^v - \boldsymbol{p}\right) - \boldsymbol{K}_D\dot{\boldsymbol{p}}\right\| + mg$. Hence,

$$\text{if } m\left\|\boldsymbol{K}_P\left(\boldsymbol{p}^v - \boldsymbol{p}\right) - \boldsymbol{K}_D\dot{\boldsymbol{p}}\right\| + mg \leq T_{\max} \Rightarrow T \leq T_{\max}, \tag{44}$$

it is therefore sufficient to ensure that, $\forall \mathbf{e} \in \mathbb{R}^3$:

$$\left[\boldsymbol{K}_P\widehat{\boldsymbol{e}}^T - \boldsymbol{K}_D\widehat{\boldsymbol{e}}^T\right]\begin{bmatrix} \boldsymbol{p}^v - \boldsymbol{p} \\ \dot{\boldsymbol{p}} \end{bmatrix} \leq \frac{T_{\max} - mg}{m}. \tag{45}$$

This is equivalent to limiting the maximum acceleration of the UAV in any direction. The main interest with **(45)** is that it defines a rotationally invariant constraint that is linear for any given unitary vector $\widehat{\boldsymbol{e}}$, which can be expressed in the linear form **(41)** with $\boldsymbol{c} = \left[\boldsymbol{c}_a^T, \boldsymbol{c}_b^T\right]^T$ by choosing $\boldsymbol{c}_a = -\boldsymbol{K}_P\widehat{\boldsymbol{e}}$, $\boldsymbol{c}_b = -\boldsymbol{K}_D\widehat{\boldsymbol{e}}$, and $d\left(\boldsymbol{p}^v\right) = T_{\max} - mg/m - \boldsymbol{K}_P\widehat{\boldsymbol{e}}^T\boldsymbol{p}^v$. Assuming unidirectional gains $\boldsymbol{K}_P = k_P\boldsymbol{I}_3$ and $\boldsymbol{K}_D = k_D\boldsymbol{I}_3$, the associated threshold value **(42)** is,

$$\Gamma_{T_{\max}} = \frac{1}{2}\frac{\left(T_{\max} - mg\right)^2}{m^2}\frac{k_P + \varepsilon\left(1 - \varepsilon\right)k_D^2}{k_P^2 + k_D^2\left(k_P + \varepsilon k_D^2 - 2\varepsilon k_P\right)}. \tag{46}$$

Similarly, $\Gamma_{T_{\min}}$ can be computed by replacing $T_{\max}$ in **(46)** with $T_{\min}$. The DSM that prevents the total thrust to saturate is

$$\Delta^s = \kappa_s\left(\min\left(\Gamma_{T_{\max}}, \Gamma_{T_{\min}}\right) - V\left(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{p}^v\right)\right), \tag{47}$$

with $\kappa_s \in \mathbb{R}_{>0}$.

Optimally Aligned Lyapunov Level Set Strategy (Opt Lyap): This section is an extension of the theory in (Garone et al., 2018) and applies it to higher-order quadrotor dynamics. Since linear systems are characterized by an infinite choice of quadratic Lyapunov functions, a way to improve the performance of the outer loop dynamics is to select the optimal Lyapunov based threshold value that is perfectly aligned with the total thrust constraints, instead of using **(46)**, which is not aligned. Hence, one can find a common Lyapunov function in the quadratic form

$$V_T\left(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{p}^v\right) = \begin{bmatrix} \boldsymbol{p} - \boldsymbol{p}^v \\ \dot{\boldsymbol{p}} \end{bmatrix}^T \boldsymbol{P}_T \begin{bmatrix} \boldsymbol{p} - \boldsymbol{p}^v \\ \dot{\boldsymbol{p}} \end{bmatrix}, \tag{48}$$

with $\boldsymbol{P}_T > 0$ that satisfies the Lyapunov equation $\boldsymbol{A}(\tilde{\boldsymbol{R}})^T\boldsymbol{P}_T + \boldsymbol{P}_T\boldsymbol{A}(\tilde{\boldsymbol{R}}) \leq 0$ and $\boldsymbol{A}(\tilde{\boldsymbol{R}})$ defined in (6.2). By taking advantage of the rotational symmetry of the system and defining

$$\boldsymbol{P}_T = \begin{bmatrix} \widehat{P}_{T,11}\boldsymbol{I}_3 & \widehat{P}_{T,12}\boldsymbol{I}_3 \\ \widehat{P}_{T,21}\boldsymbol{I}_3 & \widehat{P}_{T,22}\boldsymbol{I}_3 \end{bmatrix}, \tag{49}$$

the optimal Lyapunov function can be obtained by solving the following linear matrix inequality

$$\begin{cases} \min \log\det\left(\widehat{\boldsymbol{P}}_T\right) \\ \text{subject to :} \\ \boldsymbol{A}(0)^T\widehat{\boldsymbol{P}}_T + \widehat{\boldsymbol{P}}_T\boldsymbol{A}(0) \leq 0 \\ \boldsymbol{A}(\Delta\alpha)^T\widehat{\boldsymbol{P}}_T + \widehat{\boldsymbol{P}}_T\boldsymbol{A}(\Delta\alpha) \leq 0 \\ \widehat{\boldsymbol{P}}_T \geq \boldsymbol{c}_T\boldsymbol{c}_T^T \end{cases}, \tag{50}$$

where $\tilde{\alpha}$ and $\Delta\alpha$ are the current and the maximum allowed rotational error between $\widehat{\boldsymbol{z}}_{\mathcal{B}}$ and $\widehat{\boldsymbol{z}}_{\mathcal{B}}^d$, $\boldsymbol{c}_T = -m\left[k_P, k_D\right]^T$ and

$$\boldsymbol{A}(\tilde{\alpha}) = \begin{bmatrix} 0 & 1 \\ -k_P\cos(\tilde{\alpha}) & -k_D\cos(\tilde{\alpha}) \end{bmatrix}. \tag{51}$$

Given the quadratic Lyapunov function **(48)**, we obtain the threshold values

$$\Gamma_{T_{\max}} = \frac{\left(T_{\max} - mg\right)^2}{\boldsymbol{c}_T^T\widehat{\boldsymbol{P}}_T^{-1}\boldsymbol{c}_T}, \Gamma_{T_{\min}} = \frac{\left(T_{\min} - mg\right)^2}{\boldsymbol{c}_T^T\widehat{\boldsymbol{P}}_T^{-1}\boldsymbol{c}_T}. \tag{52}$$

The DSM that prevents the total thrust to saturate and is based on the Lyapunov function that is optimally aligned with this constraint, then becomes

$$\Delta^s = \kappa_s\left(\min\left(\Gamma_{T_{\max}}, \Gamma_{T_{\min}}\right) - V_T\left(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{p}^v\right)\right). \tag{53}$$

Optimally Aligned Invariant Level Set Strategy (Opt Inv): A more generic safe set can be obtained by considering the outer loop dynamics **(19)** with input **(12)** and computing offline the threshold value associated to the largest possible optimally aligned Lyapunov level set that satisfies the constraints of the following minimization problem

$$\begin{cases} \Gamma_{T_{\max/\min}} = \min\limits_{\boldsymbol{p},\dot{\boldsymbol{p}},\boldsymbol{p}^v} V_T\left(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{p}^v\right) \\ \text{subject to :} \\ \left\|\boldsymbol{T}^d\left(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{p}^v\right)\right\| = T\left(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{p}^v\right) = T_{\max/\min} \\ \boldsymbol{f}\left(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{p}^v\right)^T\nabla T\left(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{p}^v\right) \geq 0/\leq 0, \end{cases} \tag{54}$$

with the closed position loop dynamics $\boldsymbol{f}\left(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{p}^v\right)$ and the total thrust gradient $\nabla T\left(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{p}^v\right)$. Doing so, one can obtain a safe invariant set by taking the optimally aligned Lyapunov level set and subtracting the inadmissible region, i.e. the region where the constraints are violated $T \geq T_{\max}$ or $T \leq T_{\min}$. The invariant set based DSM can be computed as,

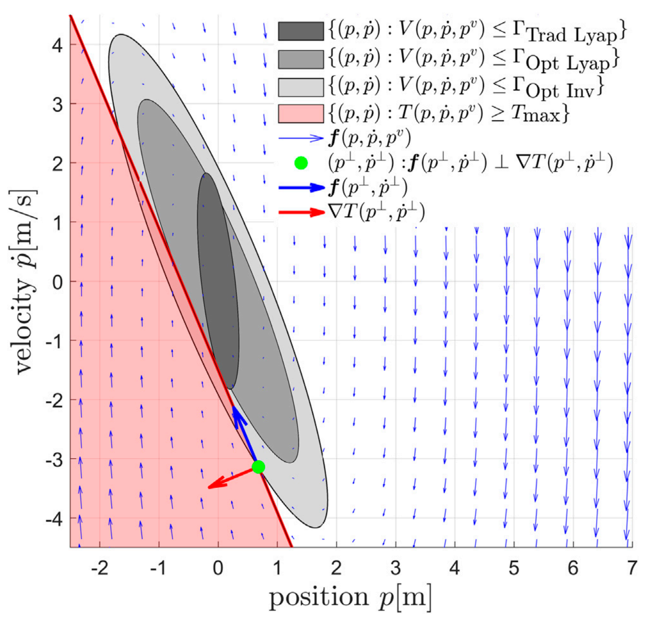


FIGURE 7 | Phase plane representation of the proposed input constraint enforcement strategies, illustrated for a second-order dynamical system $m\ddot p = T - mg$ subject to the pre-stabilizing control law $T = m(k_P(p^v - p) - k_D\dot p + g)$ with $p^v = 0$ and the input constraint $T \le T_{\max}$. The traditional Lyapunov based level-set (dark-grey) yields the most conservative DSM **(47)**. Aligning the level-set with the constraints (medium gray) by solving the offline optimization problem **(50)** drastically increases the certified safe region **(53)**. Further improvements can be obtained by solving **(54)** and using the invariant set which is the set obtained after subtracting the intersection between the light-grey Lyapunov level set and the region violating the input constraint from the light-grey Lyapunov level set. All three sets are certifiably safe since the flow vectors of the closed-loop system (in blue) all point inward. Note that, due to the high values in the first block diagonal of **(17)**, any constraint that only depends on the position error variables, e.g. of the form $p = a$ with $a \in \mathbb{R}$, is already very well aligned under the traditional strategy. Hence, performance benefits from optimal alignment are marginal.

$$\Delta^{s} = \kappa_s \min\left(\frac{\min\left(\Gamma_{T_{\max}},\Gamma_{T_{\min}}\right) - V_T\left(\boldsymbol{p},\dot{\boldsymbol{p}},\boldsymbol{p}^v\right)}{\left(\Gamma_{T_{\max}}+\Gamma_{T_{\min}}\right)/2}, \min_{j\in\{1,2,3,4\}}\left(\frac{U_{\max}-U_j}{(U_{\max}-U_{\min})/2},\frac{U_j-U_{\min}}{(U_{\max}-U_{\min})/2}\right)\right). \tag{55}$$

REMARK 2. To avoid motor saturation when tracking a non-zero yaw reference, it is also necessary to add an ERG on the yaw axis. This can be done using the NF in **(26)** and the DSM

$$\Delta^{s,\psi} = \kappa_{s,\psi} \min_{j\in\{1,2,3,4\}}\left(\frac{U_{\max}-U_j}{(U_{\max}-U_{\min})/2},\frac{U_j-U_{\min}}{(U_{\max}-U_{\min})/2}\right), \tag{56}$$

with $\kappa^{s,\psi} \in \mathbb{R}_{>0}$.

7.3.2 Wall Constraints

The convex inequality constraints **(8)** are equivalent to **(41)** with $\boldsymbol{c} = [\boldsymbol{c}^T_{w_j}, \boldsymbol{0}^T_{3\times1}]^T$, and $d(\boldsymbol{p}^v) = d_{w_j} - R_a$. As a result, the threshold value associated to the j-th wall constraint is

$$\Gamma_{w_j} = \frac{1}{2}\left(k_P + \varepsilon(1-\varepsilon)k_D^2\right)\left(\hat{\boldsymbol{c}}^T_{wj}\boldsymbol{p}^v - d_{w_j} + R_a\right)^2. \tag{57}$$

The dynamic safety margin corresponding to the wall constraint closest to violation then becomes,

$$\Delta^{w} = \kappa_w\left(\min_{j\in\{1,\ldots,N_w\}}\left(\Gamma_{w_j}\right) - V\left(\boldsymbol{p},\dot{\boldsymbol{p}},\boldsymbol{p}^v\right)\right), \tag{58}$$

with $\kappa_w \in \mathbb{R}_{>0}$.

7.3.3 Obstacle Constraints

Constraint **(9)** defines a non-convex admissible region. Given a fixed reference $\boldsymbol{p}^v$, it can be shown using triangular inequalities that

$$\left\|\boldsymbol{p}-\boldsymbol{o}_j\right\| \ge \left\|\boldsymbol{p}-\boldsymbol{p}^v\right\| - \left\|\boldsymbol{p}^v-\boldsymbol{o}_j\right\| \ge R_{o_j} + R_a. \tag{59}$$

As a result, **(9)** can be enforced by simply ensuring

$$\left(\widehat{\boldsymbol{p}^v-\boldsymbol{o}_j}\right)^T\left(\boldsymbol{p}^v-\boldsymbol{p}\right) \ge R_{o_j} + R_a + \left\|\boldsymbol{p}^v-\boldsymbol{o}_j\right\| \ge 0. \tag{60}$$

The inequality constraints define a reference-dependent virtual wall and are equivalent to **(41)** with $\boldsymbol{c} = [(\widehat{\boldsymbol{p}^v-\boldsymbol{o}_j})^T, \boldsymbol{0}^T_{3\times1}]^T$, and $d(\boldsymbol{p}^v) = (\widehat{\boldsymbol{p}^v-\boldsymbol{o}_j})^T\boldsymbol{p}^v - R_{o_j} - R_a - \left\|\boldsymbol{p}^v-\boldsymbol{o}_j\right\|$. The DSM related to this constraint then becomes,

$$\Delta^{o} = \kappa_o\left(\min_{j\in\{1,\ldots,N_o\}}\left(\Gamma_{o_j}\right) - V\left(\boldsymbol{p},\dot{\boldsymbol{p}},\boldsymbol{p}^v\right)\right). \tag{61}$$

with $\kappa_o \in \mathbb{R}_{>0}$.

7.3.4 Agent Collision Avoidance

As explained in **Section 7.2**, collision avoidance can be satisfied by also enforcing the auxiliary constraint **(38)**. Since constraint **(38)** applies equally in every direction in 3D space, it can be enforced using the Lyapunov threshold value associated to the linear constraint

$$\left[\hat{\boldsymbol{e}}^T\ \boldsymbol{0}^T_{3\times1}\right]\begin{bmatrix}\boldsymbol{p}^v-\boldsymbol{p}\\ \dot{\boldsymbol{p}}\end{bmatrix} \le S_a, \forall\hat{\boldsymbol{e}} \in \mathbb{R}^3 : \|\hat{\boldsymbol{e}}\| = 1, \tag{62}$$

thus leading to

$$\Gamma_a = \frac{1}{2}\left(k_P + \varepsilon(1-\varepsilon)k_D^2\right)S_a^2. \tag{63}$$

The DSM related to this constraint then becomes,

$$\Delta^{a} = \kappa_a\left(\Gamma^{a} - V\left(\boldsymbol{p},\dot{\boldsymbol{p}},\boldsymbol{p}^v\right)\right), \tag{64}$$

with $\kappa_a \in \mathbb{R}_{>0}$.

8 RESULTS

We present the first results of an extensive experimental validation of the ERG and the D-ERG frameworks by means of single and multi-robot hardware experiments (a video of the experiments can be found at https://youtu.be/le6WSeyTXNU) using the experimental setup described hereafter. In a comparative simulation campaign we have analyzed statistically the goal and constraint satisfaction properties of our methodology. A summary of these results can be found in **Section 9**.

8.1 Experimental Setup

The experiments are performed using Crazyflie 2.1 nano-quadrotors in a Vicon motion capture system for indoor localization based on the Crazyswarm system architecture of (Preiss et al., 2017b). The computationally efficient control and navigation layers of **Sections 6** and **7** are implemented in C and run at 500 Hz on-board the Crazyflie's STM32F4 microprocessor's firmware. The only programs running on the ground station are the special purpose motion capture tracker (Preiss et al., 2017b), a code for sending goal configurations to each quadrotor, and a code that mimics local communications between agents. Each UAV sends and receives new goal and feedback signals (i.e. the agent's own state and neighbor information) via Crazyradios PA at 100 Hz. An on-board Kalman filter updates the agent's own states at a higher rate than the motion capture system, but for the neighbor information such a Kalman filter update is not present. The experiment data is logged on-board the quadrotors on micro SD cards.

Each UAV is modeled with a static safety radius of $R_a = 0.08$ m and a mass of approximately 34.6 g. Its inertia matrix $\boldsymbol{J} = \text{diag}(17.31, 17.94, 33.75) \cdot 10^{-6}\,\text{kgm}^2$ is calculated from a CAD model and is only used to estimate the actuator torque constant. The estimated actuator thrust and torque constants amount $K_T = 0.012\,\text{N/V}^2$ and $K_\tau = 6.84 \cdot 10^{-6}\,\text{Nm/V}^2$, respectively. The nominal distance between the motor axis and the center of mass of the aircraft amounts $d = 4.65$ cm.

8.2 Tuning Guidelines

Here, we list guidelines for the tuning of the main parameters of the control and navigation layer and how this relates to the obtained performance and robustness. We advise users of this approach to tune the parameters in the order as they are listed below and to start with the input saturation constraints, followed by static and dynamic obstacle constraints.

(1) First tune the inner loop gains $\boldsymbol{K}_R, \boldsymbol{K}_\omega > 0$ and then the outer loop gains $\boldsymbol{K}_P$, $\boldsymbol{K}_D > 0$ for stable regulation control performance. The outer loop's settling time should be an order of magnitude slower than the one of the inner loop. This step is accomplished without worrying about the effect on any of the input or state constraints. The stiffer the pre-stabilized closed-loop system is tuned, the more the agents can be stacked in a smaller volume, at the cost of a more precise and higher rate odometry.

(2) Eliminate numerical noise in the attraction field by selecting a strictly positive, but small value for the smoothing radius η.

(3) Increase the DSM gains κ until no further performance increase is obtained. These gains are chosen such that the DSMs of the active constraints have the same order of magnitude.

(4) Choose medium influence margins ζ defining from how far the obstacles are considered in the repulsion field. Too large values will require too large sensing ranges for static obstacles or communication ranges for dynamic obstacles, whereas too low values do not give enough reaction time.

(5) For cooperative agent collision avoidance, choose the maximum position error radius S_a. The larger this value, the higher the maximum attainable robot's speed, but the larger the distance traveled by each agent to reach its goal.

(6) Select small circulation gains α around obstacles and agents to avoid robots getting stuck in local saddle points. Too large values tend to increase the settling time.

(7) Choose strictly positive static safety margins δ to increase robustness. This also ensures the NF's repulsion term achieves its maximum amplitude while the DSM stays strictly positive. Hence this allows moving (and not blocking) the reference in directions pointing outward the obstacle constraint.

In all the experiments, the control gains of the inner-outer loop control law detailed in **Section 6** are $\boldsymbol{K}_P = 13.0\,\boldsymbol{I}_3, \boldsymbol{K}_D = 5.0\,\boldsymbol{I}_3$, $\boldsymbol{K}_R$ = diag(0.005, 0.005, 0.0003), and $\boldsymbol{K}_\Omega$ = diag(0.001, 0.001, 0.00005), which give moderately aggressive performance. The attraction field of the navigation layer is chosen with $\eta = \eta_\psi = 0.005$. Other parameters defined in **Section 7** are specified in the following sections.

8.3 Single Aerial Robot Experiments

8.3.1 Point-to-Point Transitions – Input Constraints

In the accompanying video we show that point-to-point transitions can easily destabilize a pre-stabilized quadrotor due to actuator saturation when the changes in $\boldsymbol{p}^v$ become too abrupt.

The goal of the experiments is to validate the theory of **Section 7.3.1** by showing that the navigation layer ensures safety for whatever $\boldsymbol{p}^r$ and to quantify the difference in performance of the three strategies used to compute the DSM. To do so, we sequentially performed the following three experiments with a quadrotor where the navigation layer ensures input constraints satisfaction with $U_{\min} = 0.0$ V, $U_{\max} = 3.5$ V or $T_{\min} = 0.0$ N, $T_{\max} = 0.59$ N by using either:

- Trad Lyap: traditional Lyapunov-based DSM **(47)**, with $\kappa_s = 2.5$ and $\varepsilon = 0.5$;
- Opt Lyap: optimally aligned Lyapunov-based DSM **(53)**, with $\kappa_s = 9.45$, $\Delta\alpha = 0.349$, $\widehat{P}_{T,11} = 0.8810$, $\widehat{P}_{T,12} = \widehat{P}_{T,21} = 0.3202$, $\widehat{P}_{T,22} = 0.1511$, $\Gamma_T = \min(\Gamma_{T_{\max}}, \Gamma_{T_{\min}}) = 0.266$;
- Opt Inv: invariance-based DSM **(55)**, with $\kappa_s = 1.80$, $\Gamma_T = 3.00$.

As is depicted in **Figure 8**, in each of these experiments, the UAV starts from the initial hovering position $\boldsymbol{p}(t \le 0.5\,\text{s}) = [4.0, 1.0, 0.25]^T$ m, i.e. $\dot{\boldsymbol{p}}(t \le 0.5\,\text{s}) = 0$ m/s. At $t = 0.5$ s and at $t = 12.5$ s it is asked to transition between the points $\boldsymbol{p}^r(0.5\,\text{s} \le t < 12.5\,\text{s}) = [0.50, -1.0, 2.50]^T$m and $\boldsymbol{p}^r(t \ge 12.5\,\text{s}) = [4.0, 1.0, 1.25]^T$ m.

The desired position set-point is always reached in a stable and safe (i.e. DSM ≥ 0) manner. As expected from the theory in **Section 7.3.1**, a large reduction in settling time and an increase in the peak velocity is obtained when passing from a traditional

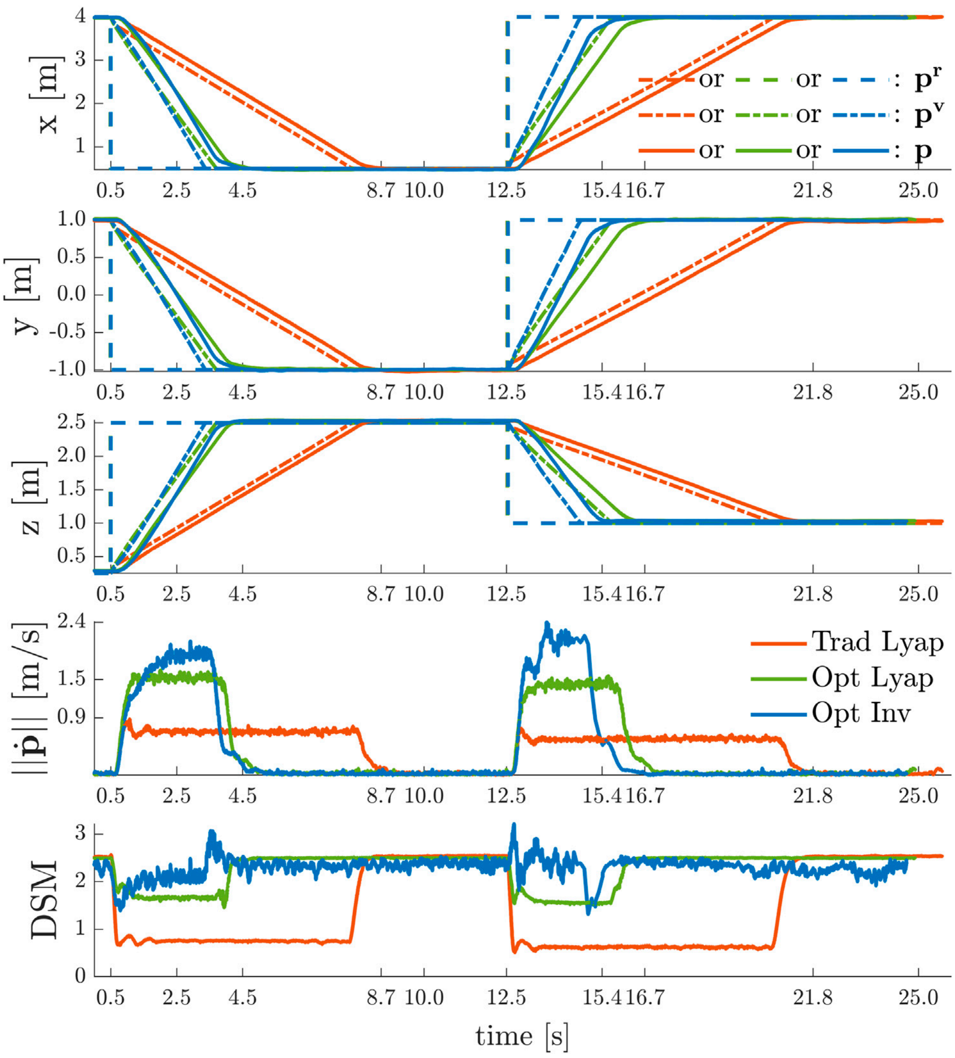


FIGURE 8 | Point-to-Point Transitions Without Violation of Input Constraints – The three strategies for computing DSMs give provably stable and safe performance. The traditional Lyapunov strategy is the most conservative one, whereas the invariance based strategy outperforms the other two in terms of settling time and peak velocity.

Lyapunov based strategy, to the optimally aligned Lyapunov based strategy, and finally to the optimally aligned invariance based strategy. The latter gives the most aggressive performance and allows the aerial vehicle to obtain peak velocities of 2.4 m/s, which is about 2.76 times larger than what is obtained with the traditional Lyapunov based strategy. Note that the values of κ for these three cases where chosen such that the value of the DSMs are equal during hovering, i.e. when $t \in [0.0, 0.5]$ s, or $t \in [8.7, 12.5]$ s, or $t \in [21.8, \infty)$ s.

To show the effect of time-varying yaw angle references, we sequentially performed the following two experiments with the quadrotor using the invariance based ERG on the total thrust constraints and using either:

- no ERG on the yaw axis ψ;
- an ERG on yaw axis ψ as in **(56)** with $\kappa_{s,\psi} = 1.80$.

In each of these experiments, depicted in **Figure 9**, the UAV starts from the initial position $\boldsymbol{p}(t \leq 1.0\,\text{s}) = [4.0, 1.0, 0.25]^T$ m while hovering. At $t = 1.0$ s and at $t = 6.0$ s it receives the same position step references as in the previous experiment, but simultaneously it also receives yaw step references between 0° and 120° (No ERG on ψ), and between $-90°$ and 270° (ERG on ψ).

In the absence of an ERG on the yaw axis, the system remains stable under severe actuator saturation for the simultaneous position and yaw commands given at $t = 1.0$ s but becomes unstable for the commands given at $t = 6.0$ s. On the other hand, the system displays a stable, safe, and aggressive behavior during the whole experiment when the ERG is also applied to the yaw axis.

8.3.2 Point-to-Point Transitions – Wall Avoidance

The results depicted in **Figure 10** show the aerial vehicle avoiding two virtual walls with $\boldsymbol{c}_{w_1} = [1, 0, 0]^T$ m, $d_{w_1} = 4.8$ m, and

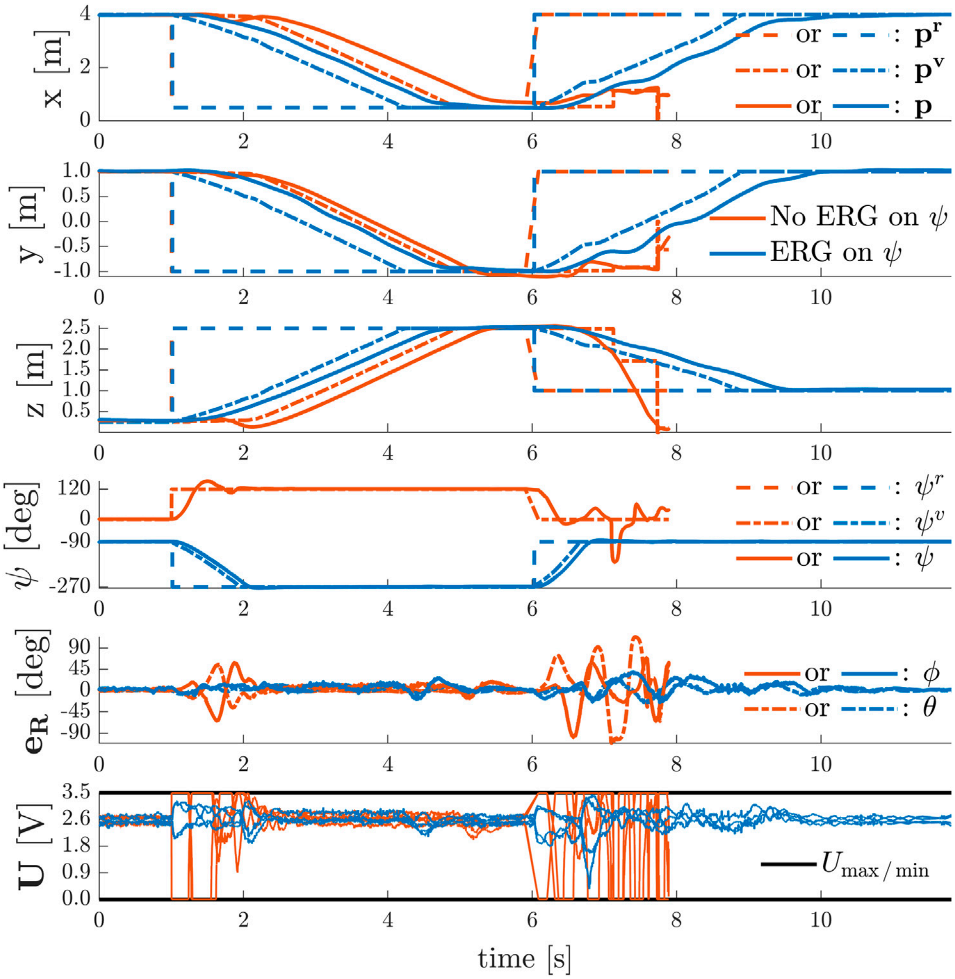


FIGURE 9 | Simultaneous Point-to-Point Transitions and Discontinuous Yaw References With and Without Violation of Input Constraints – Besides an ERG on the position variables that limits the total thrust, an invariance based ERG on the yaw axis is required to ensure safety for non-stationary yaw references.

$\boldsymbol{c}_{\mathrm{w}_2} = [0, -1, 0]^T$ m, $d_{\mathrm{w}_2} = 2.0$ m, when using an ERG with an invariance based DSM for the input constraints and a Lyapunov based DSM for the convex wall constraints with $\kappa_{\mathrm{w}} = 1.5$, $\zeta_{\mathrm{w}} = 1.0$ m, and $\delta_{\mathrm{w}} = 0.01$ m. The UAV is initially hovering at $[4.0, 1.0, 0.25]^T$ m and is commanded consecutively to the positions $[1.5, -2.5, 1.5]^T$ m, $[5.5, -2.5, 1.5]^T$ m, and $[4.0, 1.0, 1.0]^T$ m. From the logged data one can see that the quadrotor initially speeds up to a maximum speed of 2.0 m/s, and slows down such that overshoots do not cause collisions with the virtual walls. One can also see that the NF is designed such that it handles steady-state inadmissible references, which are depicted by black dots outside of the convex region in **Figure 10**.

8.4 Multiple Aerial Robots Experiments

In these experiments the UAVs are modeled as cylinders as detailed in **Section 4.2.4**, preventing them to fly over each other. Similarly to (Preiss et al., 2017a; Honig et al., 2018; Vukosavljev et al., 2019), this choice prevents a MAV's propeller downwash effect to destabilize other MAVs which are flying closely underneath.

8.4.1 Provably Safe Human-Swarm Teleoperation

In this experiment we show that the D-ERG ensures a swarm of $N_{\mathrm{a}} = 4$ quadrotors can be teleoperated by a human in a provably safe way within a confined environment composed of wall constraints with $\boldsymbol{c}_{\mathrm{w}_1} = [-1, 0, 0]^T$ m, $d_{\mathrm{w}_1} = 3$ m, $\boldsymbol{c}_{\mathrm{w}_2} = [1, 0, 0]^T$ m, $d_{\mathrm{w}_2} = 4.8$ m, $\boldsymbol{c}_{\mathrm{w}_3} = [0, -1, 0]^T$ m, $d_{\mathrm{w}_3} = 2.0$ m, and $\boldsymbol{c}_{\mathrm{w}_4} = [0, 1, 0]^T$ m, $d_{\mathrm{w}_4} = 1.5$ m. We use the same ERG parameters as in **Section 8.3.2**, and for the collision avoidance between agents, we exchange $\boldsymbol{p}^v$ between the agents and use $S_{\mathrm{a}} = 0.80$ m, $\alpha_{\mathrm{a}} = 0.0$, $\zeta_{\mathrm{a}} = 1.50$ m, $\delta_{\mathrm{a}} = 0.01$ m, and $\kappa_{\mathrm{a}} = 50.0$. The human operator accelerates and decelerates the motion capture calibration wand fast in 3D space such as to exploit the quadrotor dynamics. Each agent is tasked to yaw in the direction of the wand and follows its relative position displacement. The logged data is depicted in **Figure 11**. The requested aggregated reference that wants to keep the swarm in

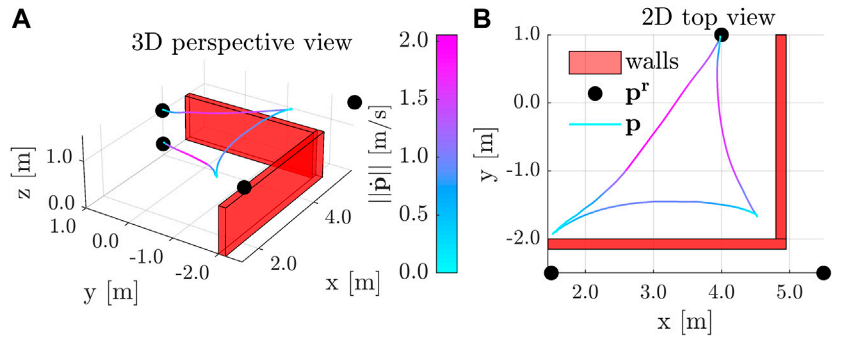


FIGURE 10 | Point-to-Point Transitions with Wall Avoidance – The quadrotor achieves top speeds of 2 m/s and slows down as to avoid wall collisions, even when the position references are steady-state inadmissible.

a rigid square formation is deformed by the navigation layer by decreasing the rate of change of the reference applied to each pre-stabilized agent when it comes closer to violations of input, wall, or agent collision constraints. One can see that around $t = 20.0$ s, there are short periods where the actuator inputs come very close to their upper and lower limits and Δ_i^s is very close to zero, such that the applied reference is kept almost constant.

8.4.2 Point-to-Point Transitions – Agent Collision Avoidance

In **Figure 12** the results of two experiments with a swarm of $N_a = 5$ agents are depicted. Every agent is commanded to transition between specific goal positions at $t = 1.0$ s and at $t = 26.0$ s, such that if the agents are coordinated effectively, this globally leads to a line formation for the swarm. Moreover, they have to stay inside a confined environment bounded by four walls with

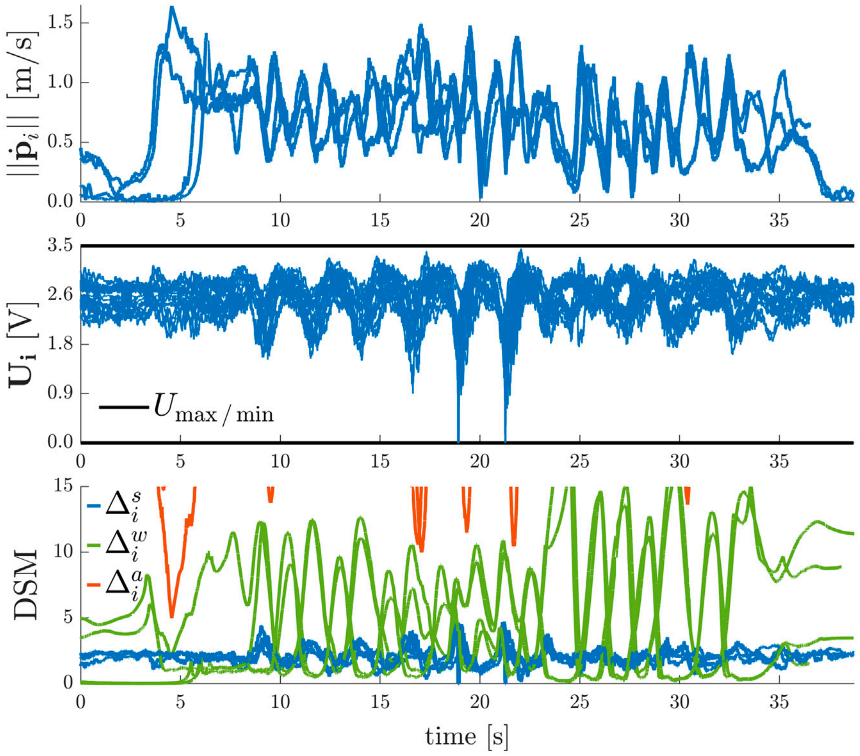


FIGURE 11 | Results of the Human-Swarm Teleoperation Experiment in a Confined Environment – The D-ERG ensures the safe coordination of the quadrotor team formation. During the short periods where the actuator inputs come very close to their upper and lower limits (around $t = 20.0$ s), the DSM decreases rapidly such that the applied reference is kept almost constant. Note that the steady-state motor voltages during hovering after $t > 37.0$ s vary in a range of 2.1 V to 3.0 V. This is caused by variability in model parameters (e.g. battery displacements from the MAV's center of mass, different motor-propeller constants) and shows the robustness of the overall approach to model uncertainty.

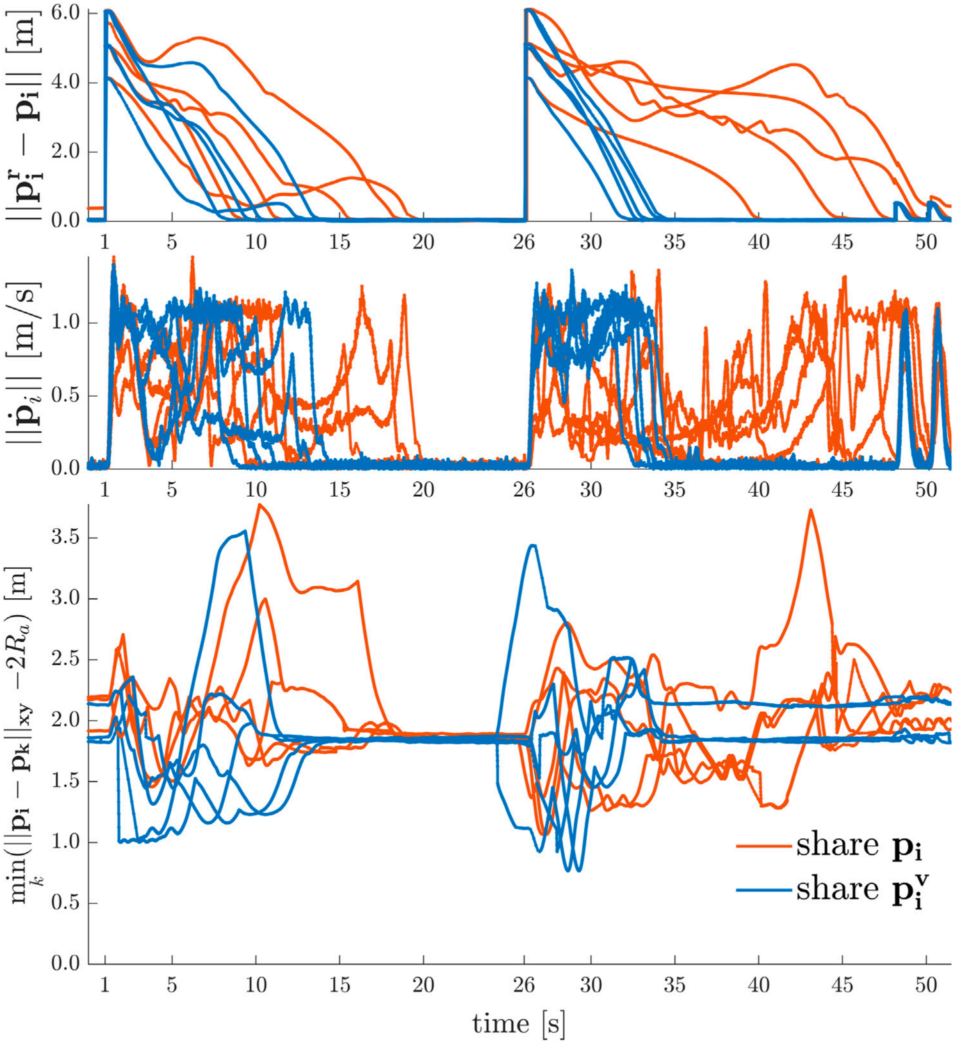


FIGURE 12 | 2D Point-to-Point Transitions with Agent Collision and Deadlock Avoidance – Exchanging the applied reference position $\boldsymbol{p}_i^v(t)$ over the actual position $\boldsymbol{p}_i(t)$ leads to a denser swarm, less distance traveled, and smaller settling times.

$\boldsymbol{c}_{\mathrm{w}_1} = [-1, 0, 0]^T$ m, $d_{\mathrm{w}_1} = 4.8$ m, $\boldsymbol{c}_{\mathrm{w}_2} = [1, 0, 0]^T$ m, $d_{\mathrm{w}_2} = 4.8$ m, $\boldsymbol{c}_{\mathrm{w}_3} = [0, -1, 0]^T$ m, $d_{\mathrm{w}_3} = 2.0$ m, and $\boldsymbol{c}_{\mathrm{w}_4} = [0, 1, 0]^T$ m, $d_{\mathrm{w}_4} = 1.5$ m. The navigation layer consists of a D-ERG using the parameters as in **Section 8.4.1**, but with $S_a = 0.55$ m, $\delta_a = 0.1$ m, $\alpha_a = 0.1$. The same navigation task is performed twice, first by sharing $\boldsymbol{p}$ and then by sharing $\boldsymbol{p}^v$ locally between the agents, as detailed in Remark 1. The results clearly show the D-ERG ensures every agent asymptotically reaches its desired position while avoiding collisions with other agents and the small circulation gain ensures the agents to not get stuck in local saddle points. Comparing the two cases one can see that sharing $\boldsymbol{p}^v$ reduces the worst-case settling time over all agents for transitioning between formations by a factor of two. This is because the swarm remains more dense and the agents have to travel less distance. A potential drawback of the latter is that this explicitly requires communication between the agents, whereas sharing $\boldsymbol{p}$ could be communication-free (i.e. decentralized) if the agents would be equipped with sensors to measure inter-agent position vectors.

Similar to the 2D line formation experiments, **Figure 13** depicts the results of formation transitioning experiments in 3D with a swarm of $N_a = 9$ agents. The actual applied reference positions $\boldsymbol{p}_i^v$ between the agents are exchanged with $S_a = 0.25$ m, $\delta_a = 0.01$ m, and $\alpha_a = 0.2$. Every agent is commanded to some set-point goal positions at $t = 3.0$ s, at $t = 28.0$ s, at $t = 53.0$ s, at $t = 71.0$ s, and at $t = 89.0$ s that must be reached without causing any undesirable agent interactions such as collisions or deadlocks. Moreover the agents stay inside a confined environment bounded by the same four wall constraints. Note that since in this experiment S_a is smaller than in **Figure 12**, this leads here to smaller peak velocities, but a more dense swarm (agents coming as close as

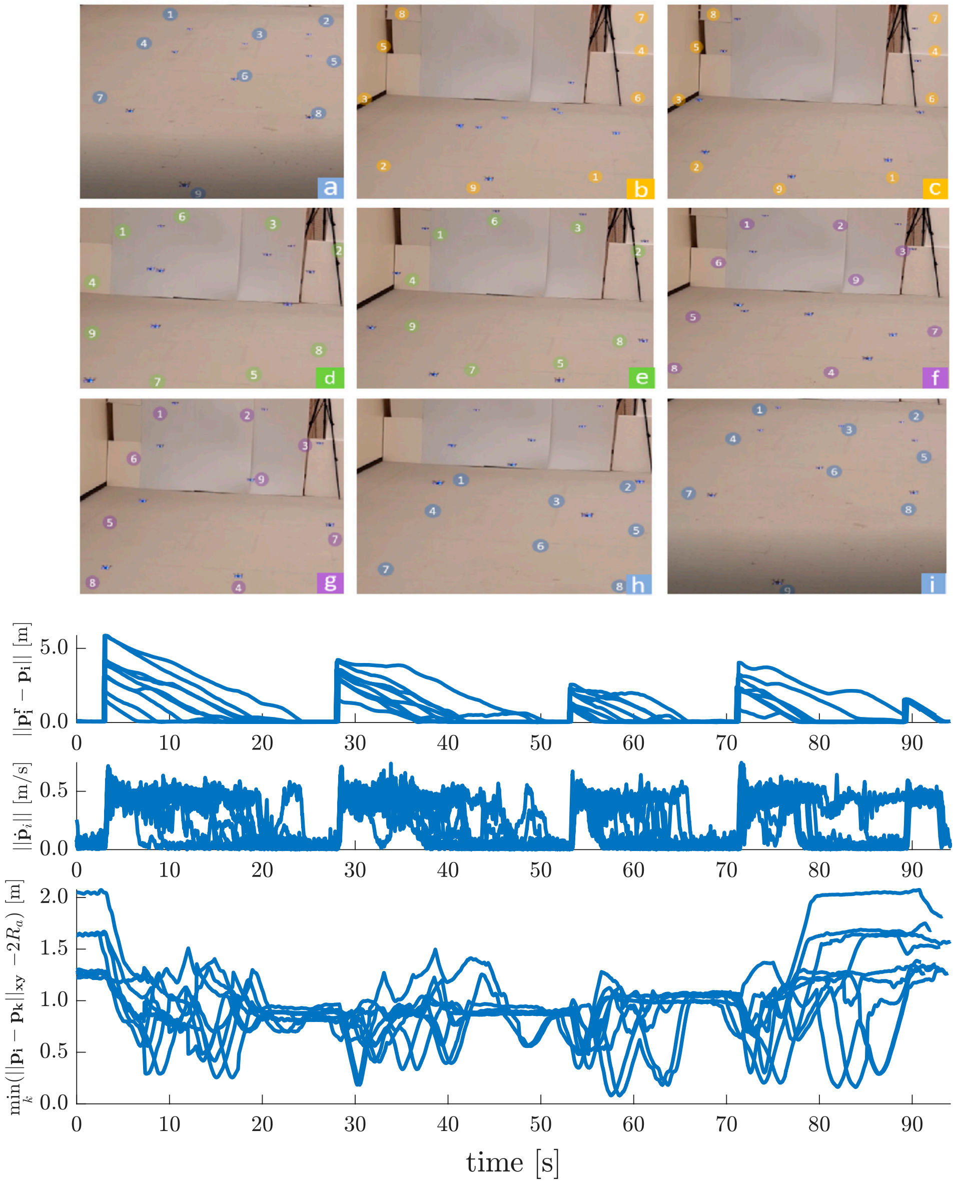


FIGURE 13 | 3D Point-to-Point Transitions with Agent Collision Avoidance – Asymptotically stable, collision free consecutive formations of the initials of the University of Colorado Boulder (UCB) are made. Nine consecutive shots **(a–i)** show the swarm members safely navigating from an initial configuration (in blue, shot a), to the U configuration (in yellow, shot c), to the C configuration (in green, shot e), to the B configuration (in purple, shot g), and finally back to the initial configuration (in blue, shot i).

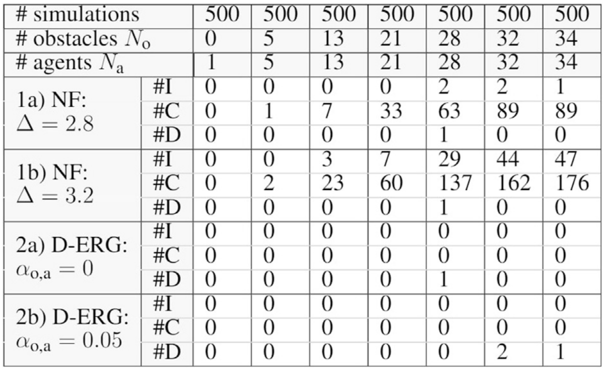
15 cm), since the agents have to travel less distance to avoid each other.

8.5 Analysis of Safety and Goal Satisfaction Certificates

In this simulation study we show some relevant statistics on the occurrence of constraint violations or deadlocks and compare the D-ERG with another optimization-free (i.e., closed form or explicit) approach solely based on attractive and repulsive Navigation Fields. The latter method is implemented by using the NF of **Section 7.2** and by setting the DSM, which is a dynamic state-dependent and reference-dependent gain, to a user-tuned constant value. The latter can be interpreted as a fixed reference filter gain, which can only be selected before executing an experiment.

The results on safety and goal satisfaction for 3D point-to-point transitions of quadrotors in an increasingly densely filled environment with static obstacles and dynamic agents are depicted in **Table 2**. We use a cubic environment with side

TABLE 2 | Simulation Statistics on Safety and Goal Satisfaction – A cubic environment is randomly and increasingly densely filled with N_o spherical obstacles and N_a spherical agents doing 3D point to point transitions. The number of simulations that contain at least one instability, collision, or deadlock are denoted by #I, #C, and #D, respectively. Almost global asymptotic stability, with no collisions and no instabilities confirm the strong safety certificates of the D-ERG. This compared with another explicit approach solely based on Navigation Fields (NF).



# simulations		500	500	500	500	500	500	500
# obstacles N_o		0	5	13	21	28	32	34
# agents N_a		1	5	13	21	28	32	34
1a) NF: $\Delta = 2.8$	#I	0	0	0	0	2	2	1
	#C	0	1	7	33	63	89	89
	#D	0	0	0	0	1	0	0
1b) NF: $\Delta = 3.2$	#I	0	0	3	7	29	44	47
	#C	0	2	23	60	137	162	176
	#D	0	0	0	0	1	0	0
2a) D-ERG: $\alpha_{o,a} = 0$	#I	0	0	0	0	0	0	0
	#C	0	0	0	0	0	0	0
	#D	0	0	0	0	1	0	0
2b) D-ERG: $\alpha_{o,a} = 0.05$	#I	0	0	0	0	0	0	0
	#C	0	0	0	0	0	0	0
	#D	0	0	0	0	0	2	1

lengths of 16 m which is symmetrically centered in the origin. For each simulation we randomly place N_o static spherical obstacles with $R_o = 0.8$ m, $\zeta_o = 1$ m, $\kappa_o = 20$, and the initial and goal positions of N_a quadrotors with $\zeta_a = 1$ m, $S_a = 1.2$ m, $\kappa_a = 20$, $\kappa_s = 6$, that exchange $\boldsymbol{p}^v$ with their neighbors. This random placement is done under the condition that none of the influence margins are overlapping in steady-state. Hence, the swarm's initial and desired position is at least steady-state admissible and convergence to the desired position of each agent can be detected as a static final error at the end of the simulation. For each defined combination of N_o obstacles and N_a agents, 500 random simulations are performed for each of the settings 1a), 1b), 2a), and 2b) depicted in **Table 2**. When there is at least one instability, one collision, or one deadlock detected in a simulation, the respective counters are incremented by one.

The strong safety certificates obtained when employing the D-ERG method are clear from the simulation data summarized in **Table 2**. The occurrence of instabilities and collisions is zero for the certified safe D-ERG, whereas for the Navigation Field (NF) method the occurrence is considerably large. When the constant reference gain in the NF approach is increased from $\Delta = 2.8$ to $\Delta = 3.2$, this leads to a larger number of collisions and instabilities due to severe control input saturation. For fair comparison, these DSM values were chosen around $\Delta = 2.9$, which is the steady-state value of the DSM in the D-ERG when a UAV hovers far away from obstacles.

For what concerns the goal satisfaction certificates, we observe almost global asymptotic stability. The statistical occurrence of deadlocks is almost negligible and only becomes measurable for very densely filled environments cluttered with agents and obstacles. Although a non-zero circulation gain ensures that pairs of agents cannot get stuck in local-saddle points, one can see that there is little benefit in using a circulation gain with a large number of agents. For some simulations it helps to avoid a deadlock, whereas in other simulations it can cause agents to get stuck in a local minimum. However, it is worth noting that this limitation is a consequence of the proposed NF and is not inherent to the D-ERG framework.

9 DISCUSSION

In **Section 8**, we presented an extensive set of experimental and simulation studies of the proposed ERG and Distributed ERG framework, with the first real-world experiments to be found in the literature. These studies demonstrate the following key results (R) when applied to a homogeneous swarm of cooperative Crazyflie 2.1 quadrotors:

- R1: Computational efficiency allows high-rate real-time (500 Hz) computation of control commands on-board small UAVs with severely constrained CPU and RAM;
- R2: Almost globally asymptotically stable control performance for arbitrary position and yaw references (e.g. point-to point transitions or human-swarm teleoperation scenarios) for swarms in constrained environments. The measured statistical deadlock occurrence is negligible;
- R3: Provable safety under actuator inputs and state constraints, including collision avoidance between dynamical agents, and between agents and static obstacles;
- R4: Robustness in the presence of real-world uncertainties (e.g. non-modeled inner loop dynamics, variability of thrust and torque constants or battery voltages, battery displacement from center of mass, sensor noise, communication delays). The low-level control layer is proven to be robust to small attitude errors. Moreover, the D-ERG leverages the robustness of low-level controllers and maintains this property. Since the D-ERG's DSMs itself relies on level-sets (i.e. Lyapunov or invariant set-based) and not on explicit state and input trajectory predictions to obtain safety guarantees, the overall approach is less model dependent and hence more robust;
- R5: Planner or reference agnostic safety certification with the ability to handle steady-state inadmissible references;
- R6: Offline ERG design strategies for the selection of safe threshold values to Lyapunov level-sets can lead to significant improvements in the control performance over traditional methods. Especially when the level sets are aligned with the constraints or when the more generic invariant safe sets are used with negligible increase of the on-board computational requirements.
- R7: The local nature of the D-ERG makes the algorithm scale very well with the number of agents. The distributed formulation that relies on local inter-agent distance and direction in applied reference positions (i.e. requiring agent communication) can lead to significantly smaller settling times and a denser swarm when compared to the decentralized formulation relying on inter-agent distance and direction in positions (i.e. requiring communication or exteroceptive sensing).

In future work, the proposed model-based add-on scheme can be further extended and combined with other control approaches, such as the adaptive control laws to deal with e.g. unmodeled dynamics, actuator deadzones as in (Wang et al., 2019; Yang et al., 2021a), and unavailable velocity measurements as in (Yang et al., 2021b) due to noisy low-cost sensors.

10 CONCLUSION

In this article we formulated the theory of a provably safe distributed constrained control framework, i.e., the Distributed Explicit Reference Governor (D-ERG), and demonstrated its efficacy on a homogeneous swarm of collaborative nano-quadrotors (i.e., a swarm of palm-sized Crazyflies 2.1) through multiple hardware and simulation experiments.

This approach has the following merits. Safety is guaranteed for agents with higher-order dynamics and with a large set of hard constraints such as the four actuator input limits and static and dynamic collision avoidance constraints. In contrast to optimization-based control schemes, this algorithm has a low cost of computation and memory and runs in real-time at a 500 Hz rate on-board the limited available robot hardware. Thereby, its local and reactive nature provides a good scalability to a large number of robots and obstacles. Since this add-on scheme only requires a pre-stabilized plant, it can be of great practical use when the controller is not accessible or not allowed to be changed, which is very often the case for commercial UAV flight control units. Its simple yet effective design makes it an interesting method for industrial robotic applications requiring safe real-time control systems.

However, some limitations still exist and can be addressed in future work. Since the Dynamic Safety Margin uses a single scalar to change the amplitude of the applied reference signal in the direction of the Navigation Field, the performance would reduce when applying this technique to systems with an increased state space dimension. Also, this robust level-set based D-ERG approach comes at the cost of an increased level of conservatism compared to approaches where the future trajectory is explicitly predicted or optimized for. Although the statistical occurrence of deadlocks is very low, the employed Navigation Field does not formally guarantee the absence of deadlocks.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

BC and MN contributed to the conception and formulation of the theory behind the D-ERG. BC was involved in designing and performing all the numerical simulations and hardware validations on the aerial robot swarm. KM assisted with the hardware validations and debugging the numerical simulations. BC wrote the first draft of the manuscript, all authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

This work was supported by the Research Foundation Flanders (FWO) under grant numbers 37472, 60523, and 62062, by the Flemish Government under the program “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen”, and by the US government’s National Science Foundation award CMMI 1904441.

ACKNOWLEDGMENTS

The authors would like to thank the Research and Engineering Center for Unmanned Vehicles (RECUV) of the University of Colorado Boulder for providing continued access to the motion capture system used for the experimental validation, even in the midst of moving to a new campus.

REFERENCES

Alonso-Mora, J., Naegeli, T., Siegwart, R., and Beardsley, P. (2015). Collision Avoidance for Aerial Vehicles in Multi-Agent Scenarios. *Autonomous Robots* 39, 101–121. doi:10.1007/s10514-015-9429-0

Bajcsy, A., Herbert, S. L., Fridovich-Keil, D., Fisac, J. F., Deglurkar, S., Dragan, A. D., et al. (2019). A Scalable Framework for Real-Time Multi-Robot, Multi-Human Collision Avoidance. *In International Conference on Robotics and Automation (IEEE)*, 936–943. doi:10.1109/ICRA.2019.8794457

Bayram, H., Stefas, N., Engin, K. S., and Isler, V. (2017). Tracking Wildlife with Multiple UAVs: System Design, Safety and Field Experiments. in Proceedings of the IEEE International Symposium on Multi-Robot and Multi-Agent Systems. Los Angeles, CA, December 4–5, 2017, 97–103. doi:10.1109/MRS.2017.8250937

Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm Robotics: a Review from the Swarm Engineering Perspective. *Swarm Intelligence* 7, 1–41. doi:10.1007/s11721-012-0075-2

Brockers, R., Hummenberger, M., Weiss, S., and Matthies, L. (2014). Towards Autonomous Navigation of Miniature UAV. in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. Washington, DC: IEEE Computer Society, 645–651. doi:10.1109/CVPRW.2014.98

Campos-Macías, L., Gómez-Gutiérrez, D., Aldana-López, R., De La Guardia, R., and Parra-Vilchis, J. I. (2017). A Hybrid Method for Online Trajectory Planning of Mobile Robots in Cluttered Environments. *IEEE Robotics Automation Lett.* 2, 935–942. doi:10.1109/LRA.2017.2655145

Cappo, E. A., Desai, A., Collins, M., and Michael, N. (2018a). Online Planning for Human-Multi-Robot Interactive Theatrical Performance. *Autonomous Robots* 42, 1771–1786. doi:10.1007/s10514-018-9755-0

Cappo, E. A., Desai, A., and Michael, N. (2018b). Robust Coordinated Aerial Deployments for Theatrical Applications Given Online User Interaction via Behavior Composition. in *Distributed Autonomous Robotic Systems*. 6 edn. Cham, Switzerland: Springer, 665–678. doi:10.1007/978-3-319-73008-0_46

Carrio, A., Tordesillas, J., Vemprala, S., Saripalli, S., Campoy, P., and How, J. P. (2020). Onboard Detection and Localization of Drones Using Depth Maps. *IEEE Access* 8, 30480–30490. doi:10.1109/access.2020.2971938

Chamanbaz, M., Mateo, D., Zoss, B. M., Tokić, G., Wilhelm, E., Bouffanais, R., et al. (2017). Swarm-Enabling Technology for Multi-Robot Systems. *Front. Robotics AI* 4, 12. doi:10.3389/frobt.2017.00012

Chen, M., Hu, Q., Fisac, J. F., Akametalu, K., Mackin, C., and Tomlin, C. J. (2017). Reachability-based Safety and Goal Satisfaction of Unmanned Aerial Platoons on Air Highways. *J. Guidance, Control Dyn.* 40, 1360–1373. doi:10.2514/1.G000774

Chung, S.-J., Paranjape, A., Dames, P., Shen, S., and Kumar, V. (2018). A Survey on Aerial Swarm Robotics. *IEEE Trans. Robotics* 34, 837–855. doi:10.1109/TRO.2018.2857475

Coppola, M., McGuire, K. N., De Wagter, C., and de Croon, G. C. (2020). A Survey on Swarming With Micro Air Vehicles: Fundamental Challenges and Constraints. *Front. Robotics AI* 7, 18. doi:10.3389/frobt.2020.00018

Desaraju, V. R., and Michael, N. (2018). Efficient Prioritization in Explicit Adaptive NMPC through Reachable-Space Search. in AIAA Guidance, Navigation, and Control Conference, Kissimmee, FL, January 8–12, 2018, 1–19. doi:10.2514/6.2018-1847

Desaraju, V. R., Spitzer, A. E., O'Meadhra, C., Lieu, L., and Michael, N. (2018). Leveraging Experience for Robust, Adaptive Nonlinear MPC on Computationally Constrained Systems with Time-Varying State Uncertainty. *Int. J. Robotics Res.* 37, 1690–1712. doi:10.1177/0278364918793717

Ding, W., Gao, W., Wang, K., and Shen, S. (2019). An Efficient B-Spline-Based Kinodynamic Replanning Framework for Quadrotors. *IEEE Trans. Robotics* 35, 1287–1306. doi:10.1109/TRO.2019.2926390

Du, X., Luis, C. E., Vukosavljev, M., and Schoellig, A. P. (2019). Fast and in Sync: Periodic Swarm Patterns for Quadrotors. in Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, May 20–24, 2019, 9143–9149. doi:10.1109/ICRA.2019.8794017

Fathian, K., Safaoui, S., Summers, T. H., and Gans, N. R. (2019). Robust 3D Distributed Formation Control with Collision Avoidance and Application to Multirotor Aerial Vehicles. in Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, May 20–24, 2019, 9209–9215. doi:10.1109/ICRA.2019.8794349

Fathian, K., Summers, T. H., and Gans, N. R. (2018). Robust Distributed Formation Control of Agents with Higher-Order Dynamics. *IEEE Control. Syst. Lett.* 2, 495–500. doi:10.1109/LCSYS.2018.2841941

Franchi, A., Secchi, C., Son, H. I., Bülthoff, H. H., and Giordano, P. R. (2012). Bilateral Teleoperation of Groups of Mobile Robots with Time-Varying Topology. *IEEE Trans. Robotics* 28, 1019–1033. doi:10.1109/TRO.2012.2196304

Franchi, A., Stegagno, P., and Oriolo, G. (2016). Decentralized Multi-Robot Encirclement of a 3D Target with Guaranteed Collision Avoidance. *Autonomous Robots* 40, 245–265. doi:10.1007/s10514-015-9450-3

Fridovich-Keil, D., Herbert, S. L., Fisac, J. F., Deglurkar, S., and Tomlin, C. J. (2018). Planning, Fast and Slow: A Framework for Adaptive Real-Time Safe Trajectory Planning. *IEEE Int. Conf. Robotics Automation*, 387–394. doi:10.1109/ICRA.2018.8460863

Garone, E., and Nicotra, M. M. (2016). Explicit Reference Governor for Constrained Nonlinear Systems. *IEEE Trans. Automatic Control.* 61, 1379–1384. doi:10.1109/TAC.2015.2476195

Garone, E., Nicotra, M., and Ntogramatzidis, L. (2018). Explicit Reference Governor for Linear Systems. *Int. J. Control.* 91, 1415–1430. doi:10.1080/00207179.2017.1317832

Herbert, S. L., Chen, M., Han, S., Bansal, S., Fisac, J. F., and Tomlin, C. J. (2017). FaSTrack: a Modular Framework for Fast and Guaranteed Safe Motion Planning. in Proceedings of the IEEE Conference on Decision and Control, Melbourne, VIC, Australia, December 12–15, 2017. 1517. doi:10.1109/CDC.2017.8263867

Honig, W., Preiss, J. A., Kumar, T. K., Sukhatme, G. S., and Ayanian, N. (2018). Trajectory Planning for Quadrotor Swarms. *IEEE Trans. Robotics* 34, 856–869. doi:10.1109/TRO.2018.2853613

Hua, M. D., Hamel, T., Morin, P., and Samson, C. (2013). Introduction to Feedback Control of Underactuated VTOL Vehicles. *IEEE Control. Syst. Mag.* 33, 61–75. doi:10.1109/MCS.2012.2225931

Jung, S., Hwang, S., Shin, H., and Shim, D. H. (2018). Perception, Guidance, and Navigation for Indoor Autonomous Drone Racing Using Deep Learning. *IEEE Robotics Automation Lett.* 3, 2539–2544. doi:10.1109/LRA.2018.2808368

Khalil, H. K. (2001). *Nonlinear Systems*. 3rd edn. Hobooken, NJ: Prentice-Hall.

Koditschek, D. E., and Rimon, E. (1990). Robot Navigation Functions on Manifolds with Boundary. *Adv. Appl. Math.* 11, 412–442. doi:10.1016/0196-8858(90)90017-S

Kolaric, P., Chen, C., Dalal, A., and Lewis, F. L. (2018). Consensus Controller for Multi-UAV Navigation. *Control. Theor. Technol.* 16, 110–121. doi:10.1007/s11768-018-8013-5

Lee, T. (2011). Geometric Tracking Control of the Attitude Dynamics of a Rigid Body on SO(3). *IEEE Am. Control. Conf.* 3, 1200–1205. doi:10.1109/ACC.2011.5990858

Liu, H., Ma, T., Lewis, F. L., and Wan, Y. (2019). Robust Formation Trajectory Tracking Control for Multiple Quadrotors With Communication Delays. *IEEE Trans. Control. Syst. Technol.* 26 (6), 1–8. doi:10.1109/tcst.2019.2942277

Liu, Y., Montenbruck, J. M., Zelazo, D., Odelga, M., Rajappa, S., Bulthoff, H. H., et al. (2018). A Distributed Control Approach to Formation Balancing and Maneuvering of Multiple Multirotor UAVs. *IEEE Trans. Robotics* 34, 870–882. doi:10.1109/TRO.2018.2853606

Luis, C. E., and Schoellig, A. P. (2019). Trajectory Generation for Multiagent Point-To-Point Transitions via Distributed Model Predictive Control. *IEEE Robotics Automation Lett.* 4, 375–382. doi:10.1109/lra.2018.2890572

Marconi, L., Melchiorri, C., Beetz, M., Pangercic, D., Siegwart, R., Leutenegger, S., et al. (2012). The SHERPA Project: Smart Collaboration between Humans and Ground-Aerial Robots for Improving Rescuing Activities in Alpine Environments (SSRR). in IEEE International Symposium on Safety, Security, and Rescue Robotics. College Station, TX, November 5–8, 2012, doi:10.1109/SSRR.2012.6523905

McGuire, K. N., De Wagter, C., Tuyls, K., Kappen, H. J., and de Croon, G. C. H. E. (2019). Minimal Navigation Solution for a Swarm of Tiny Flying Robots to Explore an Unknown Environment. *Sci. Robotics* 4, 1–14. doi:10.1126/scirobotics.aaw9710

Mellinger, D., and Kumar, V. (2011). Minimum Snap Trajectory Generation and Control for Quadrotors. *IEEE Int. Conf. Robotics Automation*, 2520–2525. doi:10.1109/ICRA.2011.5980409

Murray, R. M. (2007). Recent Research in Cooperative Control of Multivehicle Systems. *J. Dynamic Syst. Meas. Control Trans. ASME* 129, 571–583. doi:10.1115/1.2766721

Nicotra, M. M., Bartulovic, M., Garone, E., and Sinopoli, B. (2015). A Distributed Explicit Reference Governor for Constrained Control of Multiple UAVs. *IFAC-PapersOnLine* 48, 156–161. doi:10.1016/j.ifacol.2015.10.323

Nicotra, M. M., and Garone, E. (2018). The Explicit Reference Governor: A General Framework for the Closed-form Control of Constrained Nonlinear Systems. *IEEE Control. Syst. Mag.* 38, 89–107. doi:10.1109/MCS.2018.2830081

Parker, L. E., Rus, D., and Sukhatme, G. S. (2016). "Multiple Mobile Robot Systems," in *Springer Handbook of Robotics*. Editors B. Siciliano and O. Khatib (Cham: Springer), 1335–1379. doi:10.1007/978-3-319-32552-1_53

Preiss, J. A., Honig, W., Ayanian, N., and Sukhatme, G. S. (2017a). Downwash-aware Trajectory Planning for Large Quadrotor Teams. *IEEE Int. Conf. Intell. Robots Syst.*, 250–257. doi:10.1109/IROS.2017.8202165

Preiss, J. A., Hönig, W., Sukhatme, G. S., and Ayanian, N. (2017b). Crazyswarm: A Large Nano-Quadcopter Swarm. *IEEE Int. Conf. Robotics Automation*, 3299–3304. doi:10.1109/ICRA.2017.7989376

Purohit, A., Zhang, P., Sadler, B. M., and Carpin, S. (2014). Deployment of Swarms of Micro-Aerial Vehicles: From Theory to Practice. *IEEE Int. Conf. Robotics Automation*, 5408–5413. doi:10.1109/ICRA.2014.6907654

Rubies-Royo, V., Fridovich-Keil, D., Herbert, S., and Tomlin, C. J. (2019). "A Classification-Based Approach for Approximate Reachability," in International Conference on Robotics and Automation. 7697–7704. doi:10.1109/ICRA.2019.8793919

Sanket, N. J., Singh, C. D., Ganguly, K., Fermuller, C., and Aloimonos, Y. (2018). GapFlyt: Active Vision Based Minimalist Structure-Less Gap Detection for Quadrotor Flight. *IEEE Robotics Automation Lett.* 3, 2799–2806. doi:10.1109/LRA.2018.2843445

Vásárhelyi, G., Virágh, C., Somorjai, G., Nepusz, T., Eiben, A. E., and Vicsek, T. (2018). Optimized Flocking of Autonomous Drones in Confined Environments. *Sci. Robotics* 3, 1–13. doi:10.1126/scirobotics.aat3536

Vukosavljev, M., Kroeze, Z., Schoellig, A. P., and Broucke, M. E. (2019). A Modular Framework for Motion Planning Using Safe-By-Design Motion Primitives. *IEEE Trans. Robotics* 35, 1233–1252. doi:10.1109/TRO.2019.2923335

Wang, L., Ames, A. D., and Egerstedt, M. (2017). Safe Certificate-Based Maneuvers for Teams of Quadrotors Using Differential Flatness. *IEEE Int. Conf. Robotics Automation*, 3293–3298. doi:10.1109/ICRA.2017.7989375

Wang, N., Su, S. F., Han, M., and Chen, W. H. (2019). Backpropagating Constraints-Based Trajectory Tracking Control of a Quadrotor with Constrained Actuator Dynamics and Complex Unknowns. *IEEE Trans. Syst. Man Cybernetics: Syst* 49, 1322–1337. doi:10.1109/TSMC.2018.2834515

Xu, B., and Sreenath, K. (2018). Safe Teleoperation of Dynamic UAVs through Control Barrier Functions. *IEEE Int. Conf. Robotics Automation*, 7848–7855. doi:10.1109/ICRA.2018.8463194

Yang, T., Sun, N., and Fang, Y. (2021a). Adaptive Fuzzy Control for a Class of MIMO Underactuated Systems With Plant Uncertainties and Actuator Deadzones: Design and Experiments. *IEEE Trans. Cybern.* [Epub ahead of print]. doi:10.1109/TCYB.2021.3050475

Yang, T., Sun, N., Fang, Y., Xin, X., and Chen, H. (2021b). New Adaptive Control Methods for n-Link Robot Manipulators With Online Gravity Compensation: Design and Experiments. *IEEE Trans. Ind. Electron.*. doi:10.1109/TIE.2021.3050371

Zhou, D., Wang, Z., and Schwager, M. (2018). Agile Coordination and Assistive Collision Avoidance for Quadrotor Swarms Using Virtual Structures. *IEEE Trans. Robotics* 34, 916–923. doi:10.1109/TRO.2018.2857477

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.