

## Letter

### Decentralized Control of Multiagent Navigation Systems

Boyang Zhang, *Student Member, IEEE*, and Henri P. Gavin

Dear editor,

In this letter, we introduce a decentralized, nonlinear, discontinuous, and computationally simple control law for large scale multiagent navigation systems. The control is based on extending Gauss's principle of least constraint with a dynamic incorporation of inequality constraints, actuator saturation, and actuator dynamics. With no individual path planner, each agent executes its motion and generates its control actions by reacting solely to the evolution of its constrained dynamics, which is equivalent to solving a linear matrix equation with a dimension up to around 20 without iteration at each time instant. Numerical experiments are conducted on hundreds of two-dimensional (2-D) double integrators subjected to path and collision constraints, demonstrating the promise of the proposed method.

Recent years have witnessed an increasing popularity of multiagent navigation systems in diverse applications such as exploration, surveillance, and rescue [1]. Ongoing research efforts have been focusing on synthesizing multiagent control systems equipped with interagent collision avoidance and computational simplicity [2]. Velocity obstacles (VOs) [3], [4], artificial potential fields (APFs) [5], [6], mixed-integer programs (MIPs) [7], [8], and control barrier functions (CBFs) [9], [10] are among those endeavors.

With certain levels of success achieved, these methods may have shortcomings under certain circumstances. For examples, constant speeds are assumed in collision avoidance in VO methods with no solution uniqueness guarantee [3], [4]. APF methods [5], [6] presume infinitely large control actions in order to prevent collisions as agents get sufficiently close to each other, and it is acknowledged that trajectories can get stuck into local minima. When the number of agents becomes large, MIPs [7], [8] become computationally expensive. CBF methods [9], [10], instead, do not incorporate actuator dynamics and may be computationally intense when solving the associated quadratic program (QP) in which the goals are moving.

In large scale 2-D multiagent navigation systems, it is more important for multiple agents to cooperate as a team than each agent planing their own individual trajectory [1]. In this letter, we propose a control rule for multiagent systems navigation within a prescribed time, with collision avoidance, actuator saturation and dynamics, computational simplicity, and with no individual path planner.

**Related work:** Our method is based on extending Gauss's principle of least constraint (GPLC). The original GPLC [11], along with its equivalency, the Udwadia-Kalaba (U-K) equations [12], has been employed in the control of small scale multiagent systems [13]. However, both the original GPLC and the U-K equations are unable to assimilate inequality constraints, actuator saturation, or actuator dynamics and are not applicable for large scale multiagent systems control.

GPLC has been recently extended by allowing for inequality

Corresponding author: Boyang Zhang.

Citation: B. Zhang and H. P. Gavin, "Decentralized control of multiagent navigation systems," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 5, pp. 922–925, May 2022.

The authors are with the Department of Civil and Environmental Engineering, Duke University, Durham, NC 27708 USA (e-mail: boyang.zhang@duke.edu; henri.gavin@duke.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JAS.2022.105569

constraints, actuator saturation and dynamics and utilized in the centralized control of 2-D multiagent navigation systems [14], [15]. The centralized controls proposed in [14] and [15] formulate constraints in terms of scalar Euclidean distances and vector relative distances, respectively. These formulations, however, cannot allow agents to follow a guiding virtual leader that possesses high-speed, high-curvature trajectories. Moreover, their computational complexities are approximately cubically scaled in the number of agents, making them not suitable for controlling large scale multiagent systems.

In the following, we first lay out foundations for a decentralized control scheme based on a Lagrangian viewpoint of GPLC. We then develop a decentralized multiagent navigation control where constraints are decomposed along different dimensions and partitioned between colliding agents. Next, we present and discuss numerical results for navigating hundreds of agents, followed by the conclusions and future work.

**Gauss's principle of least constraint (GPLC):** Consider a mechanical system with generalized coordinate positions  $\mathbf{q}(t) \in \mathbb{R}^{N_q}$  and with symmetric positive definite inertia matrix  $\mathbf{M} \in \mathbb{R}^{N_q \times N_q}$ . Assume that at any time instant the system is subjected to the set of active holonomic constraints  $\{\mathbf{g}(\mathbf{q}, t) = \mathbf{c}\} \in \mathbb{R}^{N_g}$ , where  $\mathbf{c}$  denotes constant thresholds. By double differentiating with respect to time, these active constraints become linear in the generalized coordinate accelerations,

$$\frac{d^2}{dt^2} \mathbf{g}(\mathbf{q}, t) = \frac{\partial \mathbf{g}}{\partial \mathbf{q}} \ddot{\mathbf{q}} + \frac{d}{dt} \left( \frac{\partial \mathbf{g}}{\partial \dot{\mathbf{q}}} \right) \dot{\mathbf{q}} + \frac{d}{dt} \left( \frac{\partial \mathbf{g}}{\partial t} \right) = \mathbf{0} \quad (1)$$

which can be compactly expressed as

$$\mathbf{A}(\mathbf{q}, t) \ddot{\mathbf{q}} = \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (2)$$

where  $\mathbf{A} \in \mathbb{R}^{N_g \times N_q}$  is the constraint Jacobian matrix.

The system Lagrangian can be defined as  $L = T - V$ , where  $T = \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} / 2$  denotes the kinetic energy, and  $V$  denotes the potential energy depending only on  $\mathbf{q}$ .  $L$  may be augmented without changing its value as  $\bar{L} = L - \lambda^T (\mathbf{g} - \mathbf{c})$ , where  $\lambda \in \mathbb{R}^{N_g}$  contains the Lagrange multipliers associated with  $\mathbf{g}$ .

Applying Euler-Lagrange equations to  $\bar{L}$ , we obtain the constrained equations of motion [15]

$$\mathbf{M} \ddot{\mathbf{q}} - \mathbf{f}(\mathbf{q}) + \mathbf{A}^T \lambda = \mathbf{0} \quad (3)$$

where  $\mathbf{f}(\mathbf{q}) = -(\partial V / \partial \mathbf{q})^T$  are the nonconstraint forces due to a conservative potential  $V$ . Hence, we can obtain the constrained system dynamics by adjoining (3) and (2)

$$\begin{bmatrix} \mathbf{M} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} \quad (4)$$

where  $\mathbf{0}$  denotes a zero matrix.  $\mathbf{f}^c \triangleq -\mathbf{A}^T \lambda$  represents the control actions enforcing the constraints. In the context of multiagent navigation systems studied herein, system (4) equates the GPLC Karush-Kuhn-Tucker (KKT) system in [14] which is derived from Newton's second law.

**Active constraint stabilization:** In finite-precision numerical integration, the accumulated numerical errors prohibit the enforcement of  $\mathbf{g} = \mathbf{c}$  given  $\ddot{\mathbf{g}} = \mathbf{0}$  (i.e., (2)) and the initial conditions,  $\mathbf{g}(0) = \mathbf{c}$  and  $\dot{\mathbf{g}}(0) = \mathbf{0}$ . Moreover, the state trajectories may depart from some constraint due to any actuator saturation or dynamics. In GPLC approaches, the dynamics of active constraints provide the command control actions, and the constraint errors are asymptotically stabilized to the constant thresholds  $\mathbf{c}$  by Baumgarte's stabilization [16]

$$\ddot{\mathbf{g}} + 2\zeta\omega\dot{\mathbf{g}} + \omega^2(\mathbf{g} - \mathbf{c}) = \mathbf{0} \quad (5)$$

where  $\mathbf{g} \in \mathbf{g}$  is any active constraint with the corresponding threshold  $\mathbf{c} \in \mathbf{c}$ . Imposing the second-order dynamics (5) to all active

constraints  $\mathbf{g}$  (with individual constraint taking possibly different values of  $\omega$  and  $\zeta$ ) and expressing them in a compact manner, we get

$$\mathbf{A}(\mathbf{q}, t)\ddot{\mathbf{q}} = \hat{\mathbf{b}}(\mathbf{q}, \dot{\mathbf{q}}, t). \quad (6)$$

Compared to  $\mathbf{b}$  in (2),  $\hat{\mathbf{b}}$  contains terms involving  $\omega$  and  $\zeta$ , which may be respectively regarded as the *natural frequency* and *damping ratio* of *constraint oscillators*  $\mathbf{g}$  and thus may be interpreted as control parameters. Hence, system (4) incorporated with the stabilized constraint dynamics (6) becomes

$$\begin{bmatrix} \mathbf{M} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \hat{\mathbf{b}} \end{bmatrix}. \quad (7)$$

**Regularization:** When maneuvering a large number of agents,  $\mathbf{A}$  may be a tall matrix, i.e.,  $N_{\mathbf{g}} > N_{\mathbf{q}}$ . In addition,  $\mathbf{A}$  may be degenerate due to some constraints being linearly dependent at some time instants. In this study, we regularize system (7) as [14]

$$\begin{bmatrix} \mathbf{M} & \mathbf{A}^T \\ \mathbf{A} & -\alpha\mathbf{I} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \hat{\mathbf{b}} \end{bmatrix} \quad (8)$$

where  $\mathbf{I}$  denotes an identity matrix, and  $0 < \alpha \ll \min \Lambda(\mathbf{M})$  is the regularization factor that guarantees the solution uniqueness, where  $\min \Lambda(\mathbf{M})$  is the minimum eigenvalue of  $\mathbf{M}$ .

Proposition 1: The 2-by-2 block matrix in (8) is nonsingular.

Proof: Since  $\mathbf{M}$  is positive definite and thus nonsingular,  $\mathbf{M}^{-1}$  exists and is also positive definite. Define  $\mathbf{S} \triangleq \alpha\mathbf{I} + \mathbf{A}\mathbf{M}^{-1}\mathbf{A}^T$ , where  $\alpha > 0$ . Suppose that  $\mathbf{S}$  is singular, then  $\exists \mathbf{x} \neq \mathbf{0}$  such that  $\mathbf{S}\mathbf{x} = -\mathbf{S}\mathbf{x} = \mathbf{0}$ . Hence,  $0 = \mathbf{x}^T \mathbf{S}\mathbf{x} = \alpha \mathbf{x}^T \mathbf{x} + \mathbf{x}^T \mathbf{A}\mathbf{M}^{-1}\mathbf{A}^T \mathbf{x} = \alpha \mathbf{x}^T \mathbf{x} + (\mathbf{A}^T \mathbf{x})^T \mathbf{M}^{-1}(\mathbf{A}^T \mathbf{x})$ . But  $\alpha \mathbf{x}^T \mathbf{x} > 0$ ,  $(\mathbf{A}^T \mathbf{x})^T \mathbf{M}^{-1}(\mathbf{A}^T \mathbf{x}) \geq 0$  with equality holding only for  $\mathbf{A}^T \mathbf{x} = \mathbf{0}$  (i.e.,  $\mathbf{A}$  being singular). Thus  $0 = \mathbf{x}^T \mathbf{S}\mathbf{x} > 0$ , a contradiction. Hence,  $\mathbf{S}$  is nonsingular, as is  $-\mathbf{S}$ . By Proposition 3.9.7 of [17], the 2-by-2 block matrix in (8) is nonsingular, regardless of the rank of  $\mathbf{A}$  being full or deficient. ■

**An equivalent system with scaled constraints:** If the system is subjected to a set of scaled active constraints  $\gamma(\mathbf{g} - \mathbf{c}) = \mathbf{0}$ , where  $\gamma$  is a nonzero scalar, then we have  $\gamma\mathbf{g} = \mathbf{0}$  and  $\gamma\dot{\mathbf{g}} = \mathbf{0}$  ( $\Leftrightarrow \gamma\mathbf{A}\dot{\mathbf{q}}' = \gamma\mathbf{b}$ ). We thus obtain a new augmented Lagrangian by adjoining the scaled constraints with multipliers  $\lambda'$  and solve the new constrained dynamics analogously by applying Euler-Lagrange equations, active constraint stabilization, and regularization to arrive upon

$$\begin{bmatrix} \mathbf{M} & \gamma\mathbf{A}^T \\ \gamma\mathbf{A} & -\alpha'\mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}' \\ \lambda' \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \gamma\hat{\mathbf{b}} \end{bmatrix}. \quad (9)$$

Proposition 2: Under  $\alpha' = \gamma^2\alpha$ ,  $\gamma \neq 0$ , system (9) yields the same constrained accelerations as those of (8), i.e.,  $\ddot{\mathbf{q}} = \ddot{\mathbf{q}}'$ .

Proof: By Proposition 1, the block coefficient matrices in (8) and (9) are nonsingular. According to Proposition 3.9.7 of [17], the inverse of the coefficient matrix in (8) is

$$\begin{bmatrix} \mathbf{M}^{-1} - \mathbf{M}^{-1}\mathbf{A}^T\mathbf{S}^{-1}\mathbf{A}\mathbf{M}^{-1} & \mathbf{M}^{-1}\mathbf{A}^T\mathbf{S}^{-1} \\ \mathbf{S}^{-1}\mathbf{A}\mathbf{M}^{-1} & -\mathbf{S}^{-1} \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 \\ \mathbf{H}_3 & \mathbf{H}_4 \end{bmatrix}$$

and the inverse of the coefficient matrix in (9) is

$$\begin{bmatrix} \mathbf{H}_1 & \gamma^{-1}\mathbf{H}_2 \\ \gamma^{-1}\mathbf{H}_3 & \gamma^{-2}\mathbf{H}_4 \end{bmatrix}.$$

Thus,  $\ddot{\mathbf{q}}' = \mathbf{H}_1\mathbf{f} + \gamma^{-1}\mathbf{H}_2\gamma\hat{\mathbf{b}} = \mathbf{H}_1\mathbf{f} + \mathbf{H}_2\hat{\mathbf{b}} = \ddot{\mathbf{q}}$ , and  $\lambda' = \gamma^{-1}\mathbf{H}_3\mathbf{f} + \gamma^{-2}\mathbf{H}_4\gamma\hat{\mathbf{b}} = \gamma^{-1}(\mathbf{H}_3\mathbf{f} + \mathbf{H}_4\hat{\mathbf{b}}) = \gamma^{-1}\lambda$ . ■

Remark 1: Proposition 2 holds for any two linear matrix equations with a partitioned 2-by-2 block coefficient matrix that has square, nonsingular block matrices on its main diagonal and that the top left block of the coefficient matrix and its Schur complement are both nonsingular.

**Problem statement:** The problem at hand is a swarm of  $N$  agents navigating in a 2-D plane. Agent  $i$  possesses the position  $\mathbf{p}_i(t) = [x_i(t) \ y_i(t)]^T \in \mathbb{R}^2$  with a constant mass  $m_i$  and is subjected to an actuator saturation bound  $\boldsymbol{\kappa}_i = [\kappa_i^x \ \kappa_i^y]^T \in \mathbb{R}^2$ . In this work, we investigate a homogeneous swarm and thus assume  $\kappa_i^x = \kappa_i^y = \kappa_i$ .

In the following, the superscripts  $u, l, x$ , and  $y$  denote upper branch, lower branch,  $X$  component, and  $Y$  component, respectively, while the subscripts  $i, j$ , and  $0$  denote the indices of agent  $i$ , agent  $j$ , and the virtual leader, respectively. The subscripts and superscripts are used where it is germane to the corresponding quantity.

Assume that the  $i$ -th agent has unconstrained dynamics

$$\begin{bmatrix} m_i & 0 \\ 0 & m_i \end{bmatrix} \begin{bmatrix} \ddot{x}_i \\ \ddot{y}_i \end{bmatrix} = \begin{bmatrix} f_i^x \\ f_i^y \end{bmatrix} \Leftrightarrow \mathbf{M}_i \ddot{\mathbf{p}}_i = \mathbf{f}_i \quad (10)$$

where  $f_i^x$  and  $f_i^y$  contain all nonconstraint conservative forces exerted on agent  $i$  along the  $X$  and  $Y$  axes, respectively.

Let the swarm be forced to approach a virtual leader with position  $\mathbf{p}_0(t) = [x_0(t) \ y_0(t)]^T \in \mathbb{R}^2$ . The forces attempt to satisfy the path constraints

$$\mathbf{g}_{i0} = [g_{i0}^x \ g_{i0}^y]^T \triangleq \Delta \mathbf{p}_{i0} = \mathbf{p}_i - \mathbf{p}_0 = \mathbf{0}. \quad (11)$$

All adjacent agent pairs  $\{i, j\}$  must avoid collisions,  $\forall i, j \in \{1, \dots, N\}$ . We enclose each agent within a virtual rectangular buffer so that for agent pair  $\{i, j\}$  the buffer size between them along the  $X$  and  $Y$  axes is  $\mathbf{r}_{ij} = \mathbf{r}_{ji} = \mathbf{r}_i + \mathbf{r}_j = [r_{ij}^x \ r_{ij}^y]^T \in \mathbb{R}^2$ . Mathematically, the collision constraints are

$$\mathbf{g}_{ij} = [g_{ij}^x \ g_{ij}^y]^T \triangleq |\Delta \mathbf{p}_{ij}| = |\mathbf{p}_i - \mathbf{p}_j| \geq \mathbf{r}_{ij} \quad (12)$$

where  $|\cdot|$  denotes absolute value, and the inequality sign denotes componentwise comparison.

Constraints (12) can be decomposed into two branches: an upper and a lower bound on  $\mathbf{p}_i$ , i.e.,  $\mathbf{g}_{ij}^u = [g_{ij}^{xu} \ g_{ij}^{yu}]^T \triangleq \{\Delta \mathbf{p}_{ij} \leq -\mathbf{r}_{ij}\}$  and  $\mathbf{g}_{ij}^l = [g_{ij}^{xl} \ g_{ij}^{yl}]^T \triangleq \{\Delta \mathbf{p}_{ij} \geq \mathbf{r}_{ij}\}$ . Thus, the upper and lower branch of active  $\mathbf{g}_{ij}$  for agent  $i$  become  $\{-\mathbf{r}_{ij} < \Delta \mathbf{p}_{ij} \leq \mathbf{0}\}$  and  $\{\mathbf{0} < \Delta \mathbf{p}_{ij} < \mathbf{r}_{ij}\}$ , respectively.

During the natural evolution of the multiagent swarm dynamics, equality path constraints (11) and some inequality collision constraints (12) become active. In fact, in GPLC control, control actions result only from the time-varying set of active constraints. Likewise, agent  $i$  is not exactly tracking the leader if  $\mathbf{g}_{i0}$  is nonzero along at least one dimension, while the pair  $\{i, j\}$  is in danger of collision when both  $g_{ij}^x$  and  $g_{ij}^y$  are activated. Taking the first- and second-order time derivative of  $\mathbf{g}_{i0}$ ,  $\mathbf{g}_{ij}^u$ , and  $\mathbf{g}_{ij}^l$ , respectively, we can stabilize the imposed second-order dynamics of all active constraints as in (5).

**Constraint partitioning:** For the two types of constraints considered in this study, each path constraint is independent from the coordinates of other agents, while each collision constraint couples the dynamics of pairwise agents in collision. Upon the onset of colliding buffers between agent pair  $\{i, j\}$ , we assume that each agent only has access to the relative positions  $\mathbf{p}_{ij}$  and the relative velocities  $\dot{\mathbf{p}}_{ij}$  with respect to its neighbors and that  $\dot{\mathbf{p}}_i = -\dot{\mathbf{p}}_j$  (due to the homogeneity assumption introduced earlier). Hence,  $\dot{\mathbf{g}}_{ij}^u = \mathbf{0} \Leftrightarrow 2\dot{\mathbf{p}}_i = \mathbf{0}$  and  $\dot{\mathbf{g}}_{ij}^l = \mathbf{0} \Leftrightarrow -2\dot{\mathbf{p}}_i = \mathbf{0}$ . Equivalently, this indicates that the collision constraint in the decentralized framework is scaled by a factor of 2 compared to that of centralized frameworks [14], [15]. According to Proposition 2 and Remark 1, the constrained dynamics for agent  $i$  in the multiagent navigation system can be formulated by (9)

$$\begin{bmatrix} \mathbf{M}_i & \mathbf{I}_{2 \times 2} & 2(\mathbf{A}_i^{ac})^T \\ \mathbf{I}_{2 \times 2} & -\alpha_i \mathbf{I} & \mathbf{0}_{2 \times N_{\mathbf{g}}^{ac}} \\ 2\mathbf{A}_i^{ac} & \mathbf{0}_{N_{\mathbf{g}}^{ac} \times 2} & -4\alpha_i \mathbf{I} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}}_i \\ \lambda_i^{ap} \\ \lambda_i^{ac} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_i \\ \hat{\mathbf{b}}_i^{ap} \\ 2\hat{\mathbf{b}}_i^{ac} \end{bmatrix} \quad (13)$$

where  $\mathbf{I}_{2 \times 2}$  corresponds to *active path* constraints (11), and each row of  $\mathbf{A}_i^{ac}$  represents an *active collision* constraint. Note that  $\mathbf{A}_i^{ac}$  contains only 1, -1, and 0 as entries. We define  $\mathbf{A}_i = [\mathbf{I}_{2 \times 2} \ 2(\mathbf{A}_i^{ac})^T]^T$ ,  $\hat{\mathbf{b}}_i = [(\hat{\mathbf{b}}_i^{ap})^T \ 2(\hat{\mathbf{b}}_i^{ac})^T]^T$ , and  $\lambda_i = [(\lambda_i^{ap})^T \ (\lambda_i^{ac})^T]^T$ .

Note that the assumption  $\dot{\mathbf{p}}_i = -\dot{\mathbf{p}}_j$  needs not indicate  $\ddot{\mathbf{p}}_i = -\ddot{\mathbf{p}}_j$

and  $\mathbf{p}_i = -\mathbf{p}_j$ , since the trajectories of  $\mathbf{p}_i$  and  $\mathbf{p}_j$  depend not only on  $\dot{\mathbf{p}}_i$  and  $\dot{\mathbf{p}}_j$  but also on the initial conditions  $\mathbf{p}_i(0)$ ,  $\dot{\mathbf{p}}_i(0)$ ,  $\mathbf{p}_j(0)$ , and  $\dot{\mathbf{p}}_j(0)$ , where the time  $t = 0$  is the time instant when  $\mathbf{g}_{ij}$  becomes active. Moreover, the dynamics of other active constraints will also contribute to the trajectories of  $\mathbf{p}_i$  and  $\mathbf{p}_j$ . Also note that the collision control parameters  $\omega_{ij}$ ,  $\zeta_{ij}$ , and  $\mathbf{r}_{ij}$  can be chosen differently (and/or vary dynamically) for upper/lower branch,  $X/Y$  coordinate, and/or heterogeneous agents, as the problem at hand dictates.

In the 2-D multiagent system examined here, the number of active constraints for one agent may range from 2 path constraints in a collision-free configuration to around 8 collision constraints plus 2 path constraints in a typical configuration to around 16 collision constraints plus 2 path constraints in an extremely dense configuration. The command control law  $\mathbf{f}_i^c = -\mathbf{A}_i^T \lambda_i$  is decentralized since the information to populate  $\mathbf{M}_i$ ,  $\mathbf{A}_i$ , and  $\mathbf{b}_i$  is available to agent  $i$  from the measurements of its own motion relative to its neighbors and the virtual leader. The feedback control law reduces to solving a linear matrix equation (with a dimension of 4 in a collision-free configuration to around 12 in a typical configuration to around 20 in an extremely dense configuration, depending on the number of active collision constraints) for  $\lambda_i$  without iteration at each time instant. The feedback control law is therefore nonlinear and discontinuous in the states and is computationally simple as compared to solving a MIP or QP.

**Actuator saturation and dynamics:** Assume that the actuator of agent  $i$  is saturated as  $\mathbf{f}_i^s = \min\{\max\{\mathbf{f}_i^c, -\boldsymbol{\kappa}_i\}, \boldsymbol{\kappa}_i\}$  and responds to first-order dynamics,  $\dot{\mathbf{f}}_i^a = (\mathbf{f}_i^s - \mathbf{f}_i^a)/\tau_i$ , where  $\mathbf{f}_i^s$  is the saturated control force,  $\mathbf{f}_i^a$  is the actual control force, and  $0.05 \text{ s} \lesssim \tau_i \lesssim 1.0 \text{ s}$  is a time constant.

With the saturated and delayed actuator force  $\mathbf{f}_i^a$ , the actual acceleration is recomputed by  $\hat{\mathbf{p}}_i = \mathbf{M}_i^{-1}(\mathbf{f}_i + \mathbf{f}_i^a)$ . The state derivatives  $\dot{\mathbf{p}}_i$ ,  $\dot{\hat{\mathbf{p}}}_i$ , and  $\dot{\mathbf{f}}_i^a$  are then used in numerical integration for agent  $i$ .

**Numerical experiments:** The proposed method is numerically implemented in MATLAB on a Windows desktop with an Intel Xeon E5-2680v3 CPU at 2.50 GHz. The parameters used in this study are:  $m_i = 1 \text{ kg}$ ,  $g = 9.8 \text{ m/s}^2$ ,  $\boldsymbol{\kappa}_i = 5m_i g \text{ N}$ ,  $r_{ij}^x = r_{ij}^y = 40 \text{ m}$ ,  $\omega_{i0}^x = \omega_{i0}^y = 1 \text{ rad/s}$ ,  $\omega_{ij}^x = \omega_{ij}^y = 5 \text{ rad/s}$ ,  $\zeta_{i0}^x = \zeta_{i0}^y = 0.9$ ,  $\zeta_{ij}^x = \zeta_{ij}^y = 5$ ,  $4\alpha_i = m_i/10^4$ ,  $\tau_i = 0.2 \text{ s}$ . The fourth-order Runge-Kutta method with a constant time step of 0.005 s is used as the time integrator. The virtual leader executes a figure-eight trajectory about 5.25 km in length within a 60-second simulation window (average speed  $\sim 315 \text{ km/h}$ ). The leader's path is described as

$$\begin{aligned} x_0(t) &= \frac{L \cos(2\pi t/T)}{1 + \sin(2\pi t/T) \sin(2\pi t/T)} \\ y_0(t) &= \frac{L \sin(2\pi t/T) \cos(2\pi t/T)}{1 + \sin(2\pi t/T) \sin(2\pi t/T)}. \end{aligned} \quad (14)$$

In this study, we use  $L = 1 \text{ km}$ ,  $T = 60 \text{ s}$ , and  $N = 200$ . All agents start from randomly generated initial positions,  $x_i(0) \in [600, 1400]$  and  $y_i(0) \in [-400, 400]$ ,  $\forall i \in \{1, \dots, N\}$ , with zero initial velocities, and the conservative forces are assumed to be zero for all time.

Figs. 1–3 respectively illustrate the trajectories, the minimum active collision distances, and the extreme saturated and actual control actions for 200 agents following the virtual leader. The corresponding animation can be found online<sup>1</sup>. The chosen control parameters indicate that the path constraints are underdamped with a smaller natural frequency while the collision constraints are overdamped with a larger natural frequency and thus stronger control actions.

We can observe from Fig. 1 that all agents execute smooth trajectories by reacting purely to the natural evolution of constrained dynamics (13). In Fig. 2, the minimum pairwise distance  $\min \|\Delta \mathbf{p}_{ij}\|$

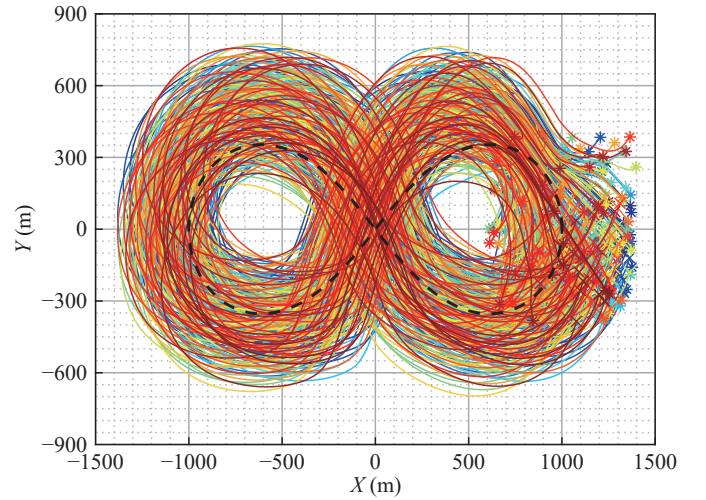


Fig. 1. Trajectories of 200 agents swarming with a virtual leader moving along a figure-eight path. The black dashed line denotes the virtual leader's path around 5.25-km long executed within 60 s (average speed  $\sim 315 \text{ km/h}$ ). The colored solid lines are the paths of all agents, with \* and  $\times$  respectively denoting the start and end positions. Each agent's decentralized navigation control law results from reacting purely to its constrained dynamics (13).

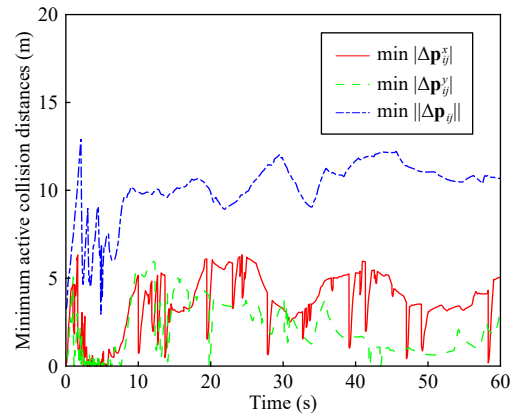


Fig. 2. The minimum relative distance over all active colliding agent pairs along the  $X$  axis, the  $Y$  axis, and between the centroids, respectively. The minimum pairwise distance starts from 3.26 m and is stabilized to around 10 m after the first 5 seconds of initial transients.

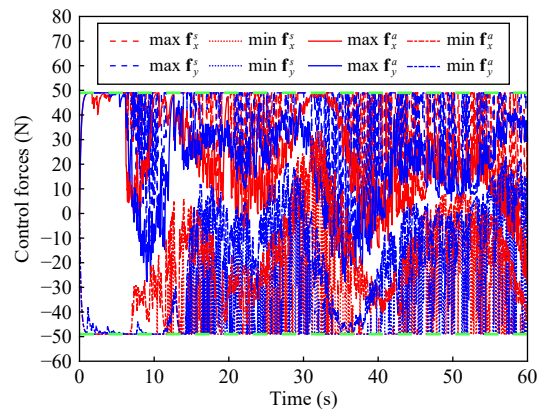


Fig. 3. The maximum and minimum saturated and actual control forces,  $\mathbf{f}_i^s$  and  $\mathbf{f}_i^a$ , over all agents.  $\mathbf{f}_i^s$  precedes  $\mathbf{f}_i^a$  due to the first-order actuator dynamics. The green dashed lines denote the actuator saturation bounds.

<sup>1</sup> [https://youtu.be/C0\\_q3lxDYyY](https://youtu.be/C0_q3lxDYyY)

starts from 3.26 m and is stabilized to around 10 m after the first 5-second of initial transients, during which the actual control forces are almost saturated, as shown in Fig. 3. This means that no actual interagent collisions occurred and that the fluctuating  $\min\|\Delta\mathbf{p}_i\|$  in  $t \in [0, 5]$  s results from the lack of enough actuation to enforce both path and active collision constraints. Note that the virtual leader's path has high speeds and high curvatures, and thus, it is an aggressive and challenging maneuver that the centralized controls developed in [14] and [15] fail to perform stably. This explains the generally large extreme control actions and even some occurrences of control saturation in Fig. 3.

In the proposed decentralized framework, each agent solves a KKT system of dimension up to around 20, regardless of the number of the agents in the swarm,  $N$ . In the centralized methods [14], [15], the corresponding KKT system is of dimension around 11 times  $N$  and of dimension around 20 times  $N$ , respectively. Since solving a linear matrix equation involves  $O(N^3)$  operations, the centralized approaches [14], [15] are more computationally intensive than the proposed decentralized scheme as  $N$  becomes large.

**Conclusions and future work:** This letter extends GPLC to address the decentralized control of multiagent navigation systems in order to guide each agent toward a moving virtual leader and away from potential collisions. This work defines constraints in terms of decomposed relative distances and partitions collision constraints between colliding agent pairs. This method has no agentwise path planner and is heuristic-free, decentralized, nonlinear, discontinuous, and computationally simple. The proposed decentralized scheme to safely maneuver an  $N$ -agent swarm requires only  $O(1/N^3)$  of the computational operations of those required by centralized GPLC control approaches.

Future work along this line of research include the examination of the proposed framework subjected to exogenous disturbances, modelling errors, and state estimation errors, which are straightforward endeavors thanks to the simplicity of the mathematical structure of the present approach.

**Acknowledgments:** The authors would like to thank U.S. Army Research Office under award number 75568 -NS-II and Duke University for the financial support of this work.

## References

- [1] Y. Liu and R. Bucknall, "A survey of formation control and motion planning of multiple unmanned vehicles," *Robotica*, vol. 36, no. 7, pp. 1019–1047, 2018.
- [2] S. Huang, R. S. H. Teo, and K. K. Tan, "Collision avoidance of multi unmanned aerial vehicles: A review," *Annu. Rev. Contr.*, vol. 48, pp. 147–164, 2019.
- [3] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Proc. Int. Symp. Robot. Res.*, 2011, pp. 3–19.
- [4] J. A. Douthwaite, S. Zhao, and L. S. Mihaylova, "Velocity obstacle approaches for multi-agent collision avoidance," *Unmanned Syst.*, vol. 7, no. 1, pp. 55–64, 2019.
- [5] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. 40th IEEE Conf. Decis. Control*, vol. 3, 2001, pp. 2968–2973.
- [6] M. T. Wolf and J. W. Burdick, "Artificial potential functions for highway driving with collision avoidance," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 3731–3736.
- [7] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *Proc. Eur. Control Conf.*, 2001, pp. 2603–2608.
- [8] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 477–483.
- [9] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Trans. Robot.*, vol. 33, no. 3, pp. 661–674, 2017.
- [10] J. Grover, C. Liu, and K. Sycara, "Deadlock analysis and resolution in multi-robot systems (extended version)," arXiv preprint arXiv: 1911.09146, 2019.
- [11] C. F. Gauß, "Über ein neues allgemeines grundgesetz der mechanik," *J. für die Reine und Angew. Math.*, vol. 4, pp. 232–235, 1829.
- [12] F. E. Udwardia and R. E. Kalaba, "A new perspective on constrained motion," *Proc. R. Soc. A*, vol. 439, no. 1906, pp. 407–410, 1992.
- [13] R. Zhao, M. Li, Q. Niu, and Y.-H. Chen, "Udwadia-Kalaba constraint-based tracking control for artificial swarm mechanical systems: Dynamic approach," *Nonlinear Dyn.*, 2020.
- [14] B. Zhang and H. P. Gavin, "Gauss' s principle with inequality constraints for multiagent navigation and control," *IEEE Trans. Automat. Contr.*, vol. 67, no. 2, pp. 810–823, 2022.
- [15] B. Zhang and H. P. Gavin, "Natural deadlock resolution for multi-agent multi-swarm navigation," in *Proc. 60th IEEE Conf. Decis. Control*, 2021, pp. 5958–5963.
- [16] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Comput. Methods in Appl. Mech. Eng.*, vol. 1, no. 1, pp. 1–16, 1972.
- [17] D. S. Bernstein, *Scalar, Vector, and Matrix Mathematics: Theory, Facts, and Formulas-Revised and Expanded Edition*. Princeton, NJ, USA: Princeton University Press, 2018.