

An energy-optimal framework for assignment and trajectory generation in teams of autonomous agents

Logan Beaver, Andreas A. Malikopoulos*

Department of Mechanical Engineering, University of Delaware, Newark, DE, 19716, USA



ARTICLE INFO

Article history:

Received 7 November 2019

Received in revised form 30 January 2020

Accepted 6 March 2020

Available online xxxx

Keywords:

Multi-agent systems
Decentralized control
Optimal control
Assignment
Trajectory generation

ABSTRACT

In this paper, we present an approach for solving the problem of moving N homogeneous agents into $M \geq N$ goal locations along energy-minimizing trajectories. We propose a decentralized framework that only requires knowledge of the goal locations and partial observations of the global state by each agent. The framework includes guarantees on safety through dynamic constraints, and a method to impose a dynamic, global priority ordering on the agents. A solution to the goal assignment and trajectory generation problems is derived in the form of a binary program and a nonlinear system of equations. Then, we present the conditions for optimality and characterize the conditions under which our algorithm is guaranteed to converge to a unique assignment of agents to goals. We also solve the fully constrained decentralized trajectory generation problem considering the state, control, and safety constraints. Finally, we validate the efficacy of our approach through a numerical simulation in MATLAB.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

1.1. Motivation

Complex systems consist of diverse entities that interact both in space and time [1]. Referring to something as complex implies that it consists of interdependent entities or agents that can adapt, i.e., they can respond to their local and global environment. Complex systems appear in many applications, including cooperation between components of autonomous systems, sensor fusion, and natural biological systems. As we move to increasingly complex systems [2], new control approaches are needed to optimize the impact of individuals on system-level behavior through the control of individual entities [3,4].

Swarms are typical complex system which have attracted considerable attention in many applications, e.g., transportation, exploration, construction, surveillance, and manufacturing. As discussed by Oh et al. [5], swarms are especially attractive due to their natural parallelization and general adaptability. One of the typical multiagent applications is creating desired formations. However, due to cost constraints on any real swarm of autonomous agents, e.g., limited computation capabilities, battery capacity, and sensing capabilities, any efficient control approach needs to take into account the energy consumption of each

agent. Moving agents into a desired formation has been explored previously; however, creating this formation while minimizing energy consumption is an open problem.

1.2. Related work

Brambilla et al. [6] classified approaches to swarms into two distinct groups, macroscopic, and microscopic. Macroscopic approaches generate group behavior from a system of partial differential equations which are spatially discretized and applied to individual agents; this approach is fundamentally based on work by Turing [7], and is used extensively in bio-inspired formation and pattern forming [8]. Our approach is microscopic; that is, we control the behavior of individual agents to achieve some desired global outcome. Microscopic approaches are based on the seminal work by Reynolds [9], which applied an agent-based method to capture the flocking behavior of birds.

There is a rich literature on the creation of a desired formation, such as the construction of rigid formations from triangular sub-structures [10,11], formation algorithms inspired by crystal growth [12], and growing swarm formations in a lattice structure [13]. It is also possible to build formations using only scalar, bearing, or distance measurements to move agents into a desired formation [14,15]. Olfati-Saber et al. [16] proved that many formation problems may be solved by applying a modified form of the basic consensus algorithm. However, none of these approaches consider the energy cost to individual agents in the swarm.

* Corresponding author.

E-mail addresses: lbeaver@udel.edu (L. Beaver), andreas@udel.edu (A.A. Malikopoulos).

A significant amount of work has applied optimization methods to designing potential fields for agent interaction [17–21]. However, these approaches optimize the shape of the potential field and do not consider the energy consumption of individual agents. Previous work by Turpin et al. [22] has generated optimal assignments using a centralized planner. Other approaches require global information about the system (e.g., [22] requires globally unique assignments a priori, [23] requires the graph diameter of the system, and [24] imposes global “seed” agents on the swarm).

Our approach is decentralized, and thus, each agent may only partially observe the entire system state. The latter results in a non-classical information structure and many techniques for solving centralized systems do not hold [25]. To address this problem, one may impose a priority ordering on the agents. This has been achieved in previous work through a centralized controller [26,27]. In general, finding an optimal ordering is NP-Hard, and an optimal ordering is not always guaranteed to exist [28]. To reduce the complexity of ordering agents, much work has been done to decentralize the ordering problem, including applying discretized path-based heuristics [29], and reinforcement learning [30]. In contrast, our approach introduces a decentralized method of dynamically ordering agents that is path independent and relies only on information directly observable by each agent.

Our approach only requires agents to make partial observations of the entire system. This may lead to performance degradation relative to a centralized controller with global knowledge. However, this is a fundamental feature of decentralized control problems in general [31]. Other efforts have attempted to circumvent this issue with information sharing, either directly [25] or through decentralized auctioning [23]. However, these approaches tend to require knowledge of the global communication graph, long delays before decisions are made, or both. In contrast, we embrace partial observability of the system and exploit it to reduce the computational load on each individual agent.

The main contributions of this paper are: (1) a decentralized set of *interaction dynamics*, which impose a priority order on agents in a decentralized manner, (2) an assignment algorithm that exploits the unconstrained optimal trajectory of the agents, (3) guarantees on the stability of the proposed control policy, and (4) optimality conditions for the fully constrained collision avoidance problem in \mathbb{R}^2 , as well as a locally optimal solution in the form of nonlinear algebraic equations.

1.3. Organization of this paper

The remainder of this paper proceeds as follows. In Section 2, we formulate the decentralized optimal control problem, and we decompose it into the coupled assignment and trajectory generation subproblems. In Section 3, we present the conditions which guarantee system convergence along with the assignment problem. Then, in Section 4, we prove that these conditions are satisfied by our framework and solve the optimal trajectory generation problem. Finally, in Section 5, we present a series of MATLAB simulations to show the performance of the algorithm, and we presented concluding remarks in Section 6.

2. Problem formulation

We consider a swarm of $N \in \mathbb{N}$ autonomous agents indexed by the set $\mathcal{A} = \{1, \dots, N\}$. Our objective is to design a decentralized control framework to move the N agents into $M \in \mathbb{N}$ goal positions, indexed by the set $\mathcal{F} = \{1, \dots, M\}$. We consider the case where $N \leq M$, i.e., no redundant agents are brought to fill the formation. This requirement can be relaxed by defining

a behavior for excess agents, such as idling [32]. Each agent $i \in \mathcal{A}$ follows double-integrator dynamics,

$$\dot{\mathbf{p}}_i(t) = \mathbf{v}_i(t), \quad (1)$$

$$\dot{\mathbf{v}}_i(t) = \mathbf{u}_i(t), \quad (2)$$

where $\mathbf{p}_i(t) \in \mathbb{R}^2$ and $\mathbf{v}_i(t) \in \mathbb{R}^2$ are the time-varying position and velocity vectors respectively, and $\mathbf{u}_i(t) \in \mathbb{R}^2$ is the control input (acceleration/deceleration) over time $t \in [t_i^0, t_i^f]$, where t_i^0 is the initial time for agent i and $t_i^f \in \mathbb{R}_{>0}$ is the terminal time for agent i . Additionally, each agent’s control input and velocity are bounded by

$$v_{\min} \leq \|\mathbf{v}_i(t)\| \leq v_{\max}, \quad (3)$$

$$u_{\min} \leq \|\mathbf{u}_i(t)\| \leq u_{\max}, \quad (4)$$

where $\|\cdot\|$ is the Euclidean norm. Thus, the state of each agent $i \in \mathcal{A}$ is given by the time-varying vector

$$\mathbf{x}_i(t) = \begin{bmatrix} \mathbf{p}_i(t) \\ \mathbf{v}_i(t) \end{bmatrix}, \quad (5)$$

and we denote the global (system) state as

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{x}_1(t) \\ \dots \\ \mathbf{x}_N(t) \end{bmatrix}. \quad (6)$$

The energy consumption of any agent $i \in \mathcal{A}$ is given by

$$\dot{E}_i(t) = \frac{1}{2} \|\mathbf{u}_i(t)\|^2. \quad (7)$$

We select the L^2 norm of the control input as our energy model since, in general, acceleration/deceleration requires more energy than applying no control input. Therefore, we expect that minimizing the acceleration/deceleration of each agent will yield a proportional reduction in energy consumption.

Our objective is to develop a decentralized framework for the N agents to optimally, in terms of energy, create any desired formation of M points while avoiding collisions between agents.

Definition 1. The *desired formation* is the set of time-varying vectors $\mathcal{G} = \{\mathbf{p}_j^*(t) \in \mathbb{R}^2 \mid j \in \mathcal{F}\}$. The set \mathcal{G} can be prescribed offline, i.e., by a human designer, or online by a high-level planner.

In this framework, the agents are cooperative and capable of communication within a neighborhood, which we define next.

Definition 2. The *neighborhood* of agent $i \in \mathcal{A}$ is the time-varying set

$$\mathcal{N}_i(t) = \left\{ j \in \mathcal{A} \mid \|\mathbf{p}_i(t) - \mathbf{p}_j(t)\| \leq h \right\},$$

where $h \in \mathbb{R}$ is the sensing and communication horizon of each agent.

Agent $i \in \mathcal{A}$ is also able to measure the relative position of any neighboring agent $j \in \mathcal{N}_i(t)$, i.e., agent i makes partial observations of the global state. We denote the relative position between two agents i and j by the vector

$$\mathbf{s}_{ij}(t) = \mathbf{p}_j(t) - \mathbf{p}_i(t). \quad (8)$$

Each agent $i \in \mathcal{A}$ occupies a closed disk of radius R ; hence, to guarantee safety for agent i we impose the following constraints for all agents $i \in \mathcal{A}$, $j \in \mathcal{N}_i(t)$, $j \neq i$,

$$\|\mathbf{s}_{ij}(t)\| \geq 2R, \quad \forall t \in [t_i^0, t_i^f], \quad (9)$$

$$h \gg 2R. \quad (10)$$

Condition (9) is our safety constraint, which ensures that no two agents collide. We also impose the strict form of (9) pairwise to

all goals in the desired formation, \mathcal{G} . Eq. (10) is a system-level constraint which ensures agents are able to detect each other prior to a collision.

In our modeling framework we impose the following assumptions:

Assumption 1. The state $\mathbf{x}_i(t)$ for each agent $i \in \mathcal{A}$ is perfectly observed and there is negligible communication delay between the agents.

Assumption 1 is required to evaluate the idealized performance of the generated optimal solution. In general, this assumption may be relaxed by formulating a stochastic optimal control problem to generate agent trajectories.

Assumption 2. The energy cost of communication is negligible; the only energy consumption is in the form of (7).

The strength of this assumption is application dependent. For cases with long-distance communications or high data rates, the trade-off between communication and motion costs can be controlled by varying the sensing and communicating radius, h , of the agents.

To solve the desired formation problem, we first relax the inter-agent collision avoidance constraint to decouple the agent trajectories. This decoupling reduces the problem from a single mixed-integer program to a coupled pair of binary and quadratic programs, which we solve sequentially. This decoupling is common in the literature [22,23], and usually does not affect the outcome of the assignment problem.

Next, we present some preliminary results before decomposing the desired formation problem into the two subproblems, minimum-energy goal assignment and trajectory generation.

2.1. Preliminaries

First we consider that any agent $i \in \mathcal{A}$ obeys double-integrator dynamics, (1)–(2), and has an energy model with the form of (7). Then, we let the state and control trajectories of agent i be unconstrained, i.e., relax (3), (4), and (9). In this case, if i is traveling between two fixed states, the unconstrained minimum-energy trajectory is given by the following system of linear equations:

$$\mathbf{u}_i(t) = \mathbf{a}_i t + \mathbf{b}_i, \quad (11)$$

$$\mathbf{v}_i(t) = \frac{\mathbf{a}_i}{2} t^2 + \mathbf{b}_i t + \mathbf{c}_i, \quad (12)$$

$$\mathbf{p}_i(t) = \frac{\mathbf{a}_i}{6} t^3 + \frac{\mathbf{b}_i}{2} t^2 + \mathbf{c}_i t + \mathbf{d}_i, \quad (13)$$

where \mathbf{a}_i , \mathbf{b}_i , \mathbf{c}_i , and \mathbf{d}_i are constant vectors of integration. The derivation of (11)–(13) is straightforward and can be found in [33].

Thus, the energy consumed for any unconstrained trajectory of agent $i \in \mathcal{A}$ at time t traveling towards the goal $j \in \mathcal{F}$ is given by

$$\begin{aligned} E_i^j(t) &= \int_t^{t_f^i} \|\mathbf{u}_i(\tau)\|^2 d\tau = (t_f^3 - t^3) \left(\frac{a_{i,x}^2 + a_{i,y}^2}{3} \right) \\ &\quad + (t_f^2 - t^2) \left(a_{i,x} b_{i,x} + a_{i,y} b_{i,y} \right) \\ &\quad + (t_f - t) \left(\frac{b_{i,x}^2 + b_{i,y}^2}{2} \right), \end{aligned} \quad (14)$$

where $t \in [t_i^0, t_i^f]$, and $\mathbf{a}_i = [a_{i,x}, a_{i,y}]^T$, $\mathbf{b}_i = [b_{i,x}, b_{i,y}]^T$ are the coefficients of (11).

Next, we present the interaction dynamics between agents. To resolve any conflict between agents, we consider the following

objectively measurable constants: (1) neighborhood size, (2) energy required to reach a goal, and (3) agent index, which may be arbitrarily assigned. Each of these quantities has an associated indicator function for comparing two agents $i, j \in \mathcal{A}$, $j \neq i$,

$$\mathbb{1}_{ij}^{\mathcal{N}}(t) := \begin{cases} 1 & |\mathcal{N}_i(t)| > |\mathcal{N}_j(t)|, \\ 0 & |\mathcal{N}_i(t)| \leq |\mathcal{N}_j(t)|, \end{cases} \quad (15)$$

$$\mathbb{1}_{ij}^{\mathcal{E}}(t) := \begin{cases} 1 & E_i(t) > E_j(t), \\ 0 & E_i(t) \leq E_j(t), \end{cases} \quad (16)$$

$$\mathbb{1}_{ij}^{\mathcal{A}}(t) := \begin{cases} 1 & i > j, \\ 0 & i < j. \end{cases} \quad (17)$$

Next, we define the interaction dynamics by combining (15)–(17) into a single indicator function.

Definition 3. We define the *interaction dynamics* between any agent $i \in \mathcal{A}$ and another agent $j \in \mathcal{N}_i(t)$, $j \neq i$ as

$$\begin{aligned} \mathbb{1}_{ij}^{\mathcal{C}}(t) &= \mathbb{1}_{ij}^{\mathcal{N}}(t) + (1 - \mathbb{1}_{ij}^{\mathcal{N}}(t))(1 - \mathbb{1}_{ji}^{\mathcal{N}}(t)) \\ &\quad \left(\mathbb{1}_{ij}^{\mathcal{E}}(t) + (1 - \mathbb{1}_{ij}^{\mathcal{E}}(t))(1 - \mathbb{1}_{ji}^{\mathcal{E}}(t))\mathbb{1}_{ij}^{\mathcal{A}}(t) \right), \end{aligned} \quad (18)$$

where $\mathbb{1}_{ij}^{\mathcal{C}} = 1$ implies agent i has priority over agent j , and $\mathbb{1}_{ij}^{\mathcal{C}} = 0$ implies that agent j has priority over agent i .

The interaction dynamics are instantaneously and noiselessly measured and communicated by each agent under **Assumption 1**. Whenever two agents have a conflict (i.e., share an assigned goal, or have overlapping assignments) (18) is used to impose an order on the agents such that higher priority agents act first.

Remark 1. For any pair of agents $i \in \mathcal{A}$, $j \in \mathcal{N}_i(t)$, $j \neq i$, it is always true that $\mathbb{1}_{ij}^{\mathcal{C}}(t) = 1 - \mathbb{1}_{ji}^{\mathcal{C}}(t)$, i.e., the outcome of the interaction dynamics (18) is always unambiguous, and therefore it imposes an order on any pair of agents.

Remark 1 can be proven by simply enumerating all cases of (15)–(17).

3. Optimal goal assignment

The optimal solution of the assignment problem must assign each agent to a goal such that the total unconstrained energy cost, given by (14), is minimized. In our framework, each agent $i \in \mathcal{A}$ only has information about the positions of its neighbors, $j \in \mathcal{N}_i(t)$, and the goal positions prescribed by \mathcal{G} . Agent i derives the goal assignment using a binary matrix $\mathbf{A}_i(t)$, which we define next.

Definition 4. For each agent $i \in \mathcal{A}$, we define the *assignment matrix*, $\mathbf{A}_i(t)$, as an $|\mathcal{N}_i(t)| \times M$ matrix with binary elements. The elements of $\mathbf{A}_i(t)$ map each agent to exactly one goal, and each goal to no more than one agent.

The assignment matrix for agent $i \in \mathcal{A}$ assigns all agents in $\mathcal{N}_i(t)$ to goals by considering the cost (14). We discuss the details of the optimal assignment problem later in this section.

Next we define the prescribed goal, which determines how each agent $i \in \mathcal{A}$ assigns itself a goal.

Definition 5. We define the *prescribed goal* for agent $i \in \mathcal{A}$ as the goal assigned to agent i by the rule,

$$\mathbf{p}_i^a(t) \in \{ \mathbf{p}_k \in \mathcal{G} \mid a_{ik} = 1, a_{ik} \in \mathbf{A}_i(t), k \in \mathcal{F} \}, \quad (19)$$

where $\mathbf{A}_i(t)$ is the assignment matrix, and the right hand side is a singleton set, i.e., agent i is assigned to exactly one goal.

Next, we present the goal assignment algorithm in terms of some agent $i \in \mathcal{A}$. However, as this framework is cooperative, each step is performed by all individuals simultaneously.

In some cases, multiple agents may select the same prescribed goal. This may occur when two agents $i \in \mathcal{A}, j \in \mathcal{N}_i(t), j \neq i$ have different neighborhoods and use conflicting information to solve their local assignment problem. This motivates the introduction of competing agents, which we define next.

Definition 6. For agent $i \in \mathcal{A}$, we define the set of *competing agents* as

$$C_i(t) = \left\{ k \in \mathcal{N}_i(t) \mid \mathbf{p}_k^a(t) = \mathbf{p}_i^a(t) \right\}.$$

When $|C_i| > 1$ there are at least two agents, $i, j \in \mathcal{N}_i(t)$ which are assigned to the same goal. In this case, all but one agent in $C_i(t)$ must be *permanently banned* from the goal $\mathbf{p}_i^a(t)$. Next, we define the banned goal set.

Definition 7. The *banned goal set* for agent $i \in \mathcal{A}$ is defined as

$$\mathcal{B}_i(t) = \left\{ g \in \mathcal{F} \mid \mathbf{p}_i^a(\tau) = \mathbf{p}_g(\tau) \in \mathcal{G}, \left(\prod_{j \in C_i(\tau), j \neq i} \mathbb{1}_{ij}^c(\tau) \right) = 0, \exists \tau \in [t_i^0, t] \right\}, \quad (20)$$

i.e., the set of all goals which agent $i \in \mathcal{A}$ had a conflict over and did not have priority per [Definition 3](#).

Banning is achieved by applying (20) to all agents $j \in C_i(t)$ whenever $|C_i(t)| > 1$. After the banning step is completed, agent $i \in \mathcal{A}$ checks if the size of $\mathcal{B}_i(t)$ has increased. If so, agent i increases the value of t_i^f by

$$t_i^f = t + T, \quad (21)$$

where t is the current time, and $T \in \mathbb{R}_{>0}$ is a system parameter. This allows agent i a sufficient amount of time to reach its new goal.

Next, each agent broadcasts its new set of banned goals to all of its neighbors. Any agent who was banned from $C_i(t)$ assigns itself to a new goal. However, this may cause new agents to enter $C_i(t)$ as they are banned from other goals. To ensure each agent $j \in \mathcal{N}_i(t)$ is assigned to a unique goal, the assignment and banning steps are iterated until the condition

$$|C_j(t)| = 1, \quad \forall j \in \mathcal{N}_i(t), \quad (22)$$

is satisfied. For a given neighborhood $\mathcal{N}_i(t)$, $i \in \mathcal{A}$, some number of agents will be assigned to the goal $g \in \mathcal{F}$. After the first banning step, all agents except the one which was assigned to goal g are permanently banned and may never be assigned to it again. If additional agents are assigned to g , then this process will repeat for at most $N - 1$ iterations. Afterwards every goal g will have at most one agent from $\mathcal{N}_i(t)$ assigned to it. Thus, we will have $|C_j(t)| = 1$ for all $j \in \mathcal{N}_i(t)$ for every $i \in \mathcal{A}$.

We enforce the banned goals through a constraint on the assignment problem, which follows.

Problem 1 (Goal Assignment). Each agent $i \in \mathcal{A}$ selects its prescribed goal ([Definition 5](#)) by solving the following binary program

$$\min_{a_{jk} \in \mathbf{A}_i} \left\{ \sum_{j \in \mathcal{N}_i(t)} \sum_{k \in \mathcal{F}} a_{jk} E_j^k(t) \right\} \quad (23)$$

subject to:

$$\sum_{j \in \mathcal{F}} a_{jk} = 1, \quad k \in \mathcal{N}_i(t), \quad (24)$$

$$\sum_{k \in \mathcal{N}_i(t)} a_{jk} \leq 1, \quad j \in \mathcal{F}, \quad (25)$$

$$a_{jk} = 0, \quad \forall j \in \mathcal{B}_k(t), k \in \mathcal{N}_i(t), \quad (26)$$

$$a_{jk} \in \{0, 1\}.$$

This process is repeated by each agent, $i \in \mathcal{A}$, until (22) is satisfied for all $j \in \mathcal{N}_i(t)$.

As the conflict condition in [Problem 1](#) explicitly depends on the neighborhood of agent $i \in \mathcal{A}$, [Problem 1](#) may need to be recalculated each time the neighborhood of agent i switches. The assignments generated by [Problem 1](#) are guaranteed to bring each agent to a unique goal; we show this with the help of [Assumption 3](#) and [Lemma 1](#).

Assumption 3. For every agent $i \in \mathcal{A}$, for all $t \in [t_i^0, t_i^f]$, the inequality $|\mathcal{F} \setminus \bigcup_{j \in \mathcal{N}_i(t)} \mathcal{B}_j(t)| \geq |\mathcal{N}_i(t)|$ holds.

[Assumption 3](#) is a condition that is sufficient but not necessary to prove convergence of our proposed optimal controller. Intuitively, [Assumption 3](#) only requires that one agent does not ban many agents from many goals. Due to the minimum-energy nature of our framework, this scenario is unlikely; additionally, permanent banning may be relaxed to temporary banning in a way that [Assumption 3](#) is always satisfied.

Lemma 1 (Solution Existence). Under [Assumption 3](#), the feasible region of [Problem 1](#) is nonempty for agent i .

Proof. Let the set of goals available to all agents in the neighborhood of agent $i \in \mathcal{A}$ be denoted by the set

$$\mathcal{L}_i(t) = \{p \in \mathcal{F} \mid p \notin \mathcal{B}_j(t), \forall j \in \mathcal{N}_i(t)\}. \quad (27)$$

Let the injective function $w : \mathcal{A} \rightarrow \mathcal{F}$ map each agent to a goal. By [Assumption 3](#), $|\mathcal{N}_i(t)| \leq |\mathcal{L}_i(t)|$, thus a function w exists. As w is injective, the imposed mapping must satisfy (24) and (25). Likewise, $\mathcal{L}_i(t) \cap \mathcal{B}_j(t) = \emptyset$ for all $j \in \mathcal{N}_i(t)$. Thus, w must satisfy (26). Therefore, the mapping imposed by the function w is a feasible solution to [Problem 1](#). \square

Next, we show that for a sufficiently large value of T the convergence of all agents to goals is guaranteed.

Theorem 1 (Assignment Convergence). Under [Assumption 3](#), for sufficiently large values of the initial t_i^f and T , and if the energy-optimal trajectories for agent $i \in \mathcal{A}$ never increase the unconstrained energy cost (14), then t_i^f must have an upper bound for all $i \in \mathcal{A}$.

Proof. Let $\{g_n\}_{n \in \mathbb{N}}$ be the sequence of goals assigned to agent $i \in \mathcal{A}$ by the solution of [Problem 1](#). By [Lemma 1](#), $\{g_n\}_{n \in \mathbb{N}}$ must not be empty, and the elements of this sequence are natural numbers bounded by $1 \leq g_n \leq M$. Thus, the range of this sequence is compact, and the sequence must be either (1) finite, or (2) convergent, or (3) periodic.

(1) For a finite sequence there is nothing to prove, as t_i^f is upper bounded by $t_{i,\text{initial}}^f + MT$.

(2) Under the discrete metric, an infinite convergent sequence requires that there exists $N \in \mathbb{N}_{>0}$ such that $g_n = p$ for all $n > N$ for some formation index $p \in \mathcal{F}$. This reduces to case 1, as t_i^f does not increase for repeated assignments to the same goal.

(3) By the Bolzano–Weierstrass Theorem, an infinite non-convergent sequence $\{g_n\}_{n \in \mathbb{N}}$ must have a convergent subsequence, i.e., agent i is assigned to some subset of goals $\mathcal{I} \subseteq \mathcal{F}$ infinitely many times with some constant number of intermediate assignments, P_g , for each goal $g \in \mathcal{I}$. Necessarily, $\mathcal{I} \cap \mathcal{B}_i(t) = \emptyset$ for all $t \in [t_i^0, t_i^f]$ from the construction of the banned goal set.

This implies that, by the update method of t_i^f , the position of all goals, $g \in \mathcal{I}$ must only be considered at time t_i^f .

This implies that the goals available to agent i , i.e., $\mathcal{I} = \mathcal{F} \setminus \mathcal{B}_i(t_i^f)$, must be shared between $n > 0$ other periodic agents. Hence, at some time t_1 a goal, $g \in \mathcal{I}$, must be an optimal assignment for agent i , a non optimal assignment at time $t_2 > t_1$ and an optimal assignment again at time t_3 which corresponds to the P_g^{th} assignment. As $t_3 > t_2 > t_1$, the energy required to move agent i to goal g satisfies

$$E_i^g(t_1) \leq E_i^k(t_1), \quad (28)$$

$$E_i^k(t_2) \leq E_i^g(t_2), \quad (29)$$

$$E_i^g(t_3) \leq E_i^k(t_3), \quad (30)$$

for any other goal $k \in \mathcal{I}, k \neq g$. Therefore, for agent i to follow an energy optimal trajectory under our premise, it must never increase the energy required to reach is assigned goal, which implies

$$E_i^g(t_1) \geq E_i^g(t_2), \quad (31)$$

$$E_i^k(t_2) \geq E_i^k(t_3), \quad (32)$$

this implies

$$E_i^k(t_1) \geq E_i^k(t_3), \quad (33)$$

which is only possible if agent i simultaneously approaches all goals $k \in \mathcal{I}$. This implies that goals g and k are arbitrarily close, which violates the minimum spacing requirements of the goals; therefore no such periodic behavior may exist. \square

Note that [Theorem 1](#) bounds the arrival time of agent $i \in \mathcal{A}$ to any goal $g \in \mathcal{F}$. A similar bound may be found for the total energy consumed, i.e.,

$$E_i^g(t) \leq \frac{1}{2}(t_{i,\text{initial}}^f + MT) \cdot \max\{|u_{\min}|, |u_{\max}|\}^2.$$

Next, we formulate the optimal trajectory generation problem for each agent and prove that the resulting trajectories always satisfy the premise of [Theorem 1](#).

4. Optimal trajectory generation

After the goal assignment is complete, each agent must generate a collision-free trajectory to their assigned goal. The trajectories must minimize the agent's total energy cost subject to dynamic, boundary, and collision constraints. The initial and final state constraints for each agent $i \in \mathcal{A}$ are given by

$$\mathbf{p}_i(t_i^0) = \mathbf{p}_i^0, \quad \mathbf{v}_i(t_i^0) = \mathbf{v}_i^0, \quad (34)$$

$$\mathbf{p}_i(t_i^f) = \mathbf{p}_i^g(t_i^f), \quad \mathbf{v}_i(t_i^f) = \dot{\mathbf{p}}_i^g(t_i^f), \quad (35)$$

where the conditions at t_i^f come from the solution of [Problem 1](#).

Whenever an agent must steer to avoid collisions, we will apply the agent interaction dynamics ([Definition 3](#)) to impose an order on the agents such that lower priority agents must steer to avoid the higher priority ones. Thus, we may simplify the collision avoidance constraint for agent $i \in \mathcal{A}$ to

$$\|\mathbf{s}_{ij}(t)\| \geq 2R, \quad \forall j \in \{k \in \mathcal{A} \mid \mathbb{1}_{ik}(t) = 0\}, \quad (36)$$

$$\forall t \in [t_i^0, t_i^f],$$

which will always guarantee safety for all agents.

We may then formulate the decentralized optimal trajectory generation problem.

Problem 2. For each agent $i \in \mathcal{A}$, find the optimal control input, $\mathbf{u}_i(t)$, which minimizes the energy consumption of agent i and satisfies its boundary conditions and safety constraints.

$$\min_{\mathbf{u}_i(t)} \frac{1}{2} \int_{t_i^0}^{t_i^f} \|\mathbf{u}_i(t)\|^2 dt \quad (37)$$

subject to: (1), (2), (34), (35), and (36).

By imposing an order on the agents, we can show that the solution of [Problem 2](#) will always satisfy the premise of [Theorem 1](#). First, [Lemma 2](#) shows that an unconstrained trajectory must never increase the energy required to reach a goal.

Lemma 2. For any agent $i \in \mathcal{A}$, following the unconstrained trajectory, the energy cost (14) required to reach a fixed goal $g \in \mathcal{F}$ is not increasing.

Proof. We may write the derivative of (14) along an unconstrained trajectory as

$$\begin{aligned} \frac{dE_i^g(t)}{dt} &= \lim_{\delta \rightarrow 0} \frac{1}{\delta} \left(\int_{t+\delta}^{t_i^f} \|\mathbf{u}_i(\tau)\|^2 d\tau - \int_t^{t_i^f} \|\mathbf{u}_i(\tau)\|^2 d\tau \right) \\ &= - \lim_{\delta \rightarrow 0} \frac{1}{\delta} \int_t^{t+\delta} \|\mathbf{u}_i(\tau)\|^2 d\tau, \end{aligned} \quad (38)$$

which is never positive. Therefore, (14) is never increasing. \square

Next, we introduce [Theorem 2](#), which proves the premise of [Theorem 1](#) is always satisfied by any trajectory which is a feasible solution to [Problem 2](#).

Theorem 2. If a solution to [Problem 2](#) exists for all agents, then [Theorem 1](#) is satisfied as long as [Assumption 3](#) holds.

Proof. The case when any agent $i \in \mathcal{A}$ is moving with an unconstrained trajectory is covered by [Lemma 2](#), so we focus on the case when any of the safety constraints are active.

Let $\mathcal{K} \subseteq \mathcal{A}$ be a group of agents which all have their safety constraint active over some interval $t \in [t_1, t_2]$. By [Definition 3](#), there exists some $i \in \mathcal{K}$ such that $\mathbb{1}_{ij}^c(t) = 1$ for all $j \in \mathcal{K}, j \neq i$. Therefore, agent i satisfies [Lemma 2](#) and always moves toward its assigned goal by [Theorem 1](#).

Next, consider agent $j \in \mathcal{V} \setminus \{i\}$ such that $\mathbb{1}_{jk}^c(t) = 1$ for all $k \in \mathcal{K} \setminus \{i\}$. As agent j may never be assigned to the same goal as i , there must exist some time $t_j < \min\{t_i^f, t_j^f\}$ such that $\|\mathbf{s}_{ij}(t_j)\| > 2R$ by the goal spacing rules. Thus, agent j will move with an unconstrained trajectory for all $t \in [t_j, t_j^f]$. The above steps can be recursively applied until only a single agent remains, which follows an unconstrained trajectory for some finite time interval. This satisfies the premise of [Theorem 1](#). \square

4.1. Hamiltonian analysis

We solve [Problem 2](#) by applying Hamiltonian analysis. We will follow the standard methodology used in optimal control theory for problems with interior point constraints. First, we start with the unconstrained solution, given by (11)–(13). If this solution violates the state, control, or safety constraints, we piece it together with solutions corresponding to the violated constraint. These two arcs yield a set of algebraic equations that must be solved simultaneously using the boundary conditions (34) and (35) and interior conditions between the arcs. If the resulting trajectory, which includes the optimal switching time between the arcs, still violates any constraints, the new solution must be pieced together with a third arc corresponding to the new violated constraint. This process is repeated until no constraints

are violated, which yields the energy-optimal state trajectory for each agent $i \in \mathcal{A}$.

The case where only the control and state constraints become active has been extensively studied in [33]. Thus, we will relax the state and control constraints and only consider the safety constraint in this section. Next, we analyze the case where the collision avoidance constraint becomes active.

First, the safety constraint (9) must be derived until the control input $\mathbf{u}_i(t)$ appears. To ensure smoothness in the derivatives we use the equivalent squared form of (9). This yields

$$\mathbf{N}_i(t, \mathbf{x}_i(t)) = \begin{bmatrix} 4R^2 - \mathbf{s}_{ij}(t) \cdot \mathbf{s}_{ij}(t) \\ -\mathbf{s}_{ij}(t) \cdot \dot{\mathbf{s}}_{ij}(t) \\ -\mathbf{s}_{ij}(t) \cdot \ddot{\mathbf{s}}_{ij}(t) - \dot{\mathbf{s}}_{ij}(t) \cdot \dot{\mathbf{s}}_{ij}(t) \end{bmatrix} \leq \mathbf{0}, \quad (39)$$

where the first two elements of $\mathbf{N}_i(t)$ are the *tangency conditions* which must be satisfied at the start of a constrained arc, while the third element is augmented to the unconstrained Hamiltonian. The Hamiltonian is

$$H_i = \frac{1}{2} \|\mathbf{u}_i(t)\|^2 + \lambda_i^p(t) \cdot \mathbf{v}_i(t) + \lambda_i^v(t) \cdot \mathbf{u}_i(t) - \sum_{j \in \mathcal{N}_i} \mu_{ij}(t) \left(\mathbf{s}_{ij}(t) \cdot \ddot{\mathbf{s}}_{ij}(t) + \dot{\mathbf{s}}_{ij}(t) \cdot \dot{\mathbf{s}}_{ij}(t) \right), \quad (40)$$

where $\lambda_i^p(t)$, $\lambda_i^v(t)$ are the position and velocity co-vectors, and $\mu_{ij}(t)$ is an inequality Lagrange multiplier with values

$$\mu_{ij}(t) = \begin{cases} > 0 & \text{if } \mathbf{s}_{ij}(t) \cdot \ddot{\mathbf{s}}_{ij}(t) + \dot{\mathbf{s}}_{ij}(t) \cdot \dot{\mathbf{s}}_{ij}(t) = 0, \\ 0 & \text{if } \mathbf{s}_{ij}(t) \cdot \ddot{\mathbf{s}}_{ij}(t) + \dot{\mathbf{s}}_{ij}(t) \cdot \dot{\mathbf{s}}_{ij}(t) > 0. \end{cases} \quad (41)$$

To solve (40) for agent $i \in \mathcal{A}$, we consider two cases:

1. all agents $j \in \mathcal{N}_i(t)$ satisfy $\mu_{ij} = 0$, and
2. any agent $j \in \mathcal{N}_i(t)$ satisfies $\mu_{ij} > 0$.

This results in a piecewise trajectory, where the first case corresponds to following an unconstrained trajectory, while the second corresponds to some collision avoidance constraints becoming active. Our task is to derive the form of the constrained arcs, then optimally piece the constrained and unconstrained arcs together. This will result in the optimal trajectory for agent i .

Next, we present the solution to the constrained case and the optimal time to transition between the two cases.

4.2. Constrained solution

As with the assignment problem, the constrained solution is presented in terms of some agent $i \in \mathcal{A}$. However, the steps given here are performed simultaneously by all agents. First, we define the conflict set.

Definition 8. We define the *conflict set* for agent $i \in \mathcal{A}$ at time $t \in [t_i^0, t_i^f]$ as

$$\mathcal{V}_i(t) = \left\{ j \in \mathcal{N}_i(t) \mid \mu_{ij}(t) > 0, \mathbb{1}_{ij}^c = 0 \right\}, \quad (42)$$

i.e., the set of all agents which i may collide with and have a higher priority than agent i .

Agent i must then steer to avoid all agents $j \in \mathcal{V}_i(t)$. To solve for the constrained trajectory we introduce Lemma 3, which considers the case when $|\mathcal{V}_i(t)| > 1$.

Lemma 3 (Collision-Avoidance Constraints). *Let any agent $i \in \mathcal{A}$ be moving along a collision constrained arc such that $|\mathcal{V}_i(t)| > 1$. Then, only feasible trajectory for agent i is to remain in contact with all agents $j \in \mathcal{V}_i(t)$ until agent i exits this constrained arc. This unique trajectory is therefore optimal.*

Proof. For any two agents $j, k \in \mathcal{V}_i(t)$ we have $\|\mathbf{s}_{ij}(t)\| = 2R$ and $\|\mathbf{s}_{ik}\| = 2R$. This implies that the points $p_i(t)$, $p_j(t)$, and $p_k(t)$ must form an isosceles triangle with two edges of length $2R$ and base of length $2R \leq \|\mathbf{s}_{jk}(t)\| \leq 4R$. Therefore, the only feasible trajectory of agent i is to maintain the isosceles triangle between i , j , and k . As this is the only one feasible trajectory for agent i , it must be the optimal trajectory. \square

Note that Lemma 3 holds for a single constrained arc. As such, agent $i \in \mathcal{A}$ may exit to a different constrained arc with a new set $\mathcal{V}_i(t)$ or may exit to an unconstrained arc.

Next we consider the case where agent i moves along a constrained arc with $|\mathcal{V}_i(t)| = 1$. First, we use the Euler-Lagrange conditions to obtain,

$$\frac{\partial H_i}{\partial \mathbf{u}_i} = 0, \quad (43)$$

$$-\dot{\lambda}_i = \frac{\partial H_i}{\partial \mathbf{x}_i}. \quad (44)$$

Application of (43) to (40) yields

$$\mathbf{u}_i(t) = -\lambda_i^v(t) - \sum_{j \in \mathcal{V}_i(t)} \mu_{ij}(t) \mathbf{s}_{ij}(t), \quad (45)$$

while (44) results in

$$-\dot{\lambda}_i^p(t) = \sum_{j \in \mathcal{V}_i(t)} \mu_{ij}(t) \ddot{\mathbf{s}}_{ij}(t), \quad (46)$$

$$-\dot{\lambda}_i^v(t) = \lambda_i^p(t) + \sum_{j \in \mathcal{V}_i(t)} \mu_{ij}(t) \dot{\mathbf{s}}_{ij}(t). \quad (47)$$

As $|\mathcal{V}_i(t)| = 1$, the optimality condition and Euler-Lagrange equations become

$$\mathbf{u}_i(t) = -\lambda_i^v(t) - \mu_{ij}(t) \mathbf{s}_{ij}(t), \quad (48)$$

$$-\dot{\lambda}_i^p(t) = \mu_{ij}(t) \ddot{\mathbf{s}}_{ij}(t), \quad (49)$$

$$-\dot{\lambda}_i^v(t) = \lambda_i^p(t) + \mu_{ij}(t) \dot{\mathbf{s}}_{ij}(t), \quad (50)$$

where $j \in \mathcal{V}_i(t)$. We denote the relative speed between two agents i and j as

$$a_{ij}(t) = \|\dot{\mathbf{s}}_{ij}(t)\|. \quad (51)$$

Next, we define a new orthonormal basis for \mathbb{R}^2 .

Definition 9. For an agent $i \in \mathcal{A}$ satisfying $|\mathcal{V}_i(t)| = 1$, over some nonzero interval $t \in [t_1, t_2]$, where $a_{ij}(t) \neq 0$, we define the *contact basis* as

$$\hat{p}_{ij}(t) = \frac{\mathbf{s}_{ij}(t)}{\|\mathbf{s}_{ij}(t)\|} = \frac{\mathbf{s}_{ij}(t)}{2R}, \quad (52)$$

$$\hat{q}_{ij}(t) = \frac{\dot{\mathbf{s}}_{ij}(t)}{\|\dot{\mathbf{s}}_{ij}(t)\|} = \frac{\dot{\mathbf{s}}_{ij}(t)}{a_{ij}(t)}, \quad (53)$$

where $\hat{p}_{ij}(t) \cdot \hat{q}_{ij}(t) = 0$ by (39), and both vectors are unit length. Thus, (52) and (53) constitute an orthonormal basis for \mathbb{R}^2 .

Next, we find the projection of $\ddot{\mathbf{s}}_{ij}(t)$ onto the new contact basis. From (39) we have

$$\ddot{\mathbf{s}}_{ij}(t) \cdot \hat{p}_{ij}(t) = \ddot{\mathbf{s}}_{ij}(t) \cdot \frac{\mathbf{s}_{ij}(t)}{2R} = \frac{-a_{ij}^2(t)}{2R}. \quad (54)$$

We apply integration by parts to find the $\hat{q}_{ij}(t)$ component of $\ddot{\mathbf{s}}_{ij}(t)$. First,

$$\int \ddot{\mathbf{s}}_{ij}(t) \cdot \dot{\mathbf{s}}_{ij}(t) dt = \dot{\mathbf{s}}_{ij}(t) \cdot \dot{\mathbf{s}}_{ij}(t) - \int \dot{\mathbf{s}}_{ij}(t) \cdot \ddot{\mathbf{s}}_{ij}(t) dt, \quad (55)$$

which implies

$$\int \dot{\mathbf{s}}_{ij}(t) \cdot \dot{\mathbf{s}}_{ij}(t) dt = \frac{1}{2} \dot{\mathbf{s}}_{ij}(t) \cdot \dot{\mathbf{s}}_{ij}(t) = \frac{1}{2} a_{ij}^2(t). \quad (56)$$

Taking a time derivative of (56) yields

$$\ddot{\mathbf{s}}_{ij}(t) \cdot \dot{\mathbf{s}}_{ij}(t) = a_{ij}(t) \dot{a}_{ij}(t). \quad (57)$$

Next we present [Theorem 3](#), which gives the optimal trajectory for agent i whenever $a_{ij}(t) = 0$ over any nonzero interval while the safety constraint is active.

Theorem 3. For any agents $i \in \mathcal{A}$ and $j \in \mathcal{V}_i(t)$, if $a_{ij}(t) = 0$ over some nonzero interval $t \in [t_1, t_2]$, then the optimal trajectory for agent i is to follow $\mathbf{u}_i(t) = \mathbf{u}_j(t)$ for all $t \in [t_1, t_2]$.

Proof. By definition we have $a_{ij}(t) = |\dot{\mathbf{s}}_{ij}(t)| = 0$. This implies $\dot{\mathbf{s}}_{ij}(t) = \mathbf{0}$, and therefore $\mathbf{v}_i(t) = \mathbf{v}_j(t)$ for all $t \in [t_1, t_2]$. Thus $\mathbf{u}_i(t) = \mathbf{u}_j(t)$ for all $t \in [t_1, t_2]$. \square

Thus, for any agent $i \in \mathcal{A}$ which has an active safety constraint with some agent $j \in \mathcal{V}_i(t)$, [Theorem 3](#) provides the optimal control input for agent i in the case that $a_{ij}(t) = 0$ over a nonzero time interval. If $a_{ij}(t) = 0$ for at single instant $t \in \mathbb{R}$, then the optimal solution at that instant must enforce continuity of $a_{ij}(t)$ and the constraint $\mathbf{s}_j(t) \cdot \mathbf{s}_j(t) = 4R^2$.

Finally, we may project the dynamics of agent i onto the basis given in [Definition 9](#) and solve for the optimal trajectory when $a_{ij}(t) \neq 0$. Next, we will use (54) and (57) to project $\ddot{\mathbf{s}}_{ij}(t)$ onto the contact basis,

$$\ddot{\mathbf{s}}_{ij}(t) = \begin{bmatrix} -\frac{a_{ij}^2(t)}{2R} \\ \dot{a}_{ij}(t) \end{bmatrix} \cdot \begin{bmatrix} \hat{p}_{ij}(t) \\ \hat{q}_{ij}(t) \end{bmatrix}, \quad (58)$$

which we use to solve for the time derivatives of (52) and (53). First,

$$\frac{d}{dt} \hat{p}_{ij}(t) = \frac{\dot{\mathbf{s}}_{ij}(t)}{2R} = \frac{a(t)}{2R} \hat{q}_{ij}(t). \quad (59)$$

Then, by the quotient rule,

$$\begin{aligned} \frac{d}{dt} \hat{q}_{ij}(t) &= \frac{\ddot{\mathbf{s}}_{ij}(t) a_{ij}(t) - \dot{\mathbf{s}}_{ij}(t) \dot{a}_{ij}(t)}{a_{ij}^2(t)} \\ &= \frac{a_{ij}(t) \left(-a^2(t) \frac{1}{2R} \hat{p}_{ij}(t) + \dot{a}_{ij}(t) \hat{q}_{ij}(t) \right)}{a_{ij}^2(t)} \\ &\quad - \frac{\dot{\mathbf{s}}_{ij}(t) \dot{a}_{ij}(t)}{a_{ij}^2(t)} \\ &= -\frac{a_{ij}(t)}{2R} \hat{p}_{ij}(t). \end{aligned} \quad (60)$$

From (8), we may now write $\ddot{\mathbf{s}}_{ij}(t)$ projected on to the contact basis ([Definition 9](#)) as

$$\begin{aligned} \ddot{\mathbf{s}}_{ij}(t) &= \mathbf{u}_j(t) + \lambda_i^v(t) + \mu_i(t) \mathbf{s}_{ij}(t) \\ &= \left(\mathbf{u}_j(t) + \lambda_i^v(t) \right) \begin{bmatrix} \hat{p}_{ij}(t) \\ \hat{q}_{ij}(t) \end{bmatrix} + \mu_i(t) \begin{bmatrix} 2R \\ 0 \end{bmatrix}. \end{aligned} \quad (61)$$

Next, we set (58) equal to (61) and rewrite it as a system of scalar equations,

$$\lambda_i^v(t) \cdot \hat{p}_{ij}(t) = -\frac{a_{ij}^2(t)}{2R} - 2R \mu_{ij}(t) - \mathbf{u}_j(t) \cdot \hat{p}_{ij}(t), \quad (62)$$

$$\lambda_i^v(t) \cdot \hat{q}_{ij}(t) = \dot{a}_{ij}(t) - \mathbf{u}_j(t) \cdot \hat{q}_{ij}(t). \quad (63)$$

Taking the time derivative of (62) yields

$$\begin{aligned} \frac{a_{ij}(t)}{2R} \lambda_i^v(t) \cdot \hat{q}_{ij}(t) + \dot{\lambda}_i^v(t) \cdot \hat{p}_{ij}(t) &= -\frac{a_{ij}(t) \dot{a}_{ij}(t)}{R} \\ &\quad - 2R \dot{\mu}_{ij}(t) - \dot{\mathbf{u}}_j(t) \cdot \hat{p}_{ij}(t) - \frac{a_{ij}(t)}{2R} \mathbf{u}_j(t) \cdot \hat{q}_{ij}(t). \end{aligned} \quad (64)$$

We then substitute (50) and (63) into (64), which yields

$$\begin{aligned} \lambda_i^p(t) \cdot \hat{p}_{ij}(t) &= \frac{3a_{ij}(t) \dot{a}_{ij}(t)}{2R} + 2R \dot{\mu}_{ij}(t) \\ &\quad + \dot{\mathbf{u}}_j(t) \cdot \hat{p}_{ij}(t). \end{aligned} \quad (65)$$

Taking a time derivative of (63) yields,

$$\begin{aligned} -\frac{a_{ij}(t)}{2R} \lambda_i^v(t) \cdot \hat{p}_{ij}(t) + \dot{\lambda}_i^v(t) \cdot \hat{q}_{ij}(t) &= \ddot{a}_{ij}(t) \\ &\quad - \dot{\mathbf{u}}_j(t) \cdot \hat{q}_{ij}(t) + \frac{a_{ij}(t)}{2R} \mathbf{u}_j(t) \cdot \hat{p}_{ij}(t), \end{aligned} \quad (66)$$

and substituting (50) and (62) into (66), yields

$$\lambda_i^p(t) \cdot \hat{q}_{ij}(t) = \dot{\mathbf{u}}_j(t) \cdot \hat{q}_{ij}(t) - \ddot{a}_{ij}(t) + \frac{a_{ij}^3(t)}{4R^2}. \quad (67)$$

We then take an additional time derivative of (65) and (67). This yields

$$\begin{aligned} \dot{\lambda}_i^p(t) \cdot \hat{p}_{ij}(t) &= -\frac{a_{ij}(t)}{2R} \lambda_i^p(t) \cdot \hat{q}_{ij}(t) \\ &\quad + \frac{3}{2R} (\dot{a}_{ij}^2(t) + a_{ij}(t) \ddot{a}_{ij}(t)) + 2R \dot{\mu}_{ij}(t) \\ &\quad + \frac{a_{ij}(t)}{2R} \dot{\mathbf{u}}_j(t) \cdot \hat{q}_{ij}(t) + \ddot{\mathbf{u}}_j(t) \cdot \hat{p}(t), \end{aligned} \quad (68)$$

$$\begin{aligned} \dot{\lambda}_i^p(t) \cdot \hat{q}_{ij}(t) &= \frac{a_{ij}(t)}{2R} \lambda_i^p(t) \cdot \hat{p}_{ij}(t) \\ &\quad - \ddot{a}_{ij}(t) + \frac{3}{4R^2} a_{ij}^2(t) \dot{a}_{ij}(t) \\ &\quad - \frac{a_{ij}(t)}{2R} \dot{\mathbf{u}}_j(t) \cdot \hat{p}_{ij}(t) + \ddot{\mathbf{u}}_j(t) \cdot \hat{q}(t). \end{aligned} \quad (69)$$

Substituting (49), (65), and (67) into (68) and (69) yields a system of nonlinear ordinary differential equations,

$$\begin{aligned} \frac{a_{ij}^2(t)}{2R} \mu_{ij}(t) + \frac{a_{ij}^4(t)}{8R^3} &= 2R \dot{\mu}_{ij}(t) + \frac{4}{2R} a_{ij}(t) \ddot{a}_{ij}(t) \\ &\quad + \frac{3}{2R} \dot{a}_{ij}(t) + \ddot{\mathbf{u}}_j(t) \cdot \hat{p}_{ij}(t), \end{aligned} \quad (70)$$

$$\begin{aligned} a_{ij}(t) \dot{\mu}_{ij}(t) + \dot{a}_{ij}(t) \mu_{ij}(t) + \ddot{\mathbf{u}}_j(t) \cdot \hat{q}_{ij}(t) \\ &\quad + \frac{6}{4R^2} a_{ij}^2(t) \dot{a}_{ij}(t) = \ddot{a}_{ij}(t). \end{aligned} \quad (71)$$

Thus, for any constrained trajectory to be energy-optimal, it must be a solution of (70) and (71). In general, finding a solution is rare, since both equations are nonlinear and (71) is third order. Therefore, our approach will be to impose $\dot{a}_{ij}(t) = 0$ over the constraint arc, which is a locally optimal trajectory per [Theorem 3](#).

Thus, the remaining unknown quantities are: the junction time when agent i transitions from the unconstrained to constrained arc, t_1 , the junction time that agent i exits from the constrained arc, t_2 , and the initial orientation of the vector $\mathbf{s}_{ij}(t_1)$. These quantities are coupled to the proceeding and following unconstrained arcs by the jump conditions [34],

$$\mathbf{x}_i(t_1^-) = \mathbf{x}_i(t_1^+), \quad (72)$$

$$\lambda_i^T(t_1^-) = \lambda_i^T(t_1^+) + \mathbf{v}_i^T \frac{\partial N_i(t, \mathbf{x}_i(t))}{\partial \mathbf{x}(t)} \Bigg|_{t=t_1}, \quad (73)$$

$$H(t_1^-) = H(t_1^+) + \mathbf{v}_i^T \frac{\partial N_i(t, \mathbf{x}_i(t))}{\partial t} \Bigg|_{t=t_1}, \quad (74)$$

$$H(t_2^-) = H(t_2^+), \quad (75)$$

$$\mathbf{x}_i(t_2^-) = \mathbf{x}_i(t_2^+), \quad (76)$$

$$\lambda_i(t_2^-) = \lambda_i(t_2^+), \quad (77)$$

where the superscripts t^- and t^+ correspond to the left and right-side limits of t , respectively. Thus, t_1^- and t_2^+ correspond to the unconstrained arcs at the junctions t_1 and t_2 . Likewise, t_1^+ and t_2^- correspond to the junctions where agent i enters and exits the constrained arc, respectively. The constant vector \mathbf{v}_i is given by [34]

$$\mathbf{v}_i = \begin{bmatrix} -\frac{\mathbf{u}_i(t_1^-) \cdot \mathbf{u}_i(t_1^-)}{2(\mathbf{s}_{ij}(t_1^-) \cdot \mathbf{v}_i(t_1^-))} \\ 0 \end{bmatrix}. \quad (78)$$

For the case when agent i 's trajectory has only a single constrained arc, (72)–(77) coupled with the initial and final conditions, (35) and (34), constitute 26 scalar equations to solve for the 26 unknowns (8+8+8 constants of integration + 2 transition times). When additional constrained arcs become active, additional jump conditions must be computed using (72)–(77). The entire system of equations is then solved simultaneously to yield the energy-optimal trajectory for agent i .

4.3. The Full Solution to Problem 2

So far, we have provided the unconstrained and safety-constrained arcs with a relaxation of the state and control constraints. An extension to the fully-constrained case for agent $i \in \mathcal{A}$ is straightforward, and the solution for every possible case is outlined below.

1. No constraints are active: Agent i will follow an unconstrained trajectory.
2. Only one safety constraint is active: Agent i will follow the trajectory outlined in Section 4.2.
3. More than one safety constraint is active: By Lemma 3, the unique trajectory of agent i will be defined by the active constraints.
4. Only one state/control constraint is active: This reduces to a steering problem, where agent i follows a known velocity profile and must arrive at a target state along a minimum-energy path [35].
5. One safety and one state/control constraint are active: Agent i must follow the path imposed by the safety constraint with a speed profile determined by the state/control constraint. The unique solution to this problem is, therefore, optimal.

The state trajectory of agent $i \in \mathcal{A}$ must be a piecewise-continuous function consisting of the five possible cases. These segments are then pieced together using the optimality conditions. We presented the optimality conditions for collision avoidance in Section 4.2; the conditions for the state and control constraints are derived in [33] and [34].

5. Simulation results

To provide insight into the behavior of the agents, a series of simulations were performed with $M = N = 10$ agents and a time parameter of $T = 10$ s. The simulations were run for $t = 20$ s or until all agents reach their assigned goal, whichever occurred later. The center of the formation moved with a velocity of $\mathbf{v}_{cg} = [0.15, 0.35]$ m/s; the leftmost and rightmost three goals each move with an additional periodic velocity of $[0.125 \cos 0.75t, 0]$ m/s relative to the formation. Videos of the simulation results can be found at <https://sites.google.com/view/ud-ids-lab/omas>.

The minimum separating distance between agents, total energy consumed, and maximum velocity for the unconstrained solutions to Problem 2 are all given as a function of the horizon in Table 1. The energy consumption only considers the energy

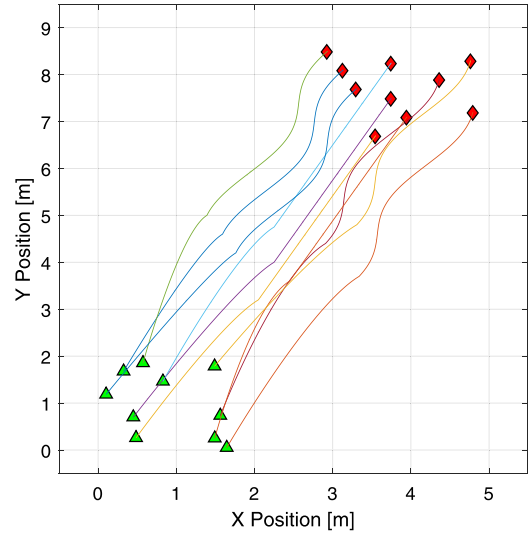


Fig. 1. Simulation result for the centralized case. Goals which minimize the unconstrained trajectories are assigned to the agents once at t_1^0 .

Table 1

Numerical results for $N = 10$ agents and $M = 10$ goals for various sensing distances.

h [m]	min. separation [cm]	Energy [J/kg]	t_f [s]	Total bans
∞	25.25	0.85	20	0
1.60	1.64	1.10	20	4
1.50	1.60	1.17	20	24
1.40	2.01	1.96	23.3	31
1.30	0.33	1670	26.05	36
1.20	0.65	866	25.35	34
1.10	1.05	5370	26.85	40
1.00	1.96	7609	30.65	35
0.95	3.12	3149	25.05	27
0.75	1.37	6.87	20	35
0.50	0.27	692.0	26.65	35

required to reach the goal, which, in this case, was significantly lower than the energy required to maintain the formation. The trajectory of each agent over time is given in Figs. 1–3 for varying sensing horizon values. Although the trajectories may appear to cross in Figs. 1–3, they are only crossing in space and not in time.

The performance of our algorithm is strongly affected by how much information is available to each agent. This is a function of the sensing horizon, initial states of the agents, and the desired formation shape. Generally, better overall performance requires the agents to have more information. However, it is not apparent what information is necessary; in fact, the results in Table 1 generally show no correlation between energy consumption and sensing horizon.

The trade-off for more information is in the computational and sensing load imposed on each agent. As an agent observes more of the system (via sensing, communication, or memory), the computational burden to solve the assignment and trajectory generation problems also increases. However, this computational cost does not necessarily result in improved system performance, as demonstrated in Table 1.

6. Conclusion

In this paper, we proposed a decentralized framework for moving a group of autonomous agents into a desired formation. The only information required a priori is the positions of the goals in a global coordinate frame. We provided guarantees of

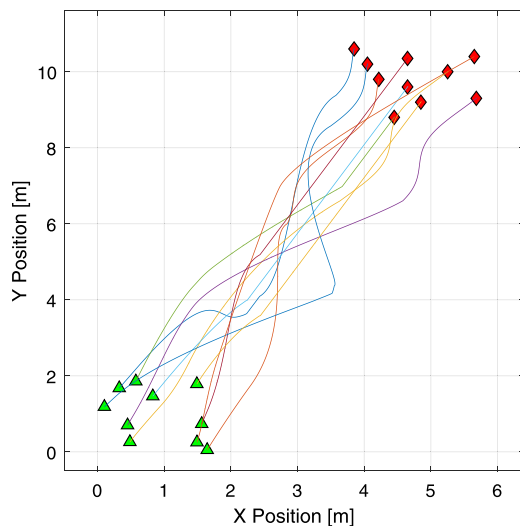


Fig. 2. Simulation result for $h = 1.30$ m. The agents do not start with a globally unique assignment, and several agents must re-route partway through the simulation. Although the trajectories cross in space they do not cross in time.

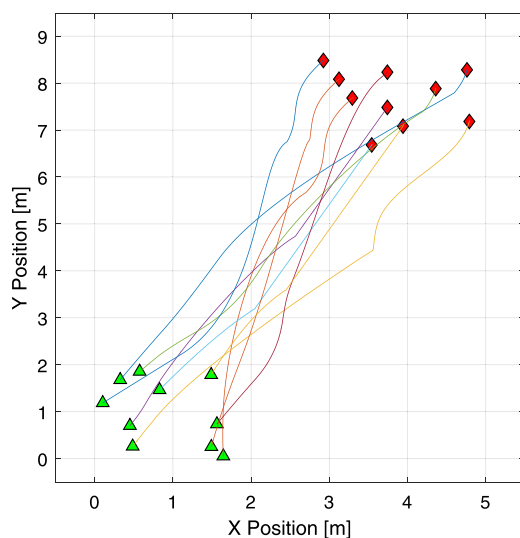


Fig. 3. Simulation result for $h = 0.75$ m. Although the horizon for this case is smaller than in Fig. 2, the system dynamics happen to result in more efficient trajectories overall.

convergence under [Assumption 3](#). Our method leveraged a set of agent interaction dynamics, which allowed a decentralized calculation of the priority order for agents. We also derived local energy-optimal trajectories for constrained and unconstrained paths and presented the conditions for optimality in the form of a boundary-value nonlinear ordinary differential equation. The resulting optimal controller was validated in MATLAB using ten agents and ten dynamic moving goals with varying values of the sensing radius.

One area for future research is finding a relaxation of [Assumption 3](#) or another fundamental condition to replace it. Deriving additional locally-optimal solutions to (70) and (71) is another research direction. A relaxation of the jump conditions to find approximately optimal trajectories that can be generated in real-time is another area which is under active research [36].

Finally, analysis of the system parameter T using the fully constrained agent trajectories is another potential direction for future research. This includes methods to optimally select T or to estimate its magnitude based on the state and observations of each agent.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Logan Beaver: Writing - original draft. **Andreas A. Malikopoulos:** Writing - review & editing.

Acknowledgment

This research was supported by the ARPAAE's NEXTCAR program under the award number DE-AR0000796.

References

- [1] A.A. Malikopoulos, A duality framework for stochastic optimal control of complex systems, *IEEE Trans. Automat. Control* 61 (10) (2016) 2756–2765.
- [2] A.A. Malikopoulos, Centralized stochastic optimal control of complex systems, in: Proceedings of the 2015 European Control Conference, 2015, pp. 721–726.
- [3] A.A. Malikopoulos, V. Maroulas, J. Xiong, A multiobjective optimization framework for stochastic control of complex systems, in: Proceedings of the 2015 American Control Conference, 2015, pp. 4263–4268.
- [4] A.A. Malikopoulos, Equilibrium control policies for markov chains, in: 50th IEEE Conference on Decision and Control and European Control Conference, 2011, pp. 7093–7098.
- [5] H. Oh, A.R. Shirazi, C. Sun, Y. Jin, Bio-inspired self-organising multi-robot pattern formation: A review, *Robot. Auton. Syst.* 91 (2017) 83–100.
- [6] M. Brambilla, E. Ferrante, M. Birattari, M. Dorigo, Swarm robotics: A review from the swarm engineering perspective, *Swarm Intell.* 7 (1) (2013) 1–41.
- [7] A.M. Turing, The chemical basis of morphogenesis, *Phil. Trans. R. Soc. B* 237 (641) (1952) 37–72.
- [8] K.-K. Oh, M.-C. Park, H.-S. Ahn, A survey of multi-agent formation control, *Automatica* 53 (2015) 424–440.
- [9] C.W. Reynolds, Flocks, herds and schools: A distributed behavioral model, *Comput. Graph.* 21 (4) (1987) 25–34.
- [10] J. Guo, Z. Lin, M. Cao, G. Yan, Adaptive control schemes for mobile robot formations with triangularised structures, *IET Control Theory Appl.* 4 (9) (2010) 1817–1827.
- [11] Y. Hanada, G. Lee, N.Y. Chong, Adaptive flocking of a swarm of robots based on local interactions, in: IEEE Swarm Intelligence Symposium, 2007, pp. 340–347.
- [12] Y. Song, J.M. O’Kane, Forming repeating patterns of mobile robots: A provably correct decentralized algorithm, in: IEEE International Conference on Intelligent Robots and Systems, Vol. 2016–Novem, 2016, pp. 5737–5744.
- [13] G. Lee, Y.C. Nak, Self-configurable mobile robot swarms with hole repair capability, in: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2008.
- [14] J.O. Swartling, I. Shames, K.H. Johansson, D.V. Dimarogonas, Collective circumnavigation, *Unmanned Syst.* 02 (03) (2014) 219–229.
- [15] Z. Lin, M. Broucke, B. Francis, Local control strategies for groups of mobile autonomous agents, *IEEE Trans. Automat. Control* 49 (4) (2004) 622–629.
- [16] R. Olfati-Saber, J.A. Fax, R.M. Murray, Consensus and cooperation in networked multi-agent systems, *Proc. IEEE* 95 (1) (2007) 215–233.
- [17] J. Wang, M. Xin, Integrated optimal formation control of multiple unmanned aerial vehicles, *IEEE Trans. Control Syst. Technol.* 21 (5) (2013) 1731–1744.
- [18] X. Sun, C.G. Cassandras, Optimal dynamic formation control of multi-agent systems in environments with obstacles, in: IEEE Conference on Decision and Control, 2015, pp. 2359–2364.
- [19] H. Xu, R.G.C. Carrillo, distributed near optimal flocking control for multiple unmanned aircraft systems, in: International Conference on Unmanned Aerial Vehicles, ICUAS, Denver, CO, USA, 2015, pp. 879–885.

- [20] R. Rajasree, Optimal formation control of unmanned aerial vehicles with obstacle avoidance, in: *International Conference on Control, Instrumentation, Communication and Computational Technologies*, 2010.
- [21] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A.E. Eiben, T. Vicsek, *Optimized Flocking of Autonomous Drones in Confined Environments*, Tech. rep., 2018.
- [22] M. Turpin, K. Mohta, N. Michael, V. Kumar, Goal assignment and trajectory planning for large teams of aerial robots, *Proc. Robot.: Sci. Syst.* 37 (2013) 401–415.
- [23] D. Morgan, G.P. Subramanian, S.-J. Chung, F.Y. Hadaegh, Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming, *Int. J. Robot. Res.* 35 (10) (2016) 1261–1285.
- [24] M. Rubenstein, C. Ahler, R. Nagpal, Kilobot: A low cost scalable robot system for collective behaviors, in: *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, Saint Paul, MN, 2012.
- [25] A. Dave, A. Malikopoulos, Structural results for decentralized stochastic control with a word-of-mouth communication, in: *2020 American Control Conference*, 2020, [arXiv:1809.08712](https://arxiv.org/abs/1809.08712), in press.
- [26] M. Turpin, N. Michael, V. Kumar, Concurrent assignment and planning of trajectories for large teams of interchangeable robots, in: *2013 IEEE International Conference Robotics and Automation, ICRA, Karlsruhe, Germany*, 2013, pp. 842–848.
- [27] B. Chalaki, A.A. Malikopoulos, An optimal coordination framework for connected and automated vehicles in two interconnected intersections, *Proceedings of 2019 IEEE Conference on Control Technology and Applications*, 2019, pp. 888–893.
- [28] H. Ma, D. Harabor, P.J. Stuckey, J. Li, S. Koenig, Searching with consistent prioritization for multi-agent path finding, in: *33rd AAAI Conference on Artificial Intelligence*, 2019, pp. 7643–7650.
- [29] W. Wu, S. Bhattacharya, A. Prorok, Multi-Robot Path Deconfliction through Prioritization by Path Prospects, Tech. Rep., 2019.
- [30] G. Sartoretti, J. Kerr, Y. Shi, G. Wagner, T.K.S. Kumar, S. Koenig, H. Choset, PRIMAL: Pathfinding via reinforcement and imitation multi-agent learning, *IEEE Robot. Autom. Lett.* 4 (3) (2018) 2378–2385.
- [31] H. Witsenhausen, A counterexample in stochastic optimum control, *SIAM J. Control* 6 (1) (1968) 131–147.
- [32] M. Turpin, N. Michael, V. Kumar, CAPT: Concurrent assignment and planning of trajectories for multiple robots, *Int. J. Robot. Res.* 33 (1) (2014) 98–112.
- [33] A.A. Malikopoulos, C.G. Cassandras, Y.J. Zhang, A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections, *Automatica* 93 (April) (2018) 244–256.
- [34] A.E. Bryson Jr., Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control*, John Wiley and Sons, 1975, p. 496.
- [35] I.M. Ross, *A Primer on Pontryagin's Principle in Optimal Control*, second ed., Collegiate Publishers, San Francisco, 2015, p. 336.
- [36] A.M. L. E. Beaver, An optimal control approach to flocking, in: *Proceedings of the 2020 American Control Conference Denver, CO, 2020*, [arXiv:1909.12330](https://arxiv.org/abs/1909.12330), in press.