Energy Efficient Rendezvous Algorithms for Heterogeneous Multi-Agent Systems

Banu Kabakulak, *Member, IEEE*

***Abstract*—This paper addresses the challenge of energy-efficient rendezvous for heterogeneous multi-agent systems (MASs) operating within a convex polytope that may contain convex denied regions. We formulate centralized CRM−D and distributed DRM$_i$−D models to identify energy-optimal feasible rendezvous points, considering both unconstrained and denied-region scenarios. For each model, we develop exact solution methods for the unconstrained case and heuristic approaches to handle denied regions effectively. Additionally, we propose synchronous C−MAS and asynchronous D−MAS controllers based on the solutions of the mathematical models. The controllers aim to minimize energy consumption by generating low-power control commands and smoothing agent trajectories to reduce abrupt velocity and acceleration changes. We conduct extensive simulations to assess the robustness and energy efficiency of the proposed controllers in guiding a MAS to the computed rendezvous points under varying system dynamics. The results demonstrate that both C−MAS and D−MAS effectively achieve consensus on the rendezvous point while maintaining robust performance even in the presence of denied regions and agent failures. These findings highlight the scalability and practical applicability of the proposed controllers in dynamic and resource-constrained MAS environments.**

***Index Terms*—Centralized and distributed control, energy efficiency, multi-agent systems, rendezvous problem.**

I. Introduction

RESEARCH on the intelligent control of multi-agent systems (MASs) has gained considerable attention in recent decades, driven by applications in artificial satellites, UAVs, UGVs, and UNVs [1]. A *heterogeneous* MAS consists of *agents* like autonomous robots, each with distinct capabilities. The diverse functionalities in heterogeneous MASs enable more adaptable formations and efficient mission execution, making them increasingly relevant for complex tasks. However, controlling MASs is challenging due to nonlinear dynamics, uncertainties, external disturbances, and potential sensor and actuator failures.

The intelligent control of MASs encompasses key research areas such as rendezvous [2], consensus [3], aggregation, formation [4], flocking [5], and tracking [6], while ensuring collision avoidance [7] and obstacle navigation [8]. Advanced MAS control enables complex cooperative missions, including area mapping [9], reconnaissance and surveillance [10], telecommunications, and disaster-related search and rescue [11], supporting effective mission execution [12, 13].

The multi-agent *rendezvous* problem involves agents converging at a common, unspecified location. *Consensus*, derived from distributed computing, refers to agents agreeing on a specific value or function [14]. The two problems are equivalent when agents can move freely [15].

The rendezvous problem in MASs can be classified into *deterministic* and *stochastic* approaches, with deterministic methods relying on predefined protocols and exact positional knowledge [16], while stochastic methods address uncertainties through probabilistic convergence techniques [17]. It can also be categorized as *synchronous* or *asynchronous* based on whether agents move simultaneously or independently [18]. Additional classifications include *discrete* vs. *continuous* space, *centralized* vs. *distributed* control [19], and *constrained* vs. *unconstrained* regions, where agents navigate obstacles or restricted zones to reach a common point [20–22].

Rendezvous problems in MASs are addressed using diverse solution approaches, including *analytical*, *algorithmic*, *optimization-based*, *control-theoretic*, *graph-theoretic*, and *learning-based* methods. Analytical approaches use methods like centroid calculation, geometric averaging, and Lyapunov functions to determine convergence points under idealized conditions [23], but often ignore practical constraints like delays or obstacles. Algorithmic methods, including consensus algorithms [24], gossip protocols, and probabilistic random walks, enable decentralized coordination through local interactions [25]. Models like Vicsek [26] and leader-follower [27, 28] frameworks also guide agents to a common point in synchronous and asynchronous settings.

Optimization-based methods frame rendezvous as an optimization problem, using techniques like linear programming, quadratic programming, and gradient descent to minimize metrics such as energy use, convergence time, or control effort [29]. Potential and control barrier functions are also applied for obstacle avoidance and collision prevention [30]. Control-theoretic approaches employ PID control, adaptive control, and sliding mode control to direct agents to the rendezvous point while maintaining stability and robustness [31].

Graph-theoretic methods model agent interactions as communication graphs, using tools like spanning trees, connectivity matrices, and consensus protocols to ensure convergence under dynamic topologies and intermittent links [32]. These approaches are effective in distributed settings with limited communication. Learning-based methods, such as reinforcement learning [33], neural networks, and multi-agent deep Q-networks [34], address complex rendezvous problems in uncertain or adversarial environments.

B. Kabakulak is with the Department of Industrial Engineering, İstanbul Bilgi University, İstanbul, Türkiye. e-mail: banu.kabakulak@boun.edu.tr.

This research makes the following key contributions:

1) We define an energy-efficient rendezvous problem for heterogeneous MASs operating within a convex polytope that may include convex denied regions.
2) We develop centralized `CRM-D` and distributed `DRM`$_i$`-D` models to identify energy-optimal feasible rendezvous points, proposing exact solutions for unconstrained cases and heuristic methods for scenarios involving denied regions.
3) We design synchronous `C-MAS` and asynchronous `D-MAS` controllers that leverage the model solutions to generate energy-efficient control commands and smooth agent trajectories.
4) We conduct simulations to evaluate the robustness and energy efficiency of the proposed controllers in guiding a MAS to the desired rendezvous point.

The paper is structured as follows: Sec. II defines the notation and provides preliminary theory, while Sec. III describes the rendezvous problem over a convex region. Sec. IV presents the centralized and distributed rendezvous formulations, including both exact and heuristic solution methods, followed by the corresponding MAS controllers in Sec. V. Simulation results assessing controller robustness are provided in Sec. VI. Finally, Sec. VII concludes the paper with insights and potential future research directions.

II. Preliminaries and Notation

A. Basic Graph Theory

A *directed graph* $\mathscr{G}=(V,E)$, where $V=\{i_1,...,i_N\}$ represents the set of nodes and $E\subseteq V\times V$ denotes the set of arcs, is used to model the communication topology of a MAS. An agent can share information with other agents within its *communication range*. That is, for $i_1,i_2\in V$ there exists an arc $(i_1,i_2)\in E$ if agent i_2 can receive information from agent i_1. Note that in a heterogeneous MAS, the existence of $(i_1,i_2)\in E$ does not necessarily imply $(i_2,i_1)\in E$, as the agents may have non-identical communication ranges. The set of *neighbors* $\mathscr{N}_i=\{j\in V\mid(j,i)\in E\}$ represents the agents that can send information to agent i.

The vector $\mathbf{1}_N$ represents the unit column vector and $\mathbf{1}_j$ is a column vector in $\mathbb{R}^N$ with only its jth entry equal to one. The notation $\|.\|$ indicates Euclidean norm and for vectors $p,q\in\mathbb{R}^n$

$$\|p-q\|^2=\sum_{\tau=1}^{n}(p_\tau-q_\tau)^2 \tag{1}$$

is the squared Euclidean distance.

B. Constrained Convex Non-Linear Programming

A set $\mathscr{S}$ is *convex* if and only if, for all $x_1,x_2\in\mathscr{S}$ and $0\le\lambda\le1$,

$$\lambda x_1+(1-\lambda)x_2\in\mathscr{S}. \tag{2}$$

A *convex combination* of the vectors $x_1,x_2,...,x_l\in\mathbb{R}^n$ is defined as $y=\sum_{k=1}^{l}\lambda_k x_k$, where the coefficients $\lambda_1,...,\lambda_l$ satisfy $\lambda_k\ge0$ for all k and $\sum_{k=1}^{l}\lambda_k=1$ [35, 36].

Let $f:D_f\to\mathbb{R}$ be a continuously twice differentiable function defined on a convex domain $D_f\subseteq\mathbb{R}^n$. Then, f is a *convex function* if and only if its Hessian matrix $\nabla^2 f(x)$ is positive semi-definite for all $x\in D_f$. In a *constrained non-linear programming* (CNLP) problem, the objective function f is minimized subject to the equality constraints $h_i(x)=0$ for $i=1,...,m$, and the inequality constraints $g_k(x)\le0$ for $k=1,...,p$ as described in the system (3)–(5).

$$\min_{x\in D_f}\; f(x) \tag{3}$$
$$\text{s.t.}\quad h_i(x)=0 \quad i=1,...,m \quad : \pi_i \tag{4}$$
$$g_k(x)\le0 \quad k=1,...,p \quad : \mu_k \tag{5}$$

Corollary 1. *A CNLP is a* convex programming *(CP) problem if*

1) *the objective f is a convex function,*
2) *the equality constraints h_i are linear functions,*
3) *the inequality constraints g_k are convex functions.*

An inequality constraint, say $g_k(x)\le0$ for $k=1,...,p$, is *active (tight)* at a given point x^* if $g_k(x^*)=0$ for $k\in\mathscr{I}$. The constraint is *inactive* if $g_k(x^*)<0$ at that point. Let $\mathscr{S}_c$ be the surface defined by the active constraints as

$$\mathscr{S}_c=\begin{Bmatrix}x\in\mathbb{R}^n\mid h_i(x)=0,\ i=1,...,m,\\ g_k(x)=0,\ k\in\mathscr{I}\end{Bmatrix}. \tag{6}$$

x^* is a *regular point* on $\mathscr{S}_c$ if the gradients $\nabla h_i(x^*)$ for $i=1,...,m$ and $\nabla g_k(x^*)$ for $k\in\mathscr{I}$ are *linearly independent*.

In a CNLP, a *stationary point* $x^*\in D_f$ is a regular point on $\mathscr{S}_c$ and can be found by solving the *Karush-Kuhn-Tucker (KKT) first order conditions* as given in Definition 1. Here, π_i and μ_k are the dual variables associated with the equality and inequality constraints, respectively.

Definition 1. (Karush-Kuhn-Tucker First Order Conditions)

1) *Dual Feasibility*, $(\nabla_x L(x,\pi,\mu)=0,\ \mu_k\ge0)$

$$\nabla f(x)+\sum_{i=1}^{m}\pi_i\nabla h_i(x)+\sum_{k=1}^{p}\mu_k\nabla g_k(x)=0 \tag{7}$$
$$\mu_k\ge0 \quad k=1,...,p \tag{8}$$

2) *Complementary Slackness*

$$\mu_k\, g_k(x)=0 \quad k=1,...,p \tag{9}$$

3) *Primal Feasibility*, $(\nabla_\lambda L(x,\pi,\mu)=0)$

$$h_i(x)=0 \quad i=1,...,m \tag{10}$$
$$g_k(x)=0 \quad k\in\mathscr{I} \tag{11}$$
$$g_k(x)<0 \quad k\notin\mathscr{I} \tag{12}$$

Corollary 2. *In a CP problem, the KKT first order conditions are* sufficient *to conclude that a stationary point x^* is a* local minimum *of the objective function f, subject to the constraints $h_i(x)=0$ for $i=1,...,m$, and $g_k(x)\le0$ for $k=1,...,p$.*

III. System Description

In a MAS consisting of K types of agents, let N denote the number of agents, with their indices collected in the set $\mathscr{N} = \{1,...,N\}$. Equations (13) and (14) describe a double integrator model, offering a simplified yet effective approximation of agents' dynamics.

$$\dot{p}_i = v_i \quad i \in \mathscr{N} \tag{13}$$

$$\ddot{p}_i = u_i \quad i \in \mathscr{N} \tag{14}$$

Here, $u_i \in \mathbb{R}^n$ represents the control input (acceleration), $p_i \in \mathbb{R}^n$ denotes the position, and $v_i \in \mathbb{R}^n$ is the velocity of agent $i \in \mathscr{N}$. The velocity and acceleration of agent i are constrained by the bounds $(\underline{v}_i, \bar{v}_i)$ and $(\underline{a}_i, \bar{a}_i)$, respectively. The power consumption of agent i, denoted as w_i watts, is a known convex function of its instantaneous velocity v_i meter/sec. The state of an agent i is expressed as $[p_i^T, v_i^T, w_i] \in \mathbb{R}^{2n+1}$. Additionally, an agent can communicate with others within a range of ω_i meters.

The region of interest is described as a convex polytope $\mathscr{R} \subseteq \mathbb{R}^n$, defined by M *extreme points*, which are collected in the finite set $\mathscr{M} = \{1,...,M\}$. The polytope $\mathscr{R}$ can be expressed as a convex combination of its extreme points $e_k \in \mathbb{R}^n$ for $k \in \mathscr{M}$, as described in (15).

$$\mathscr{R} = \left\{ \begin{matrix} x \in \mathbb{R}^n \mid x = \sum_{k \in \mathscr{M}} \lambda_k e_k, \sum_{k \in \mathscr{M}} \lambda_k = 1, \\ \lambda_k \geq 0, \ \lambda_k \in \mathbb{R} \end{matrix} \right\} \tag{15}$$

The region $\mathscr{R}$ contains D distinct *denied convex polytopes*, each defined by a finite number of D_l extreme points for $l \in \{1,...,D\}$. Equation (16) defines the lth denied polytope $\mathscr{D}_l$ as a convex combination of its extreme points $\bar{e}_{lk} \in \mathbb{R}^n$, where $l \in \{1,...,D\}$ and $k \in \{1,...,D_l\}$.

$$\mathscr{D}_l = \left\{ \begin{matrix} x \in \mathbb{R}^n \mid x = \sum_{k=1}^{D_l} \bar{\lambda}_{lk} \bar{e}_{lk}, \sum_{k=1}^{D_l} \bar{\lambda}_{lk} = 1, \\ \bar{\lambda}_{lk} \geq 0, \ \bar{\lambda}_{lk} \in \mathbb{R} \end{matrix} \right\} \tag{16}$$

The union set $\mathscr{D} = \cup_{l=1}^{D} \mathscr{D}_l$ represents the set of denied regions within the region $\mathscr{R}$, which is not necessarily a convex set. The ultimate goal is for the MAS to achieve rendezvous at a non-denied position r within the region $\mathscr{R}$ while minimizing the energy consumption of the system.

An agent $i \in \mathscr{N}$ can acquire the state of a local agent $j \in \mathscr{N}_i$ and the local environment through its onboard sensors, such as cameras, IR or LIDAR sensors, and by utilizing multi-hop communication with its neighbors [37]. Let an agent i perceive a *local polytope* $\mathscr{R}^{(i)} \subseteq \mathscr{R}$, defined by the extreme points $e_k^{(i)}$, indexed by $\mathscr{M}^{(i)} = \{1,...,M^{(i)}\} \subseteq \mathscr{M}$. Similarly, $\mathscr{D}_l^{(i)} \subseteq \mathscr{D}_l$ represents the lth *local denied polytope* of agent i, with extreme points $\bar{e}_{lk}^{(i)}$ for $l \in \{1,...,D\}$ and $k \in \{1,...,D_l^{(i)}\}$. The union set $\mathscr{D}^{(i)} = \cup_{l=1}^{D} \mathscr{D}_l^{(i)}$ represents the set of local denied regions for agent i. The regions $\mathscr{R}^{(i)}$ and $\mathscr{D}_l^{(i)}$ define the *local environment* of agent i and can be obtained by rewriting Equations (15) and (16) using the local extreme points $e_k^{(i)}$ and $\bar{e}_{lk}^{(i)}$, respectively.

IV. Mathematical Formulations

A. Rendezvous Models

The *centralized rendezvous model with denied zones* (CRM-D), defined by (17), identifies the energy-optimal rendezvous point $r \in \mathbb{R}^n$ within non-denied zones. This optimization is performed given the positions p_i and instantaneous power consumption w_i, which are determined based on the current velocities v_i of the agents $i \in \mathscr{N}$. The *centralized rendezvous model* (CRM) is derived by relaxing the denied-zone constraints, allowing $r \in \mathscr{R}$ in (17).

$$\min_{r \in \mathscr{R} \setminus \mathscr{D}} \quad \sum_{i \in \mathscr{N}} \frac{w_i}{\|v_i\|} \|p_i - r\|^2 \tag{17}$$

The *distributed rendezvous model for agent i with denied zones* (DRM$_i$-D), as given by (18), determines the energy-optimal local non-denied rendezvous point r_i for $i \in \mathscr{N}$. The *distributed rendezvous model for agent i* (DRM$_i$) is a local model obtained by relaxing the feasible region in (18), such that $r_i \in \mathscr{R}^{(i)}$.

$$\min_{r_i \in \mathscr{R}^{(i)} \setminus \mathscr{D}^{(i)}} \quad \sum_{j \in \mathscr{N}_i} \frac{w_j}{\|v_j\|} \|p_j - r_i\|^2 \tag{18}$$

We observe that, while CRM and DRM$_i$ are CP problems, CRM-D and DRM$_i$-D are not due to their non-convex feasible sets. However, if the MAS communication network is connected, then DRM$_i$-D is equivalent to CRM-D for $i \in \mathscr{N}$ since $\mathscr{N}_i = \mathscr{N}$, $\mathscr{R}^{(i)} = \mathscr{R}$, and $\mathscr{D}^{(i)} = \mathscr{D}$.

B. Analytical Solutions

1) Analytical Solution of CRM: Firstly, we represent the convex CRM problem as

$$\min \quad \sum_{i \in \mathscr{N}} \frac{w_i}{\|v_i\|} \|p_i - r\|^2 \tag{19}$$

s.t.

$$r = \sum_{k \in \mathscr{M}} \lambda_k e_k \tag{20}$$

$$\sum_{k \in \mathscr{M}} \lambda_k = 1 \tag{21}$$

$$\lambda_k \geq 0 \quad k \in \mathscr{M} \tag{22}$$

Then, we replace r in the objective (19) with (20) and rewrite the CRM model as

$$\min \quad \sum_{i \in \mathscr{N}} \frac{w_i}{\|v_i\|} \left\| p_i - \sum_{k \in \mathscr{M}} \lambda_k e_k \right\|^2 \tag{23}$$

s.t.

$$\sum_{k \in \mathscr{M}} \lambda_k - 1 = 0 \qquad : \pi \tag{24}$$

$$-\lambda_k \leq 0 \quad k \in \mathscr{M} \qquad : \mu_k \tag{25}$$

which is a convex CNLP with an equality (24) and M inequality (25) constraints. The dual variables are π and μ_k for $k \in \mathscr{M}$, respectively.

Let $h(\lambda) = \sum_{k \in \mathscr{M}} \lambda_k - 1$, $g_k(\lambda) = -\lambda_k$ for $k \in \mathscr{M}$ and

$$f(\lambda) = \sum_{i \in \mathscr{N}} \frac{w_i}{\|v_i\|} \sum_{\tau=1}^{n} \left(p_{i\tau} - \sum_{k \in \mathscr{M}} \lambda_k e_{k\tau} \right)^2 \tag{26}$$

for $\lambda \in \mathbb{R}^M$. Then, the gradient vectors are $\nabla h(\lambda) = \mathbf{1}_M$ and $\nabla g_k(\lambda) = -\mathbf{1}_k$ for $k \in \mathscr{M}$ and

$$\begin{aligned} \nabla f(\lambda) &= \begin{bmatrix} & & \vdots & & \\ a_{k1} & \ldots & a_{kk'} & \ldots & a_{kM} \\ & & \vdots & & \end{bmatrix} \begin{bmatrix} \vdots \\ \lambda_{k'} \\ \vdots \end{bmatrix} + \begin{bmatrix} \vdots \\ b_k \\ \vdots \end{bmatrix} \\ &= \mathbf{A}\lambda + \mathbf{b} \end{aligned} \tag{27}$$

where

$$b_k = \sum_{i \in \mathscr{N}} \sum_{\tau=1}^{n} -2 \frac{w_i}{\|v_i\|} e_{k\tau} p_{i\tau} \qquad k \in \mathscr{M} \tag{28}$$

are known constants,

$$a_{kk'} = \sum_{i \in \mathscr{N}} \sum_{\tau=1}^{n} 2 \frac{w_i}{\|v_i\|} e_{k\tau} e_{k'\tau} \qquad k, k' \in \mathscr{M} \tag{29}$$

are known coefficients and $\mathbf{A}$ is a symmetric matrix.

Proposition 1. *At a regular point $\lambda \in \mathbb{R}^M$ of the* CRM *model, there is at least one $\lambda_k > 0$ for $k \in \mathscr{M}$.*

Proof. Consider the equality

$$\alpha_0 \nabla h(\lambda) + \alpha_1 \nabla g_1(\lambda) + \ldots + \alpha_M \nabla g_M(\lambda) = 0$$

for α_0, α_1, ..., $\alpha_M \in \mathbb{R}$. We observe that the equality is satisfied for $\alpha_0 = \alpha_1 = \ldots = \alpha_M = 1$, which proves that the vectors $\nabla h(\lambda)$ and $\nabla g_k(\lambda)$ for $k \in \mathscr{M}$ are linearly dependent.

At a regular point λ of CRM, the gradients of the active constraints should be linearly independent. The constraint $h(\lambda) = 0$ is active due to feasibility restrictions (24). Then, there can be at most $(M-1)$ active $g_k(\lambda) = 0$ constraints, i.e., $\lambda_k = 0$, and at least one $\lambda_k > 0$. ■

We can write the KKT first order conditions for CRM as follows:

1) *Dual Feasibility*

$$\sum_{k' \in \mathscr{M}} a_{kk'} \lambda_{k'} + b_k + \pi - \mu_k = 0 \qquad k \in \mathscr{M} \tag{30}$$

$$\mu_k \geq 0 \qquad k \in \mathscr{M} \tag{31}$$

2) *Complementary Slackness*

$$\mu_k \lambda_k = 0 \qquad k \in \mathscr{M} \tag{32}$$

3) *Primal Feasibility*

$$\lambda_1 + \ldots + \lambda_M - 1 = 0 \tag{33}$$

$$\lambda_k = 0 \quad k \in \mathscr{I} \tag{34}$$

$$\lambda_k > 0 \quad k \notin \mathscr{I} \tag{35}$$

A regular point $\lambda \in \mathbb{R}^M$, which satisfies KKT conditions (30)–(35), along with feasible values for $\pi \in \mathbb{R}$ and $\mu \in \mathbb{R}^M$, is a stationary point of CRM. One can represent (30) and (33) as a system of linear equations $\mathbf{A}'x + \mathbf{b}' = \mathbf{0}$, where

$$\mathbf{A}' = \begin{bmatrix} a_{11} & \ldots & a_{1M} & 1 & -1 & & \\ & \ddots & & 1 & & \ddots & \\ a_{M1} & \ldots & a_{MM} & 1 & & & -1 \\ 1 & \ldots & 1 & 0 & 0 & \ldots & 0 \end{bmatrix} \tag{36}$$

$$x = \begin{bmatrix} \lambda_1 \ldots \lambda_M \; \pi \; \mu_1 \ldots \mu_M \end{bmatrix}^T \text{ and } \mathbf{b}' = \begin{bmatrix} b_1 \ldots b_M \; -1 \end{bmatrix}^T. \tag{37}$$

Remark 1. The rank of the matrix $\mathbf{A}'$ is at most $(M+1)$ in (36), and the vector $x \in \mathbb{R}^{2M+1}$ in (37). That is, there are infinitely many solutions to the linear system $\mathbf{A}'x + \mathbf{b}' = \mathbf{0}$.

According to Proposition 1, there is at least one $\lambda_k > 0$ for which $\mu_k = 0$ due to complementary slackness (32). Let there be $|\mathscr{M} \setminus \mathscr{I}|$ values of $\lambda_k > 0$ in (35) and $\mu_k = 0$ in (32). Then, we are left with $|\mathscr{I}|$ values of $\lambda_k = 0$ in (34) and $\mu_k \geq 0$ in (31) and (32).

Hence, one can chose $(M+1)$ basic variables as π, $|\mathscr{M} \setminus \mathscr{I}|$ arbitrary values for λ_k, and the remaining $|\mathscr{I}|$ values for μ_k. Note that if $\mathbf{A}$ in (27) is of full rank, so does the basic submatrix of $\mathbf{A}'$. As a result, the basic variables can be uniquely determined in the system $\mathbf{A}'x + \mathbf{b}' = \mathbf{0}$.

Proposition 2. *Let π, $|\mathscr{M} \setminus \mathscr{I}|$ arbitrary values of λ_k, and the remaining $|\mathscr{I}|$ values of μ_k be chosen as the $(M+1)$ basic variables of the system $\mathbf{A}'x + \mathbf{b}' = \mathbf{0}$. Let $x^* = [\lambda^* \; \pi^* \; \mu^*]^T$ denote the unique solution to this system.*
Then, λ^* is a stationary point of CRM *if $\lambda_k^* \geq 0$ and $\mu_k^* \geq 0$ for all $k \in \mathscr{M}$.*

Proof. We can rewrite the CRM KKT conditions (30)–(35) as follows:

$$\mathbf{A}' \begin{bmatrix} \lambda \\ \pi \\ \mu \end{bmatrix} + \mathbf{b}' = \mathbf{0} \tag{38}$$

$$\mu_k \lambda_k = 0 \quad k \in \mathscr{M} \tag{39}$$

$$\lambda_k, \mu_k \geq 0 \quad k \in \mathscr{M} \tag{40}$$

The selection of basic variables obeys the complementary slackness (39). Therefore, if a feasible solution $x^* = [\lambda^* \; \pi^* \; \mu^*]^T$ of the system (38) satisfies (40), then it is a stationary point of CRM. ■

Remark 2. As CRM is a CP, Corollary 2 implies that a stationary point obtained from (30)–(35) is a local minimum of CRM. Remark 1 indicates that there can be infinitely many stationary points λ^* of CRM. However, CRM has a unique rendezvous point (41) which is the global minimizer r^* of the strictly convex objective function (19).

The CRM-OPT algorithm computes a global minimum of CRM by iteratively solving the linear system (38) until the solution satisfies (40). Initially, we assume that $\mathscr{I} = \emptyset$, i.e., all λ_k variables are basic. If the unique solution x^* of (38) does not satisfy (40), then the violating λ_k variables are switched non-basic, and the corresponding μ_k variables are made basic to ensure compliance with (39).

Proposition 1 establishes that there is at least one basic λ_k for a stationary point. Therefore, the number of variable shifts in Steps 4–10 is bounded by at most $(M-1)$ times. In Step 2, Gaussian elimination can solve the system of linear equations in $\mathscr{O}(M^3)$ time [38]. Consequently, the CRM-OPT algorithm has a worst-case runtime of $\mathscr{O}(M^4)$ in polynomial time, where M is the number of extreme points of $\mathscr{R}$. The complexity of CRM-OPT is substantially better than that of the active set algorithm, a standard quadratic programming solver, which executes with a worst-case complexity of $\mathscr{O}(M^3 2^M)$ [39].

Algorithm 1 A Global Minimum of CRM (CRM-OPT)

```
Input: An instance of CRM
Output: A global minimum of CRM
    Initialization : Let $\mathscr{B}$ be the set of basic variables,
    $\bar{\mathscr{B}}$ be the set of non-basic variables.
 1: Let $\mathscr{B} = \{\pi, \lambda_1, ..., \lambda_M\}$, $\bar{\mathscr{B}} = \{\mu_1, ..., \mu_M\}$.
 2: Set the variables in $\bar{\mathscr{B}}$ to zero and solve the system (38)
    for the variables in $\mathscr{B}$ uniquely.
    Let $x^* = [\lambda^* \ \pi^* \ \mu^*]^T$ be the solution.
 3: if $x^*$ violates (40) then
 4:     for $k \in \mathscr{M}$ do
 5:         if $\lambda_k < 0$ then
 6:             $\mathscr{B} \leftarrow \mathscr{B} \setminus \{\lambda_k\}$ and $\bar{\mathscr{B}} \leftarrow \bar{\mathscr{B}} \cup \{\lambda_k\}$
 7:             $\mathscr{B} \leftarrow \mathscr{B} \cup \{\mu_k\}$ and $\bar{\mathscr{B}} \leftarrow \bar{\mathscr{B}} \setminus \{\mu_k\}$
 8:             Go to Step 2.
 9:         end if
10:     end for
11: end if
12: return $\lambda^*$
```

Let $\lambda^* = [\lambda_1^*, ..., \lambda_M^*]^T$ denote an optimal solution of CRM obtained using CRM-OPT. The global optimal rendezvous point r^* for CRM is given by

$$r^* = \sum_{k \in \mathscr{M}} \lambda_k^* e_k \tag{41}$$

where e_k for $k \in \mathscr{M}$ are the extreme points of $\mathscr{R}$.

Corollary 3. *The optimal rendezvous point* r^* *of the unconstrained* CRM *is the weighted average of the initial agent positions* p_i, *i.e.,*

$$\mathrm{r}^* = \frac{\sum\limits_{i \in \mathscr{N}} \frac{w_i}{\|v_i\|} p_i}{\sum\limits_{i \in \mathscr{N}} \frac{w_i}{\|v_i\|}}. \tag{42}$$

Moreover, if $\mathrm{r}^* \in \mathscr{R}$, *then it is the optimal rendezvous point* r^* *for* CRM.

Example 1. Fig. 1 illustrates an instance of CRM with $N = 3$ agents, where the rectangular $\mathscr{R}$ is defined by $M = 4$ extreme points. The initial agent positions p_i, power consumptions w_i for $i \in \{1, 2, 3\}$ at instantaneous velocity $\|v_i\| = 1$ meter/sec, and extreme points e_k for $k \in \{1, 2, 3, 4\}$ are shown in Fig. 1.

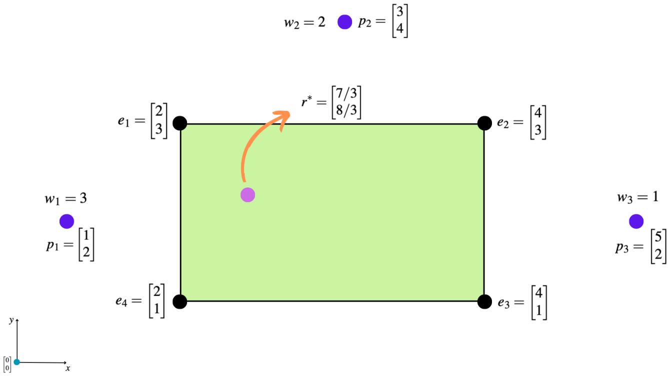
$w_2 = 2$
$p_2 = \begin{bmatrix}3\\4\end{bmatrix}$
$r^* = \begin{bmatrix}7/3\\8/3\end{bmatrix}$
$e_1 = \begin{bmatrix}2\\3\end{bmatrix}$
$e_2 = \begin{bmatrix}4\\3\end{bmatrix}$
$w_1 = 3$
$p_1 = \begin{bmatrix}1\\2\end{bmatrix}$
$w_3 = 1$
$p_3 = \begin{bmatrix}5\\2\end{bmatrix}$
$e_4 = \begin{bmatrix}2\\1\end{bmatrix}$
$e_3 = \begin{bmatrix}4\\1\end{bmatrix}$
y
$\begin{bmatrix}0\\0\end{bmatrix}$
x

Fig. 1. A CRM instance in $\mathbb{R}^2$.

Corollary 3 states that the optimal rendezvous point r^* of the unconstrained CRM is given by

$$\mathrm{r}^* = \frac{1}{6}\left(3\begin{bmatrix}1\\2\end{bmatrix} + 2\begin{bmatrix}3\\4\end{bmatrix} + 1\begin{bmatrix}5\\2\end{bmatrix}\right) = \begin{bmatrix}7/3\\8/3\end{bmatrix} \tag{43}$$

which is the optimal solution of CRM, as it lies within $\mathscr{R}$.

Formally, by applying (28) and (29), we derive the linear equation system $\mathbf{A}'x = -\mathbf{b}'$ as follows

$$\begin{bmatrix} 156 & 204 & 132 & 84 & 1 & -1 & 0 & 0 & 0 \\ 204 & 300 & 228 & 132 & 1 & 0 & -1 & 0 & 0 \\ 132 & 228 & 188 & 108 & 1 & 0 & 0 & -1 & 0 \\ 84 & 132 & 108 & 60 & 1 & 0 & 0 & 0 & -1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \pi \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix} = \begin{bmatrix} 152 \\ 208 \\ 144 \\ 88 \\ 1 \end{bmatrix}. \tag{44}$$

A solution of (44) is $\lambda_1 = 2/3$, $\lambda_2 = 1/6$, $\lambda_4 = 1/6$ and $\lambda_3 = \pi = \mu_1 = \mu_2 = \mu_3 = \mu_4 = 0$, which is a stationary point of CRM since (39) and (40) are satisfied. (41) implies the optimal rendezvous point of CRM is

$$r^* = \frac{2}{3}\begin{bmatrix}2\\3\end{bmatrix} + \frac{1}{6}\begin{bmatrix}4\\3\end{bmatrix} + \frac{1}{6}\begin{bmatrix}2\\1\end{bmatrix} = \begin{bmatrix}7/3\\8/3\end{bmatrix} \tag{45}$$

which is equal to r^* as expected. Note that CRM-OPT is capable of generating a global minimum of CRM even if the unconstrained solution $\mathrm{r}^* \notin \mathscr{R}$.

2) Analytical Solution of DRM$_i$*:* A similar approach to the one discussed in Sec. IV-B1 can be used to design the DRM$_i$-OPT algorithm for an agent i. This involves utilizing the local neighborhood $j \in \mathscr{N}_i$ and the extreme points $e_k^{(i)}$ for $k \in \mathscr{M}^{(i)}$ that define the local environment $\mathscr{R}^{(i)}$. DRM$_i$-OPT computes a global optimum solution $\lambda^* = [\lambda_1^*, ..., \lambda_{M^{(i)}}^*]^T$ for DRM$_i$ in $\mathscr{O}\left((M^{(i)})^4\right)$ polynomial time in the worst case, where $M^{(i)}$ is the number of extreme points of $\mathscr{R}^{(i)}$. The optimal rendezvous point r_i^* for an agent i can then be expressed as

$$r_i^* = \sum_{k \in \mathscr{M}^{(i)}} \lambda_k^* e_k^{(i)}. \tag{46}$$

Rendezvous consensus over a time-varying communication topology can be achieved using the following update rule:

$$r_i^* \leftarrow r_i^* + \frac{1}{t+1} \sum_{j \in \mathscr{N}_i} \frac{w_j}{\sum\limits_{j \in \mathscr{N}_i} w_j} (r_j^* - r_i^*) \tag{47}$$

where w_j represents the instantaneous power consumption of a neighboring agent $j \in \mathscr{N}_i$, which depends on its current velocity v_j.

C. Heuristic Solutions

1) A Heuristic Solution of CRM-D*:* As noted in Sec. IV-A CRM-D is not a CP, and therefore, the KKT first-order conditions are not sufficient to conclude the optimality of a stationary point, as stated in Corollary 2. Let r^* denote the global optimal solution of CRM, as defined in (41). If $r^* \in \mathscr{D}_l$ for $l \in \{1, ..., D\}$, then r^* can be expressed as a convex combination of the extreme points $\bar{e}_{lk}$ for $k \in \{1, ..., D_l\}$, i.e., the set $\mathscr{C}_l \neq \emptyset$, as described in (48).

$$\mathscr{C}_l = \left\{ \begin{array}{l} \bar{\lambda}_{lk} \in \mathbb{R} \mid \sum_{k=1}^{D_l} \bar{\lambda}_{lk} \bar{e}_{lk} = r^* \\ \sum_{k=1}^{D_l} \bar{\lambda}_{lk} = 1 \\ \bar{\lambda}_{lk} \geq 0,\ k = 1, \dots, D_l \end{array} \right\} \tag{48}$$

Constraint programming can verify whether $\mathscr{C}_l \neq \emptyset$ by applying the first phase of the Two-Phase method, an initialization technique, in $\mathscr{O}(n)$ polynomial time [40]. Additionally, we define the function $\texttt{DeniedIndex}(r^*)$ to return the index l if $r^* \in \mathscr{D}_l$, or -1 if $r^* \in \mathscr{R} \setminus \mathscr{D}$, with a runtime of $\mathscr{O}(nD)$.

The H-CRM-D algorithm starts with the optimal solution r^* of CRM and determines its denied region index l^* in Step 2. To move the point r^* out of the denied region l^*, an updated position $r^* + d\rho$ is computed in Step 5, where $d \in \mathbb{R}^n$ is a properly chosen offset direction and $\rho \geq 0$ is the step length. If the new point r^*, a local optimum of CRM-D, lies outside all denied regions, the offset process terminates at Step 6.

Proposition 3. *Let $r^* \in \mathscr{D}_l$, and let $\bar{e}_{lk}$ for $k \in \{1, \dots, D_l\}$ represent the extreme points of $\mathscr{D}_l$. Define the step length as*

$$\rho = \max_{k \in \{1, \dots, D_l\}} \{ \| \bar{e}_{lk} - r^* \| \}. \tag{49}$$

For any unit direction $d \in \mathbb{R}^n$, the point $(r^* + \rho d) \notin \mathscr{D}_l$.

Proof. The step length ρ in (49) represents the maximum Euclidean distance from r^* to the extreme points of $\mathscr{D}_l$. Consequently, moving ρ steps in any unit direction ensures that the point r^* is shifted beyond the boundaries of $\mathscr{D}_l$.

Proposition 4. *Let $r^* \in \mathscr{D}_l$ and the step length ρ be defined as in (49). Then, the minimum energy-consuming offset direction $\bar{d} \in \mathbb{R}^n$ is given by*

$$\bar{d}_\tau = \frac{\sum_{i \in \mathscr{N}} \left(2\rho \frac{w_i}{\|v_i\|} p_{i\tau} - 2\rho \frac{w_i}{\|v_i\|} r^*_\tau \right) - \eta}{\sum_{i \in \mathscr{N}} 2\rho^2 \frac{w_i}{\|v_i\|}} \qquad \tau = 1, \dots, n \tag{50}$$

where $\eta \in \mathbb{R} \setminus \{0\}$ is a sufficiently small constant. The corresponding unit offset direction is

$$d = \bar{d} / \|\bar{d}\|. \tag{51}$$

Proof. Let $\bar{d}$ be an offset direction, and let the step length ρ be defined as in (49). Consider the new point $r^* + \rho \bar{d}$. The energy consumption function in (17) can be rewritten as a function of $\bar{d}$ as

$$f(\bar{d}) = \sum_{i \in \mathscr{N}} \frac{w_i}{\|v_i\|} \sum_{\tau=1}^{n} \left(p_{i\tau} - r^*_\tau - \rho \bar{d}_\tau \right)^2 \tag{52}$$

since p_i, r^* and ρ are given. A non-zero offset direction $\bar{d}$ can be found by solving the following CP formulation:

$$\min \quad f(\bar{d}) \tag{53}$$

$$\text{s.t.} \quad \sum_{\tau=1}^{n} \bar{d}_\tau = c \quad : \eta \tag{54}$$

where $c \in \mathbb{R}$ is a constant and $\eta \in \mathbb{R}$ is the dual variable of (54). By applying KKT first-order conditions, $\bar{d}_\tau$ is obtained as given in (50) for $\tau = 1, \dots, n$. To find a minimum energy-consuming direction, a sufficiently small non-zero value for η, such as $\eta = 0.01$, can be selected. The corresponding unit-length direction d is then computed using (51). ■

Algorithm 2 A Heuristic Method for CRM-D (H-CRM-D)

```
Input: An instance of CRM-D
Output: A local minimum of CRM-D
1: Apply the CRM-OPT algorithm to find the optimum solution $r^*$ of CRM.
   Find the denied region of $r^*$
2: Let $l^* =$ DeniedIndex($r^*$)
3: while $l^* > 0$ do
4:     Find the step length $\rho$ in (49) and the offset direction $d$ in (51).
5:     Update $r^* \leftarrow r^* + \rho d$
6:     $l^* =$ DeniedIndex($r^*$)
7: end while
8: return $r^*$
```

H-CRM-D executes in $\mathscr{O}(nD)$ polynomial time if the denied regions are sufficiently separated. In such scenarios, r^* can be pushed out of the denied region $\mathscr{D}$ within a finite number of updates in Step 5.

Example 2. A CRM-D instance is illustrated in Fig. 2, which represents the same scenario as in Fig. 1, but with an additional denied region inside. The extreme points of the denied region are provided as follows

$$\bar{e}_1 = \begin{bmatrix} 2.2 \\ 2.7 \end{bmatrix}, \quad \bar{e}_2 = \begin{bmatrix} 2.5 \\ 2.8 \end{bmatrix}, \quad \bar{e}_3 = \begin{bmatrix} 2.8 \\ 2.7 \end{bmatrix}, \quad \bar{e}_4 = \begin{bmatrix} 2.4 \\ 2.5 \end{bmatrix}. \tag{55}$$

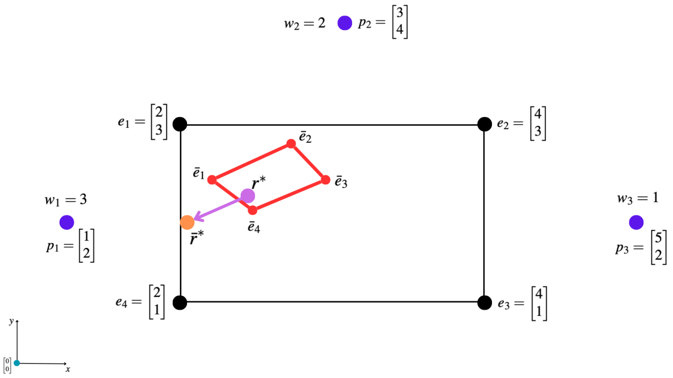
$w_2 = 2$ $p_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$
$e_1 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$
$e_2 = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$
$\bar{e}_2$
$\bar{e}_1$
r^*
$\bar{e}_3$
$w_1 = 3$
$w_3 = 1$
$\bar{e}_4$
$\bar{r}^*$
$p_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$
$p_3 = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$
$e_4 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$
$e_3 = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$
y
$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
x

Fig. 2. A CRM-D instance in $\mathbb{R}^2$.

Note that the CRM rendezvous point $r^* = \begin{bmatrix} 2.3 \\ 2.7 \end{bmatrix}$ is located within the denied region in Fig. 2. H-CRM-D calculates the step length ρ as

$$\rho = \max_{k \in \{1, \dots, 4\}} \{ \| \bar{e}_k - r^* \| \} = 0.47. \tag{56}$$

An offset direction $\bar{d}$ is determined as

$$\bar{d} = \begin{bmatrix} \bar{d}_1 \\ \bar{d}_2 \end{bmatrix} = \begin{bmatrix} -\frac{\eta}{12\rho^2} \\ -\frac{\eta}{12\rho^2} \end{bmatrix} \quad \text{for } \eta \in \mathbb{R} \setminus \{0\}, \tag{57}$$

which gives the corresponding unit direction d as

$$d = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}. \tag{58}$$

H-CRM-D updates the rendezvous point $\bar{r}^*$ as

$$\bar{r}^* = \begin{bmatrix} 2.3 \\ 2.7 \end{bmatrix} + 0.47 \begin{bmatrix} -0.71 \\ -0.71 \end{bmatrix} = \begin{bmatrix} 2 \\ 2.3 \end{bmatrix} \tag{59}$$

which lies outside the denied region as shown in Fig. 2.

2) A Heuristic Solution of $\mathtt{DRM}_i$*-D:* We can follow a similar approach as in Sec. IV-C1 to obtain a local optimal solution of $\mathtt{DRM}_i$-D by utilizing the local neighborhood $j \in \mathcal{N}_i$ and the local environment $\mathscr{R}^{(i)}$ and $\mathscr{D}^{(i)}$.

Specifically, $\mathtt{DeniedIndex}(r_i^*)$ returns the index l if $r_i^* \in \mathscr{D}_l^{(i)}$, or -1 if $r_i^* \in \mathscr{R}^{(i)} \setminus \mathscr{D}^{(i)}$ for an agent i. The step length $\rho^{(i)}$ is defined by rewriting (49) using the local extreme points $\bar{e}_{lk}^{(i)}$ for $l \in \{1,...,D\}$ and $k \in \{1,...,D_l^{(i)}\}$. Then, the unit offset direction $d^{(i)}$ is obtained by rewriting (51) with $\rho^{(i)}$ and $j \in \mathcal{N}_i$.

The H-$\mathtt{DRM}_i$-D algorithm begins with the global optimal solution r_i^* of $\mathtt{DRM}_i$, as given in (46). Agents then achieve consensus on the rendezvous point r^* by applying the update rule in (47). The algorithm iteratively updates $r^* \leftarrow r^* + \rho d$ until the condition $r^* \in \mathscr{R}^{(i)} \setminus \mathscr{D}^{(i)}$ holds for all $i \in \mathcal{N}$. This procedure completes in $\mathscr{O}(nD)$ time.

The H-$\mathtt{DRM}_i$-D algorithm is lightweight, enabling easy implementation on microprocessors. Moreover, an agent can apply H-$\mathtt{DRM}_i$-D to rendezvous with the MAS at any time by utilizing the most up-to-date local information.

D. Distributed Aggregation Control

In the double integrator model, the state of agent i is given by (p_i, v_i), and the MAS centroid is defined as $\bar{p} = \frac{1}{N}\sum_{i\in\mathcal{N}} p_i$. Let r^* denote the rendezvous point generated by the H-CRM-D or H-$\mathtt{DRM}_i$-D algorithms.

The distributed aggregation control of a MAS around a rendezvous point r^* is governed by multiple potential functions. Specifically, the attraction potential (60) and the repulsion potential (61) regulate the interactions among agents, while the tracking potential (62) directs the agents toward r^*. The control parameters α, β, γ, $\delta \in \mathbb{R}_+$ are employed, with the condition that $\alpha < \beta$.

$$J^a(p) = \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}_i} \frac{\alpha}{2} \|p_i - p_j\|^2 \tag{60}$$

$$J^r(p) = \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}_i} \frac{\beta\gamma}{2} \exp\left(-\frac{\|p_i - p_j\|^2}{\gamma}\right) \tag{61}$$

$$J^t(p) = \frac{N\delta}{2} \|\bar{p} - r^*\|^2 \tag{62}$$

A multimodal Gaussian profile models the obstacles within $\mathscr{R}$, while the environment profile (63) captures the interactions between the agents and the obstacles. This profile is parameterized by the amplitude $\zeta_l \in \mathbb{R}$, the centroid $\theta_l \in \mathbb{R}^n$, and the deviation $\sigma_l \in \mathbb{R}_+$ for $l \in \{1,...,D\}$.

$$J^o(p) = \sum_{i\in\mathcal{N}} \sum_{l=1}^{D} \frac{\zeta_l}{2} \exp\left(-\frac{\|\theta_l - p_i\|^2}{\sigma_l}\right) \tag{63}$$

The energy potential (64) is the total instantaneous energy consumption, where w_i is a continuously differentiable convex function representing the power consumption of agent i.

$$J^e(v) = \sum_{i\in\mathcal{N}} w_i(v_i) \tag{64}$$

The candidate Lyapunov function (65) represents the total aggregation potential, weighted by W_a, W_t, W_o and W_e, corresponding to attraction–repulsion, tracking, obstacle interaction, and energy consumption, respectively.

$$J(p,v) = W_a\left(J^a(p) + J^r(p)\right) + W_t J^t(p) + W_o J^o(p) + W_e J^e(v) \tag{65}$$

The potential $J(p,v)$ guarantees the convergence of $(p(t), v(t))$ to an equilibrium state within $\Omega_e = \{(p,v) \mid v_i = 0 \text{ and } \nabla_i J(p,v) = 0 \;\; \forall i \in \mathcal{N}\}$ as $t \to \infty$, since $\dot{J}(p,v)$ is negative semi-definite. As a result, the acceleration control command for agent i is given as

$$\begin{aligned} u_i &= -\nabla_{p_i} J(p,v) - \nabla_{v_i} J(p,v) \\ &= -\sum_{j\in\mathcal{N}_i} W_a \left[\alpha - \beta \, \exp\left(-\frac{\|p_i - p_j\|^2}{\gamma}\right)\right](p_i - p_j) \\ &\quad - W_t\, \delta(\bar{p} - r^*) - W_e\, \dot{w}_i(v_i) \\ &\quad - \sum_{l=1}^{D} W_o \frac{\zeta_l}{\sigma_l} \exp\left(-\frac{\|\theta_l - p_i\|^2}{\sigma_l}\right)(\theta_l - p_i) \end{aligned} \tag{66}$$

The acceleration update equation (67) ensures that a_i remains within the specified bounds $(\underline{a}_i, \bar{a}_i)$. The velocity v_i is smoothed using the parameter $\psi \in [0,1]$ in (68) to improve energy efficiency. Finally, the new position p_i is computed on the control frequency ν_2 Hz in (69).

$$a_i = \langle u_i \rangle_{\underline{a}_i}^{\bar{a}_i} = \max\{\underline{a}_i, \min\{\|u_i\|, \bar{a}_i\}\} \frac{u_i}{\|u_i\|} \tag{67}$$

$$v_i \leftarrow \left\langle \psi(v_i + a_i/\nu_2) + (1-\psi)\, v_i \right\rangle_{\underline{v}_i}^{\bar{v}_i} \tag{68}$$

$$p_i \leftarrow p_i + v_i/\nu_2 \tag{69}$$

V. Rendezvous Controllers

A. Centralized MAS Rendezvous Controller

The *centralized MAS rendezvous controller* (C-MAS) operates using two control loops: a low-frequency *outer* loop and a high-frequency *inner* loop. In the outer loop, a central decision maker gathers the current states of the agents, (p_i, v_i, w_i), and solves CRM-D using the H-CRM-D algorithm. The decision maker updates the rendezvous point r^* and broadcasts it to the MAS at a frequency of ν_1 Hz. In the inner loop, each agent i receives the centrally determined rendezvous point r^* and computes its next state (p_i, v_i, w_i) by executing distributed aggregation control commands, as given in (66), at a frequency of ν_2 Hz.

In the *synchronous* C-MAS controller, we assume that $\nu_2 \geq \nu_1$. The outer loop is responsible for adjusting the rendezvous point r^* in response to dynamic changes within the MAS, such as the addition or removal of agents, redefinition of $\mathscr{R}$, or modifications to $\mathscr{D}$.

Algorithm 3 Centralized MAS Rendezvous Controller (C-MAS)

```
Input: An instance of CRM-D
Output: Agent trajectories to the rendezvous point
    Outer Control of the Central Decision Maker with $\nu_1$ Hz
    Get the current agent states $(p_i, v_i, w_i)$ for $i \in \mathcal{N}$.
    Apply the H-CRM-D algorithm to find the rendezvous point $r^*$ and publish to the Inner Control.
    Inner Control of an Agent $i$ for $i \in \mathcal{N}$ with $\nu_2$ Hz
    Get the central rendezvous point $r^*$
        and the current neighbor states $(p_j, v_j, w_j)$ for $j \in \mathcal{N}_i$.
    Apply aggregation control (66) towards $r^*$.
    Publish the current state $(p_i, v_i, w_i)$ to the Outer Control.
```

B. Distributed MAS Rendezvous Controller

The *distributed MAS rendezvous controller* (D-MAS) operates by controlling each agent i at a frequency of ν_2 Hz for $i \in \mathcal{N}$. An agent i receives the current state (p_j, v_j, w_j) and the local region data from its neighboring agents $j \in \mathcal{N}_i$. Agent i computes the rendezvous point r_i^* using the H-DRM$_i$-D algorithm, which is based on its local regions $\mathscr{R}^{(i)}$ and $\mathscr{D}^{(i)}$, and operates at a frequency of ν_1 Hz. Subsequently, agent i moves towards r_i^* by applying the distributed aggregation control commands (66). The D-MAS controller iterates by continuously publishing the current state of agent i to its neighbors.

Algorithm 4 Distributed MAS Rendezvous Controller (D-MAS)

```
Input: An instance of DRM$_i$-D, an agent $i$ for $i \in \mathcal{N}$
Output: Trajectory of agent $i$ to the rendezvous point
    Control of Agent $i$ with $\nu_2$ Hz
    Get the current neighbor states $(p_j, v_j, w_j)$ and
        the current local regions $\mathscr{R}^{(j)}$ and $\mathscr{D}^{(j)}$ for $j \in \mathcal{N}_i$.
    Combine the neighbor regions to obtain $\mathscr{R}^{(i)}$ and $\mathscr{D}^{(i)}$.
        Update of $r_i^*$ with $\nu_1$ Hz
        Apply the H-DRM$_i$-D algorithm to find the rendezvous point $r_i^*$.
    Apply aggregation control (66) towards $r_i^*$.
    Publish the current state $(p_i, v_i, w_i)$ and the local regions $\mathscr{R}^{(i)}$ and $\mathscr{D}^{(i)}$ to the neighbor agents.
```

We assume that $\nu_2 \geq \nu_1$ in the *asynchronous* D-MAS controller, meaning that the frequency at which an agent i receives neighbor data is higher than the frequency at which the rendezvous point is updated.

VI. Simulation Results

A. Instance Generation

Simulations are performed in MATLAB Online within a $(10 \text{ m} \times 10 \text{ m} \times 10 \text{ m})$ flight arena, as illustrated in Fig. 3. We randomly initialize $N = 10$ agents around a region $\mathscr{R}$, which is defined by $M = 8$ extreme points and overlaps with a denied region $(D = 1)$. Initially, the weighted centroid of the agent positions, computed using Equation (42), lies outside the region $\mathscr{R}$ and within the denied area $\mathscr{D}$.

TABLE I
Simulation Parameters

General Parameters		Aggregation Parameters	
(N, K, M, D)	(10, 2, 8, 1)	$(\alpha, \beta, \gamma, \delta)$	(2.5, 50, 0.2, 20)
T	200 secs	ζ, ψ	100, 0.8
w_i	8.8 watts	(W_a, W_t, W_o, W_e)	(0.4, 0.6, 1, 0.1)
ω_i	(8.5, 9.5) meter	(ν_1, ν_2)	(0.01, 10) Hz

The heterogeneous MAS consists of $K = 2$ agent types in equal numbers, both governed by double integrator dynamics. Agent specifications are based on the CrazyFlie 2.1 quadcopter, which is powered by a 250 mAh 3.7V LiPo battery, offering approximately 3–4 minutes of flight time [41]. Velocity bounds are set as $(\underline{v}_i, \bar{v}_i) = (0, 1)$ meter/sec for all agents. Acceleration limits are defined as $(\underline{a}_i, \bar{a}_i) = (-0.5, 0.5)$ meter/sec^2 for type-1 agents and $(\underline{a}_i, \bar{a}_i) = (-1, 1)$ meter/sec^2 for type-2 agents. In D-MAS simulations, the communication range is set to $\omega_i = 8.5$ meters for type-1 agents and $\omega_i = 9.5$ meters for type-2 agents. Each agent i can directly receive information from its neighbors $j \in \mathcal{N}_i$. Experimental studies report that CrazyFlie 2.1 maintains a nearly constant power consumption of $w_i = 8.8$ watts across varying flight speeds [42].

Let r_s^* represent the rendezvous point on which the agents reach consensus during simulation round s. Rendezvous is considered achieved when all agents remain within an $\varepsilon = 0.5$ neighborhood of this point for five consecutive control steps, each lasting $1/\nu_2$ secs. The two-round simulation, with a total duration of $T = 2/\nu_1$ secs, proceeds through the following sequence of events:

1) *Initialization* $(t = 0)$: The controller computes a rendezvous point r_1^*.
2) *Round 1* $\left(t \in \left[0, \frac{1}{\nu_1}\right)\right)$: Agents move to aggregate within an $\varepsilon-$ball around r_1^*.
3) *Agent Addition* $\left(t = \frac{1}{\nu_1}\right)$: A new agent is introduced at a random position, and existing agents are randomly perturbed. The controller recalculates the rendezvous point r_2^*.
4) *Round 2* $\left(t \in \left[\frac{1}{\nu_1}, \frac{2}{\nu_1}\right]\right)$: Agents aggregate around the new rendezvous point r_2^*.

The general and controller parameters are summarized in TABLE I. For the denied region, the parameter θ is defined as the centroid of its extreme points, and σ is the maximum distance from this centroid to any of the extreme points. The simulation code is available in our GitHub repository [43].

B. Simulation Results

Corollary 3 states that the weighted centroid of agent positions is the optimal solution to the unconstrained rendezvous problem. However, since this centroid lies outside the feasible region $\mathscr{R} \setminus \mathscr{D}$, as shown in Fig. 3, the H-CRM-D and H-DRM$_i$-D algorithms are employed by the C-MAS and D-MAS controllers to compute valid rendezvous points.

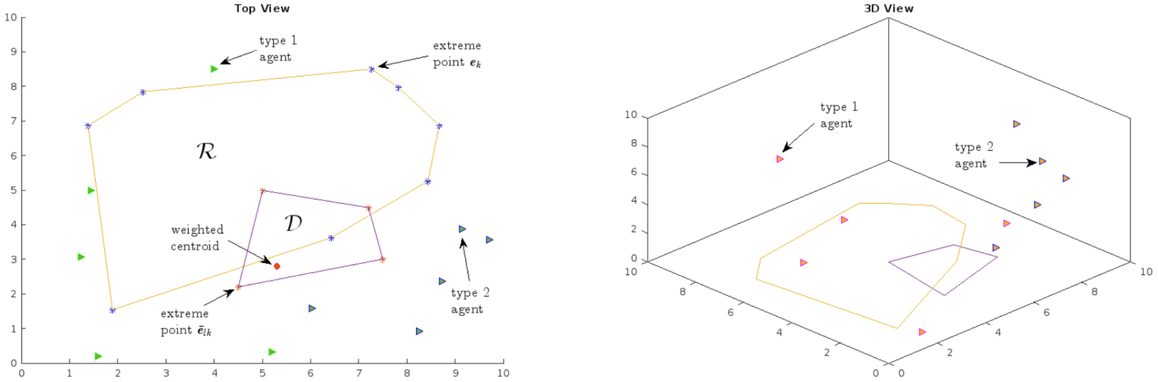

Fig. 3. Top and 3D views of a MAS rendezvous scenario with $N = 10$ agents, $K = 2$ agent types, $M = 8$ vertices, and $D = 1$ denied region.

Fig. 4 depicts the two-round simulation of the C-MAS controller. Agents follow trajectories shown by blue dots and reach their final positions marked by red stars, while the rendezvous points are indicated by orange stars. In Fig. 4, the first column presents the 3D trajectories, the second column shows the top view, and the third column provides a zoomed top view around the rendezvous point.

In the first round of the C-MAS simulation, CRM-OPT computes a rendezvous point within $\mathscr{R}$ that falls inside the denied region $\mathscr{D}$. Consequently, H-CRM-D corrects this by projecting the point into the feasible region $\mathscr{R} \setminus \mathscr{D}$, yielding r_1^*. In the second round, a new agent joins the system, following the orange trajectory, and all agents converge around the updated rendezvous point r_2^*. In both rounds, the agents maneuver around the denied region en route to the rendezvous point. The convergence of agent positions over time is illustrated in Fig. 5. The corresponding simulation video is available at https://youtu.be/4FA6B6IgK_E.

In the D-MAS simulation, all agents are assumed to have complete knowledge of the region $\mathscr{R}$ and the denied region $\mathscr{D}$, i.e., $\mathscr{R}^{(i)} = \mathscr{R}$ and $\mathscr{D}^{(i)} = \mathscr{D}$ for all $i \in \mathscr{N}$. However, each agent i has only local information about the positions of its neighboring agents, limited by its communication range ω_i.

At each round of the D-MAS simulation, the H-DRM$_i$-D algorithm initially computes a local rendezvous point $r_i^* \in \mathscr{R} \setminus \mathscr{D}$ for each agent i. It then applies the update rule in (47) to iteratively reach consensus across the network, resulting in a shared rendezvous point r_s^* at round s. As expected, the rendezvous points computed by the C-MAS and D-MAS controllers are found to be closely aligned across both simulation rounds, as summarized in TABLE II. The convergence of agent positions in the D-MAS simulation is illustrated in Fig.6. Notably, the agents follow trajectories similar to those in the C-MAS simulation (Fig. 4). The corresponding simulation video is available at https://youtu.be/OA8x--Fmo_U.

The comparison between the C-MAS and D-MAS controllers highlights key differences in consensus formation, robustness, and operational efficiency. The C-MAS controller achieves consensus via a centrally computed rendezvous point, ensuring coordinated behavior but introducing vulnerabilities related to central decision-maker failure and limited communication ranges. In contrast, the D-MAS controller employs a distributed consensus rule, enabling agents to independently compute and iteratively align on a common rendezvous point that closely matches the one derived by C-MAS. This decentralized mechanism enhances robustness to agent failures and dynamic changes within the MAS, while also reducing dependence on global communication and centralized control. Furthermore, D-MAS demonstrates greater scalability and adaptability, operating as an asymmetric *meet anytime* algorithm that supports flexible integration of agents into the system at any time.

TABLE II
COMPARISON OF RENDEZVOUS POINTS COMPUTED BY C-MAS AND D-MAS CONTROLLERS

Point	**Controller**	x	y	z
r_1^*	C-MAS D-MAS	6.9482 7.3254	4.9523 5.0849	4.5505 4.5505
r_2^*	C-MAS D-MAS	8.0442 8.3231	6.2275 6.1254	5.4064 5.4090

VII. CONCLUSION

This research addresses the energy-efficient control of a heterogeneous MAS operating within a convex polyhedral region that may contain denied zones. To achieve this, we developed both centralized and distributed mathematical formulations to identify a feasible, energy-efficient rendezvous point within the region of interest. Exact and heuristic solution methods were proposed to solve these formulations. Subsequently, the centralized C-MAS and distributed D-MAS controllers were designed to guide the MAS to the designated rendezvous point, effectively managing collision avoidance and obstacle navigation.

Energy efficiency was pursued through three key strategies: (1) optimizing the rendezvous point based on agent power consumption, (2) computing aggregation control to reduce overall power usage, and (3) smoothing agent trajectories to minimize abrupt velocity and acceleration changes. Simulation results demonstrate that both controllers effectively direct the MAS to the desired rendezvous point, with the D-MAS controller offering scalability, robustness, and a lightweight implementation as a practical asymmetric distributed controller.

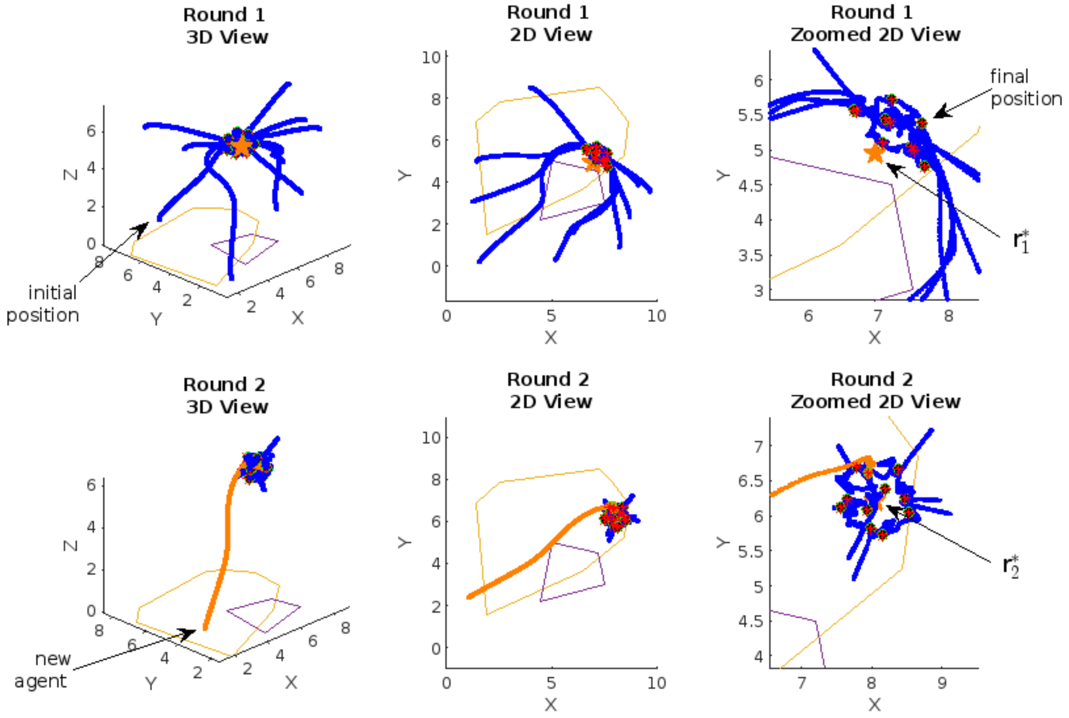

Fig. 4. Agent trajectories in the C-MAS simulation rounds. The corresponding simulation video is available here. The D-MAS controller produces nearly identical rendezvous points. Its simulation video is available here.

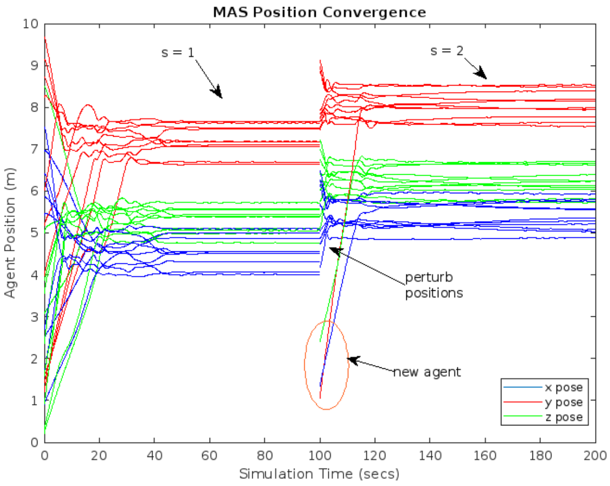

Fig. 5. Convergence of agent positions under the C-MAS controller.

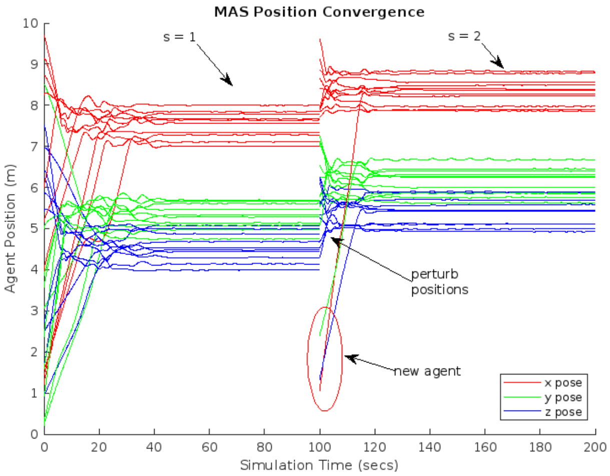

Fig. 6. Convergence of agent positions under the D-MAS controller.

Future work could explore optimizing rendezvous point selection based on agent-specific preferences, considering openness and asymmetry alongside energy efficiency, rather than assuming all non-denied positions are equally acceptable.

Acknowledgment

The author cordially would like to thank Şeref Naci Engin and Veysel Gazi for their insightful comments on this research.

References

[1] P. Shi and B. Yan, "A survey on intelligent control for multiagent systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 161–175, 2021.

[2] U. Munz, A. Papachristodoulou, and F. Allgower, "Delay-dependent rendezvous and flocking of large scale multi-agent systems with communication delays," in *2008 47th IEEE Conference on Decision and Control*, 2008, pp. 2038–2043.

[3] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[4] F. Xiao, L. Wang, J. Chen, and Y. Gao, "Finite-time formation control for multi-agent systems," *Automatica*, vol. 45, no. 11, pp. 2605–2611, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109809003586

[5] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[6] J. Hu and G. Feng, "Distributed tracking control of leader–follower multi-agent systems under noisy measurement," *Automatica*, vol. 46, no. 8, pp. 1382–1387, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109810002268

[7] D. Chang, S. Shadden, J. Marsden, and R. Olfati-Saber, "Collision avoidance for multiple agent systems," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 1, 2003, pp. 539–543 Vol.1.

[8] J. Yan, X. P. Guan, and F. X. Tan, "Target tracking and obstacle avoidance for multi-agent systems," *Int. J. Autom. Comput.*, vol. 7, pp. 550–556, 2010.

[9] P. Alliez, F. Bonardi, S. Bouchafa, J.-Y. Didier, H. Hadj-Abdelkader, F. I. Muñoz, V. Kachurka, B. Rault, M. Robin, and D. Roussel, "Real-time multi-slam system for agent localization and 3d mapping in dynamic scenarios," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4894–4900.

[10] J. Pavón, J. Gómez-Sanz, A. Fernández-Caballero, and J. J. Valencia-Jiménez, "Development of intelligent multisensor surveillance systems with agents," *Robotics and Autonomous Systems*, vol. 55, no. 12, pp. 892–903, 2007, robotics and Autonomous Systems in the 50th Anniversary of Artificial Intelligence. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889007001182

[11] D. S. Drew, "Multi-agent systems for search and rescue applications," *Curr Robot Rep*, vol. 2, pp. 189–200, 2021.

[12] J. Song, K. Zhao, and Y. Liu, "Survey on mission planning of multiple unmanned aerial vehicles," *Aerospace*, vol. 10, no. 3, 2023. [Online]. Available: https://www.mdpi.com/2226-4310/10/3/208

[13] J. Xie and C.-C. Liu, "Multi-agent systems and their applications," *Journal of International Council on Electrical Engineering*, vol. 7, no. 1, pp. 188–197, 2017.

[14] J. Lin, A. Morse, and B. Anderson, "The multi-agent rendezvous problem - the asynchronous case," in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, vol. 2, 2004, pp. 1926–1931 Vol.2.

[15] C. H. Caicedo-Nunez and M. Zefran, "Consensus-based rendezvous," in *2008 IEEE International Conference on Control Applications*, 2008, pp. 1031–1036.

[16] A. Pelc, "Deterministic rendezvous in networks: A comprehensive survey," *Networks*, vol. 59, no. 3, pp. 331–347, 2012. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/net.21453

[17] M. Woodside, J. Neilson, D. Petriu, and S. Majumdar, "The stochastic rendezvous network model for performance of synchronous client-server-like distributed software," *IEEE Transactions on Computers*, vol. 44, no. 1, pp. 20–34, 1995.

[18] J. Lin, A. Morse, and B. Anderson, "The multi-agent rendezvous problem," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 2, 2003, pp. 1508–1513 Vol.2.

[19] S. P. Li, X and Y. Wang, "Distributed cooperative adaptive tracking control for heterogeneous systems with hybrid nonlinear dynamics," *Nonlinear Dyn*, vol. 95, pp. 2131—-2141, 2019.

[20] Z. Peng, B. Li, X. Chen, and J. Wu, "Online route planning for uav based on model predictive control and particle swarm optimization algorithm," in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, 2012, pp. 397–401.

[21] P. B. Sujit and R. Beard, "Multiple uav path planning using anytime algorithms," in *2009 American Control Conference*, 2009, pp. 2978–2983.

[22] Z. Cheng, E. Wang, Y. Tang, and Y. Wang, "Real-time path planning strategy for uav based on improved particle swarm optimization." *J. Comput.*, vol. 9, no. 1, pp. 209–214, 2014.

[23] D. Wu, T. Zhang, Y. Zhong, F. Jiang, and J. Li, "Analytical shaping method for low-thrust rendezvous trajectory using cubic spline functions," *Acta Astronautica*, vol. 193, pp. 511–520, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576522000285

[24] A. Amirkhani and A. H. Barshooi, "Consensus in multi-agent systems: a review," *Artif Intell Rev*, vol. 55, p. 3897–3935, 2022.

[25] J. Song, "Observer-based consensus control for networked multi-agent systems with delays and packet-dropouts," vol. 12, pp. 1287–1302, 08 2016.

[26] X. Lu, C. Zhang, and B. Qin, "An improved vicsek model of swarm based on remote neighbors strategy," *Physica A: Statistical Mechanics and its Applications*, vol. 587, p. 126553, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437121008268

[27] Z. Yang, S. Zhu, C. Chen, G. Feng, and X. Guan, "Leader-follower formation control of nonholonomic mobile robots with bearing-only measurements," *Journal of the Franklin Institute*, vol. 357, no. 3, pp. 1628–1643, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0016003219308014

[28] L. Ding, Q.-L. Han, and G. Guo, "Network-based leader-following consensus for distributed multi-agent systems," *Automatica*, vol. 49, p. 2281–2286, 07 2013.

[29] A.-Y. Huang, Y.-Z. Luo, and H.-N. Li, "Analytical gradient-guided nonlinear programming approach for multitarget rendezvous mission," *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 3, pp. 581–589, 2023. [Online]. Available: https://doi.org/10.2514/1.G006967

[30] L. Dai, Q. Cao, Y. Xia, and Y. Gao, "Distributed mpc for formation of multi-agent systems with collision avoidance and obstacle avoidance," *Journal of the Franklin Institute*, vol. 354, no. 4, pp. 2068–2085, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0016003216304926

[31] Z. Li, G. Yu, Q. Zhang, S. Song, and H. Cui, "Adaptive sliding mode control for spacecraft rendezvous with unknown system parameters and input saturation," *IEEE Access*, vol. 9, pp. 67 724–67 733, 2021.

[32] G. Xing, M. Li, T. Wang, W. Jia, and J. Huang, "Efficient rendezvous algorithms for mobility-enabled wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 47–60, 2012.

[33] L. Xu, G. Zhang, S. Qiu, and X. Cao, "Optimal multi-impulse linear rendezvous via reinforcement learning," *Space: Science &amp; Technology*, vol. 3, p. 0047, 2023. [Online]. Available: https://spj.science.org/doi/abs/10.34133/space.0047

[34] Y. Zhang, Y. Li, Z. Wu, and J. Xu, "Deep reinforcement learning for uav swarm rendezvous behavior," *Journal of Systems Engineering and Electronics*, vol. 34, no. 2, pp. 360–373, 2023.

[35] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.

[36] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms*. John wiley & sons, 2013.

[37] Shawn. (2019) Types of distance sensors and how to select one? [Online]. Available: https://www.seeedstudio.com

[38] N. J. Higham, "Gaussian elimination," *WIREs Computational Statistics*, vol. 3, no. 3, pp. 230–238, 2011. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.164

[39] M. Best and N. Chakravarti, *Active set algorithms for isotonic regression; A unifying framework*. Springer, 1990.

[40] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear programming and network flows*. John Wiley & Sons, 2011.

[41] Crazyflie 2.1, make your ideas fly! [Online]. Available: https://www.bitcraze.io

[42] D. A. Olejnik, S. Wang, J. J. Dupeyroux, S. Stroobants, M. Karásek, C. D. Wagter, and G. C. de Croon, "An experimental study of wind resistance and power consumption in mavs with a low-speed multi-fan wind system," in *Proceedings of the 2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2989–2994. [Online]. Available: https://ieeexplore.ieee.org/document/9811834/

[43] B. Kabakulak. Rendezvous algorithms. [Online]. Available: https://github.com/banukabakulak/RendezvousAlgorithms

Banu Kabakulak received her B.Sc. degrees in Industrial Engineering and Mathematics (double major) from Boğaziçi University, İstanbul, Türkiye, in 2007, followed by M.Sc. and Ph.D. degrees in Industrial Engineering from the same institution in 2010 and 2018, respectively. Since 2019, she has been an Assistant Professor in the Department of Industrial Engineering at İstanbul Bilgi University. She is currently a visiting researcher in the School of Mathematics at the University of Edinburgh, UK. Her research interests span IoT networks, telecommunications, large-scale optimization, mathematical programming, and intelligent multi-agent systems. In 2020, she received the *Operations Engineering & Analytics Best Application Paper Award* from *IISE Transactions.*