A Distributed Pipeline for Collaborative Pursuit in the Target Guarding Problem

Yansong Chen, Yuchen Wu, Helei Yang, *Graduate Student Member, IEEE*, Junjie Cao, Qinqin Wang, and Yong Liu

***Abstract*—The target guarding problem (TGP) is a classical combat game where pursuers aim to capture evaders to protect a territory from intrusion. This paper proposes a distributed pipeline for multi-pursuer multi-evader TGP with the capability to accommodate varying numbers of evaders and criteria for successful pursuit. The pipeline integrates a cooperative encirclement-oriented distributed model predictive control (CEO-DMPC) method with a collaborative grouping strategy for trajectory planning of pursuers. This integration achieves cooperation and collision avoidance during the capture process across various scenarios. Besides, the objective function of CEO-DMPC employs sequences of predicted states instead of only a terminal state. Evaders are guided by the artificial potential field (APF) policy to reach their goals without being captured. Simulations with different parameters are conducted to validate the whole pipeline and the experiment results are illustrated and analyzed.**

***Index Terms*—Cooperating robots, multi-robot systems, task and motion planning, target guarding problem (TGP), model predictive control (MPC).**

I. Introduction

THE pursuit-evasion game (PEG) has garnered significant attention due to its wide range of applications in navigation, confrontation operations, and surveillance [1]. The most basic version of PEG involves two agents: a pursuer and an evader [2]. The pursuer attempts to capture the evader while the evader strives to escape. The target guarding problem (TGP), a specific variant of PEG, is initially studied by Isaacs [3]. TGP focuses on protecting a target asset from breach by capturing all the evaders in the shortest time [4].

Up to now, numerous studies have proposed a variety of methods for different TGP scenarios. Similar to differential games, TGP can be categorized into four types: 1) one pursuer and one evader, 2) N pursuers and one evader, 3) one pursuer and M evaders, and 4) N pursuers and M evaders [1]. The complexity of TGP increases significantly as the number of agents involved grows. Consequently, classical models are better suited for small-scale TGP. Hence, many researchers primarily focus on elementary versions of TGP where each side comprises no more than two agents [2], [5], [6], [7].

This paper proposes a distributed pipeline consisting of two components: a trajectory planning method named cooperative encirclement-oriented distributed model predictive control (CEO-DMPC) and a cooperative grouping strategy. This pipeline is designed for multi-pursuer multi-evader TGP with two phases depicted in Fig. 1. The objective of the game is for the pursuing side to intercept all the evaders before they reach their respective goals near the protected target. Failure to do so

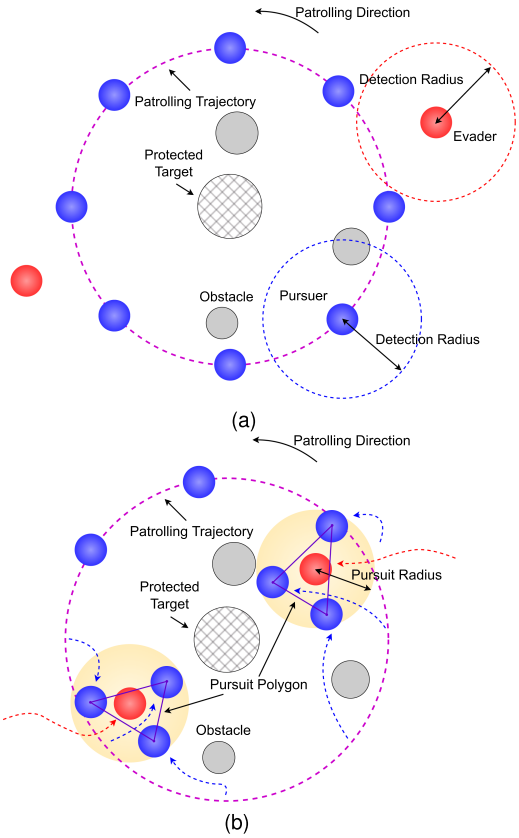

Fig. 1. Two phases of the whole game process in two-dimensional domain with a protected target area located at the origin and several static obstacles around. The game phase transitions instantaneously upon the detection of an evader. All pursuers navigate a given circular trajectory with a fixed radius r_{patrol} in the patrolling phase while some of them are selected and organized into groups in the pursuing phase. The chosen pursuers follow the trajectories from CEO-DMPC to encircle evaders. (a) The patrolling phase. (b) The pursuing phase.

Manuscript received 7 October 2023; accepted 7 December 2023. Date of publication 4 January 2024; date of current version 22 January 2024. This letter was recommended for publication by Associate Editor Harold Soh and Editor Hanna Kurniawati upon evaluation of the reviewers' comments. This work was supported by NSFC under Grant 62088101 Autonomous Intelligent Unmanned Systems. *(Corresponding authors: Junjie Cao; Yong Liu.)*

Yansong Chen, Yuchen Wu, Helei Yang, Junjie Cao, and Yong Liu are with the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China (e-mail: 22132009@zju.edu.cn; wyuchen@zju.edu.cn; helei_yang@zju.edu.cn; cjunjie@zju.edu.cn; yongliu@iipc.zju.edu.cn).

Qinqin Wang is with the Science and Technology Innovation Center, Beijing 100012, China (e-mail: qinwqin@163.com).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3349977, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3349977

results in the pursuing side losing the game. The protected area is considered as a unique obstacle, as it is often inaccessible in real-world scenarios, such as an island defended by ships. Moreover, the initial conditions of the game, including the numbers of evaders and the criteria for successful pursuit, are variable and predefined prior to the start of the game. Consequently, CEO-DMPC has to be adaptable to different game conditions while ensuring collision avoidance. To achieve high adaptability, sequences of predicted states are employed in the objective function of CEO-DMPC rather than a single target state. Simulations are performed in realistic scenarios resembling navy or air force expulsion missions and the results are visualized and analyzed, demonstrating the practical value of the pipeline. Specifically, the pipeline is tested using quad-rotor unmanned aerial vehicles (UAVs) at a fixed altitude in the Gazebo environment.

The main contributions of this paper are summarized as follows:

1) A distributed pipeline that is adaptive to different numbers of evaders and criteria for successful pursuit is proposed.
2) A cooperative trajectory planning method, CEO-DMPC, is designed for TGP and demonstrates good performance under different initial conditions.
3) A cooperative grouping strategy is proposed to handle varying initial conditions by dividing the grouping problem into the ordinary grouping subproblem and the position allocation subproblem.
4) Simulations and analyses are conducted in the numerical environment and Gazebo with quad-rotor UAVs in the elaborately designed scenarios.

The rest of the paper is organized as follows: Section II introduces some related works as well as their advantages and disadvantages. The problem is stated in Section III. Section IV presents cooperative grouping strategies for pursuers and Section V formulates CEO-DMPC. Experiments are shown in Section VI and conclusions are drawn in Section VII.

II. Related Work

Numerical researchers have studied TGP with a variety of methods, such as geometric methods, optimization-based methods, deep reinforcement learning (DRL), etc. The rest of this section will introduce these categories of methods.

A. Geometric Methods

Geometric analysis can derive dominance range for the game of kind and produce the optimal strategy in the close form for the game of degree [5]. Shishika et al. [8] use Apollonius circle to give the dominance range in a one-pursuer one-evader scenario. Oyler et al. [9] employ bundles of isochrones to identify singular surfaces as boundaries of dominance range in the presence of obstacles. Mohanan et al. [5] modify a strategy named a command to optimal interception point (COIP). TGP with more complex elements, such as high-dimensional space [10], noisy environment [11], and convex target set [12], can also be solved by geometric methods. Additionally, some works prove the optimality of their methods by showing that they conform to the saddle point of the game with Hamilton-Jacobi-Isaacs (HJI) equation [10], [12]. However, geometric methods cannot handle a large quantity of agents that is hard to be modeled as a differential game. Consequently, many factors in real applications, such as cooperation, collision avoidance, task assignment, and changing numbers of evaders, are not fully considered.

B. Optimization-Based Methods

TGP can also be modeled as an optimization problem. Some studies combine optimization theories with other approaches, such as barrier lines [6], Hamilton-Jacobi reachability [13], and Nash equilibrium [14] to achieve safety and robustness. Among optimization theories, model predictive control (MPC) is a widely used method that needs to solve a constrained optimization problem. But the interaction between both sides poses a challenge in designing a concise objective function describing the mission goal. To solve the dilemma, state estimation techniques, such as inverse optimal control (IOC) [15], Bayesian fisher information matrix (FIM) [16], and extended Kalman filter (EKF) [2], are combined with MPC to decouple the two sides. Besides, some articles focus on specific phases of TGP, such as preserving a state where an evader is already surrounded by pursuers of different quantities [17], [18]. But optimization-based methods cannot deal with intricate real-world scenarios and do not scale well with the increasing number of agents. Consequently, they are primarily restricted to simple small-scale applications [19].

C. Deep Reinforcement Learning

Unlike traditional methods, DRL offers adaptability to diverse scenarios through the design of suitable reward functions and frameworks. Many well-established DRL algorithms, such as deep Q-network (DQN) [20], deep deterministic policy gradient (DDPG) [21], proximal policy optimization (PPO) [22], and soft actor-critic (SAC) [23] are employed to deal with conflict scenarios like TGP. For multi-pursuer multi-evader scenarios, certain works design hierarchical DRL frameworks [24], [25] instead of end-to-end schemes. Besides, other studies focus on the task assignment process, which determines the target evaders for pursuers [26]. Nevertheless, most of the works are limited to small-scale scenarios without obstacles partially due to the high computational expense of training and the instability of models in complex environments. Additionally, the numbers of agents are generally fixed because DRL needs unchanged state and action dimensions.

III. Problem Statement

A. Dynamic Model and Constraints

We firstly introduce two sets, $\mathcal{N}_p := \{0, 1, \ldots, n_p - 1\}$ and $\mathcal{N}_e := \{n_p, n_p + 1, \ldots, n_p + n_e - 1\}$, to represent n_p pursuers and n_e evaders, respectively. Then, all the agents can be expressed as $\mathcal{N} := \mathcal{N}_p \cup \mathcal{N}_e$. With a given discretization time period h, agent $i \in \mathcal{N}$ is modeled as second order integral dynamics satisfying

$$\begin{aligned} \mathbf{p}_i[k+1] &= \mathbf{p}_i[k] + h\mathbf{v}_i[k] + \frac{h^2}{2}\mathbf{u}_i[k] \\ \mathbf{v}_i[k+1] &= \mathbf{v}_i[k] + h\mathbf{u}_i[k] \end{aligned} \tag{1}$$

where $\mathbf{p}_i[k]$, $\mathbf{v}_i[k]$, $\mathbf{u}_i[k] \in \mathbb{R}^2$ are the position, velocity, and acceleration in the XY plane at discrete time step k respectively with $\mathbf{u}_i[k]$ as the system input. Note that all agents are homogeneous basically. Besides, the dynamic constraints are given by

$$\begin{aligned} \mathbf{u}_i &\in \{\mathbf{u} \mid \|\mathbf{u}\|_\infty \le u_{*,\max}\}, \forall i \in \mathcal{N}_* \\ \mathbf{v}_i &\in \{\mathbf{v} \mid \|\mathbf{v}\|_\infty \le v_{*,\max}\}, \forall i \in \mathcal{N}_* \end{aligned} \tag{2}$$

where $* \in \{p, e\}$.

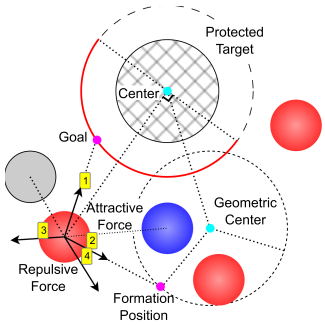

Fig. 2. Illustration of evaders' APF policy. The circle with a reticulate pattern is the protected target. An evader's goal is initialized randomly on the red semicircle arc before the game starts. The goal and formation position of the bottom left evader exert attractive forces 1 and 2, while the blue pursuer and the gray obstacle, assumed to be the only two objects within its detection range, generate repulsive forces 3 and 4. Note that force 2 can be removed to obtain the non-cooperative policy.

B. Conditions for Successful Pursuit

As shown in Fig. 1, we establish a set of criteria for successful pursuit including three parts: the encirclement condition, the distance condition, and the collision-free condition.

The encirclement condition requires that evader $i \in \mathcal{N}_e$ needs to be rounded up by the polygon with all the pursuers in a group as vertices, which can be formulated as

$$\mathbf{p}_i \in \left\{\mathbf{p} \in \mathbb{R}^2 \mid \mathrm{Ind}(\mathcal{G}_i, \mathbf{p}) = 1\right\}, i \in \mathcal{N}_e \tag{3}$$

where $\mathcal{G}_i$ represents the group pursuing evader i and $\mathrm{Ind}(\mathcal{G}_i, \mathbf{p})$ is the winding number of the polygon formed by $\mathcal{G}_i$ about the point $\mathbf{p}$.

The distance condition requires that the distances between the target evader and all the group members are smaller than $d_{pursuit}$. This condition is given by

$$\|\mathbf{p}_i - \mathbf{p}_j\|_2 \leq d_{pursuit}, \forall i \in \mathcal{G}_j \tag{4}$$

which is a circular range around the evader j.

Besides, under the precondition that all the objects are disc-shaped, the collision-free condition is formulated as

$$\|\mathbf{p}_i - \mathbf{p}_j\|_2 \geq r_i + r_j, \forall i, j \in \mathcal{N} \cup \mathcal{O}, i \neq j \tag{5}$$

where $\mathcal{O}$ is a set containing all the obstacles with different radii in the environment.

C. Penetration Policy of Evaders

We utilize the artificial potential field (APF) policy, applying attractive and repulsive forces on evaders, as illustrated in Fig. 2. To enhance the diversity of evaders' cooperative and non-cooperative behavior in case of overfitting, we incorporate adjustable weighting factors in the APF policy. A straightforward approach to foster cooperation is to guide evaders into a simple regular polygonal formation. This formation shape results in an even distribution of the formation positions along a circle, as shown in Fig. 2. Besides, in order to equalize evaders' distances to formation positions, we set the formation center at their geometric center $\sum_{i \in \mathcal{N}_e} \mathbf{p}_i / n_e$. One of the formation positions is the intersection of the circle and the line that connects evaders' geometric center and the origin as illustrated in Fig. 2. Following these principles, the APF control signal for evader i at time step k satisfies

$$\mathbf{u}_i = c_{r,i} \sum_{j \in \mathcal{D}_i} \frac{\mathbf{p}_i - \mathbf{p}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|_2} + \sum_{* \in \{g,f\}} c_{*,i} \frac{\mathbf{p}_{*,i} - \mathbf{p}_i}{\|\mathbf{p}_{*,i} - \mathbf{p}_i\|_2} \tag{6}$$

where k is omitted. $c_{r,i}$, $c_{g,i}$, and $c_{f,i}$ are time-invariant weighting factors and enable evaders to avoid collisions, approach goals, and form a formation, respectively. Note that the $c_{f,i}$ term exerts attractive forces to guide evaders to their formation positions if $c_{f,i} > 0$. Otherwise, they act in a non-cooperative manner. $\mathbf{p}_{g,i}$ and $\mathbf{p}_{f,i}$ are evader i's goal and formation position, respectively. $\mathcal{D}_i$, with the full expression as $\mathcal{D}_i[k] := \{j \mid \|\mathbf{p}_j[k] - \mathbf{p}_i[k]\|_2 \leq r_{d,*} \forall j \in \mathcal{N} \cup \mathcal{O}, j \neq i\}$, represents all the objects within agent i's detection scope. $r_{d,*}$ with $* \in \{p, e\}$ is the detection radius for pursuers and evaders, respectively.

IV. Cooperative Grouping of Pursuers

The cooperative grouping problem of multi-pursuer multi-evader TGP consists of two essential sequential subproblems: the ordinary grouping subproblem and the position allocation subproblem. The former subproblem is to designate the required number of pursuers as a group aimed at a certain evader while the latter one is to allocate positions of a formation to group members. In our distributed pipeline, the cooperative grouping problem is solved by all the pursuers and then they will know their own roles. Due to the changing environment, real-time grouping is necessary throughout the game. These two subproblems are modeled as mixed integer linear programming (MILP) problems with binary variables.

A. The Ordinary Grouping Subproblem

Generally, successful pursuit is easier to achieve when pursuers are closer to evaders. Hence, the overall strategy for this subproblem is to minimize the sum of distances between pursuers and evaders. Group members are adjusted dynamically to guarantee minimal distances. To formulate the strategy, we firstly define a binary variable $y_{ij} \in \{0, 1\}$. $y_{ij} = 1$ means that pursuer i is allocated to group $\mathcal{G}_j$ while $y_{ij} = 0$ is the contrary. Then, the ordinary grouping subproblem can be modeled as

$$\min_{\substack{y_{ij} \text{ for all} \\ i \in \mathcal{N}_p \\ j \in \mathcal{N}_e}} \sum_{i \in \mathcal{N}_p} \sum_{j \in \mathcal{N}_e} y_{ij} \|\mathbf{p}_{ij}\|_2 \tag{7a}$$

$$\text{s.t.} \sum_{j \in \mathcal{N}_e} y_{ij} \leq 1, \forall i \in \mathcal{N}_p \tag{7b}$$

$$\sum_{i \in \mathcal{N}_p} y_{ij} = n_{req}, \forall j \in \mathcal{N}_e \tag{7c}$$

where $\mathbf{p}_{ij} := \mathbf{p}_i - \mathbf{p}_j$ and n_{req} is the number of pursuers required to encircle an evader. (7b) indicates that each pursuer is assigned no more than one evader. (7c) implies that each evader is allocated n_{req} pursuers.

B. The Position Allocation Subproblem

Although (3) only regulates the final result, we intend to form a n_{req}-regular-polygon-shaped formation for pursuit as quickly as possible using a leader-follower framework to resemble real-world cooperation. Then, the subproblem can be divided into two consecutive steps: position allocation and leader-follower framework construction.

Algorithm 1: Leader-follower Framework Construction.

Require: $\mathcal{F}_{result,j}$, $\mathcal{G}_j$, evader j, predicted state sequences $\mathbf{P}_i, \mathbf{V}_i \in \mathbb{R}^{2K}$ for all $i \in \mathcal{G}_j \cup \{j\}$
Ensure: Pursuer $i_j \in \mathcal{G}_j$ as leader, predicted goal state sequences $\mathbf{P}_{g,i}, \mathbf{V}_{g,i} \in \mathbb{R}^{2K}$ for all $i \in \mathcal{G}_j$
$i_j \leftarrow \arg\min_{i\in\mathcal{G}_j} \|\mathbf{p}_i - \mathbf{p}_j\|_2$
$\mathbf{p}_{g,i_j j} \leftarrow \mathbf{p}_{res,i_j} - \mathbf{p}_j$
for all $\mathbf{p}_{res,i} \in \mathcal{F}_{result,j}$ **do**
  $\mathbf{p}_{g,ii_j} \leftarrow \mathbf{p}_{res,i} - \mathbf{p}_{res,i_j}$
end for
for all $i \in \mathcal{G}_j$ **do**
  $i_{ref} \leftarrow$ SelectReferenceAgent($\mathcal{G}_j \cup \{j\}$)
  $\mathbf{P}_{g,i} \leftarrow \mathbf{P}_{i_{ref}} + [\mathbf{p}^{\mathrm{T}}_{g,ii_{ref}}\ \mathbf{p}^{\mathrm{T}}_{g,ii_{ref}}\ \cdots\ \mathbf{p}^{\mathrm{T}}_{g,ii_{ref}}]^{\mathrm{T}}$
  $\mathbf{V}_{g,i} \leftarrow \mathbf{V}_{i_{ref}}$
end for
return $\mathbf{P}_{g,i}, \mathbf{V}_{g,i}$ for all $i \in \mathcal{G}_j$, i_j

First of all, we denote the radius of the regular polygon as $r_{pol} \le d_{pursuit}$ and $\mathcal{N}_{req} := \{0, 1, \ldots, n_{req} - 1\}$. Then the set containing all vertices of a n_{req} regular polygon with its center at the origin is defined as $\mathcal{F}_{pol} := \{\mathbf{p}_{pol,l} \mid l \in \mathcal{N}_{req}\}$ where $\mathbf{p}_{pol,l} = r_{pol}[\cos(2\pi l/n_{req}),\ \sin(2\pi l/n_{req})]^{\mathrm{T}}$ is the $(l+1)$-th vertex of the polygon. Then the formation around evader j can be derived as $\mathcal{F}_j := \{\mathbf{p}_{f,l} \mid l \in \mathcal{N}_{req}\}$ where $\mathbf{p}_{f,l} = \mathbf{p}_{pol,l} + \mathbf{p}_j$.

We define another binary variable $z_{il} \in \{0, 1\}$. $z_{il} = 1$ means that pursuer i is allocated to $\mathbf{p}_{f,l} \in \mathcal{F}_j$ while $z_{il} = 0$ is the opposite. For evader j, the position allocation subproblem can be written as

$$\min_{\substack{z_{il}\ \text{for all}\\ i\in\mathcal{G}_j\\ l\in\mathcal{N}_{req}}} \sum_{i\in\mathcal{G}_j}\sum_{l=0}^{n_{req}-1} z_{il}\|\mathbf{p}_i - \mathbf{p}_{f,l}\|_2 \tag{8a}$$

$$\text{s.t.} \sum_{l=0}^{n_{req}-1} z_{il} = 1, \forall i \in \mathcal{G}_j \tag{8b}$$

$$\sum_{i\in\mathcal{G}_j} z_{il} = 1, \forall l \in \mathcal{N}_{req} \tag{8c}$$

where (8b) and (8c) show one-to-one correspondence between group members and formation positions.

The leader-follower framework is constructed using algorithm 1 as the game evolves. In algorithm 1, K is the length of the prediction horizon. $\mathcal{F}_{result,j} := \{\mathbf{p}_{res,i} \mid i \in \mathcal{G}_j\}$ with pursuer i's formation position $\mathbf{p}_{res,i} \in \mathcal{F}_j$ represents the allocation result. For $\mathcal{G}_j$, $i_{ref} = j$ when $i = i_j$, and $i_{ref} = i_j$ otherwise.

V. Cooperative Trajectory Planning

During the patrolling phase, pursuers are relatively individually driven by PID controllers to guarantee equivalent patrolling interval. However, in the pursuing phase, we propose CEO-DMPC for cooperative trajectory planning. *For the sake of conciseness, this section displays time stamps solely for individual predicted values while omitting them for value sequences and current actual values.*

A. Linear Predictive Model

To formulate a linear expression based on (1) over a horizon of length K, we define the notation $\hat{(\cdot)}[k \mid k_c]$ as the state at future time step $k \in \mathcal{K} := \{0, \ldots, K-1\}$ predicted at k_c. Then we introduce two matrix $\mathbf{A} \in \mathbb{R}^{4\times4}$ and $\mathbf{B} \in \mathbb{R}^{4\times2}$ as

$$\mathbf{A} := \begin{bmatrix} \mathbf{I}_2 & h\mathbf{I}_2 \\ \mathbf{O}_2 & \mathbf{I}_2 \end{bmatrix}, \mathbf{B} := \begin{bmatrix} (h^2/2)\,\mathbf{I}_2 \\ h\mathbf{I}_2 \end{bmatrix} \tag{9}$$

where $\mathbf{I}_2, \mathbf{O}_2 \in \mathbb{R}^{2\times2}$ are the identity and zero matrices, respectively.

With the state vector $\hat{\mathbf{x}}_i[k \mid k_c] := [\hat{\mathbf{p}}_i^{\mathrm{T}}[k \mid k_c]\ \hat{\mathbf{v}}_i^{\mathrm{T}}[k \mid k_c]]^{\mathrm{T}} \in \mathbb{R}^4$, the predictive expression within one step is written as

$$\hat{\mathbf{x}}_i[k+1 \mid k_c] = \mathbf{A}\hat{\mathbf{x}}_i[k \mid k_c] + \mathbf{B}\hat{\mathbf{u}}_i[k \mid k_c] \tag{10}$$

Then we define the position selection matrix $\mathbf{\Psi} := [\ \mathbf{I_2}\quad \mathbf{O}_2\ ]$ and the velocity selection matrix $\mathbf{\Phi} := [\ \mathbf{O_2}\quad \mathbf{I}_2\ ]$. Furthermore, four predictive matrices $\mathbf{A}_{pos}, \mathbf{A}_{vel} \in \mathbb{R}^{2K\times4}$ and $\mathbf{B}_{pos}$, $\mathbf{B}_{vel} \in \mathbb{R}^{2K\times2K}$ need supplementary introduction as below.

$$\mathbf{A}_{pos} = \begin{bmatrix} (\mathbf{\Psi A})^{\mathrm{T}} & (\mathbf{\Psi A}^2)^{\mathrm{T}} & \cdots & (\mathbf{\Psi A}^K)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{11a}$$

$$\mathbf{A}_{vel} = \begin{bmatrix} (\mathbf{\Phi A})^{\mathrm{T}} & (\mathbf{\Phi A}^2)^{\mathrm{T}} & \cdots & (\mathbf{\Phi A}^K)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{11b}$$

$$\mathbf{B}_{pos} = \begin{bmatrix} \mathbf{\Psi B} & \mathbf{O}_2 & \cdots & \mathbf{O}_2 \\ \mathbf{\Psi A B} & \mathbf{\Psi B} & \cdots & \mathbf{O}_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Psi A}^{K-1}\mathbf{B} & \mathbf{\Psi A}^{K-2}\mathbf{B} & \cdots & \mathbf{\Psi B} \end{bmatrix} \tag{11c}$$

$$\mathbf{B}_{vel} = \begin{bmatrix} \mathbf{\Phi B} & \mathbf{O}_2 & \cdots & \mathbf{O}_2 \\ \mathbf{\Phi A B} & \mathbf{\Psi B} & \cdots & \mathbf{O}_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Phi A}^{K-1}\mathbf{B} & \mathbf{\Phi A}^{K-2}\mathbf{B} & \cdots & \mathbf{\Psi B} \end{bmatrix} \tag{11d}$$

Additionally, we notate the initial state at time index k_c as $\mathbf{X}_{c,i} := \mathbf{x}_i[k_c]$. Based on (10) and (11), the position and velocity sequences $\mathbf{P_i}, \mathbf{V_i} \in \mathbb{R}^{2K}$ are given in affine functions of the input sequence $\mathbf{U}_i \in \mathbb{R}^{2K}$ by

$$\mathbf{P}_i = \mathbf{A}_{pos}\mathbf{X}_{c,i} + \mathbf{B}_{pos}\mathbf{U}_i, \quad \mathbf{V}_i = \mathbf{A}_{vel}\mathbf{X}_{c,i} + \mathbf{B}_{vel}\mathbf{U}_i. \tag{12}$$

B. Objective Function

To encircle the evaders with smooth trajectories, the objective function to be minimized is designed as

$$F_i(\mathbf{U}_i) = F_{pos,i} + F_{vel,i} + F_{u,i} + F_{\delta,i} + F_{\epsilon,i} \tag{13}$$

where there are five components on the right-hand side representing position error $F_{pos,i}$, velocity error $F_{vel,i}$, consumed energy $F_{u,i}$, input upheavals $F_{\delta,i}$ and collision avoidance $F_{\epsilon,i}$. Some of them involve future information that will be introduced in Section V-C. *If there is no special explanation, the block-diagonal weight matrix is denoted as $\tilde{\mathbf{W}}_{(\cdot)} \in \mathbb{S}_+^{2K}$ consisting of the diagonal matrix $\mathbf{W}_{(\cdot)} \in \mathbb{S}_{++}^{2}$ with the same subscript in the following.*

1) Position Error Penalty: This term enables the pursuers to form a formation and approach the evaders by minimizing position errors. The quadratic position error penalty is

$$F_{pos,i} = \mathbf{U}_i^{\mathrm{T}}\left(\mathbf{B}_{pos}^{\mathrm{T}}\tilde{\mathbf{W}}_{pos}\mathbf{B}_{pos}\right)\mathbf{U}_i - 2\left(\mathbf{P}_{g,i}^{\mathrm{T}}\tilde{\mathbf{W}}_{pos}\mathbf{B}_{pos} - (\mathbf{A}_{pos}\mathbf{X}_{c,i})^{\mathrm{T}}\tilde{\mathbf{W}}_{pos}\mathbf{B}_{pos}\right)\mathbf{U}_i \tag{14}$$

where $\tilde{\mathbf{W}}_{pos} = \mathrm{diag}(\mathbf{O}_2, \mathbf{W}_{pos}, \ldots, \mathbf{W}_{pos}) \in \mathbb{S}_+^{2K}$.

2) Velocity Error Penalty: This term drives pursuers to move at the consistent velocities of their reference agents. This aim can be converted to the penalty of velocity errors, which is expressed in the quadratic form as

$$F_{vel,i} = \mathbf{U}_i^{\mathrm{T}} \left(\mathbf{B}_{vel}^{\mathrm{T}} \tilde{\mathbf{W}}_{vel} \mathbf{B}_{vel} \right) \mathbf{U}_i - 2 \left(\mathbf{V}_{g,i}^{\mathrm{T}} \tilde{\mathbf{W}}_{vel} \mathbf{B}_{vel} - \left(\mathbf{A}_{vel} \mathbf{X}_{c,i} \right)^{\mathrm{T}} \tilde{\mathbf{W}}_{vel} \mathbf{B}_{vel} \right) \mathbf{U}_i \quad (15)$$

where $\tilde{\mathbf{W}}_{vel} = \mathrm{diag}(\mathbf{O}_2, \mathbf{W}_{vel}, \ldots, \mathbf{W}_{vel}) \in \mathbb{S}_+^{2K}$.

3) Consumed Energy Penalty: For the sake of energy saving, we add a term representing consumed energy in the quadratic function as

$$F_{u,i} = \mathbf{U}_i^{\mathrm{T}} \tilde{\mathbf{W}}_u \mathbf{U}_i \quad (16)$$

where $\tilde{\mathbf{W}}_u = \mathrm{diag}(\mathbf{W}_u, \ldots, \mathbf{W}_u) \in \mathbb{S}_{++}^{2K}$.

4) Input Upheaval Penalty: Smooth trajectories require gradually changing inputs. Therefore, a term penalizing input variation is added in the quadratic form as

$$F_{\delta,i} = \mathbf{U}_i^{\mathrm{T}} \left(\mathbf{S}^{\mathrm{T}} \tilde{\mathbf{W}}_\delta \mathbf{S} \right) \mathbf{U}_i - 2 \left(\overline{\mathbf{U}}_i^{\mathrm{T}} \tilde{\mathbf{W}}_\delta \mathbf{S} \right) \mathbf{U}_i \quad (17)$$

where $\tilde{\mathbf{W}}_\delta = \mathrm{diag}(\mathbf{W}_\delta, \ldots, \mathbf{W}_\delta) \in \mathbb{S}_{++}^{2K}$. $\mathbf{S} \in \mathbb{R}^{2K \times 2K}$ for subtraction of adjacent states and $\overline{\mathbf{U}}_i \in \mathbb{R}^{2K}$ are given by

$$\mathbf{S} = \mathbf{I}_{2K} - \begin{bmatrix} \mathbf{O}_{2\times 2(K-1)} & \mathbf{O}_2 \\ \mathbf{I}_{2(K-1)} & \mathbf{O}_{2(K-1)\times 2} \end{bmatrix} \quad (18a)$$

$$\overline{\mathbf{U}}_i = \begin{bmatrix} \mathbf{u}_i [k_c - 1]^{\mathrm{T}} \mathbf{O}_{2\times 1}^{\mathrm{T}} \cdots \mathbf{O}_{2\times 1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}. \quad (18b)$$

5) Collision Avoidance Term: The collision-free condition (5) is a set of hard-to-use nonlinear constraints. Therefore, we add distance penalty $F_{\epsilon,i}$ as soft collision avoidance constraints. The collision avoidance term is formulated as

$$F_{\epsilon,i} = \mathbf{U}_i^{\mathrm{T}} \left(\mathbf{B}_{pos}^{\mathrm{T}} \tilde{\mathbf{W}}_\epsilon \mathbf{B}_{pos} \right) \mathbf{U}_i - 2 \left(\sum_{j \in \mathcal{D}_i} \left(\mathbf{P}_{ca,ij}^{\mathrm{T}} \tilde{\mathbf{W}}_{\epsilon,j} \right) \mathbf{B}_{pos} - \left(\mathbf{A}_{pos} \mathbf{X}_{c,i} \right)^{\mathrm{T}} \tilde{\mathbf{W}}_\epsilon \mathbf{B}_{pos} \right) \mathbf{U}_i \quad (19)$$

where $\tilde{\mathbf{W}}_\epsilon = \sum_{j \in \mathcal{D}_i} \tilde{\mathbf{W}}_{\epsilon,j} \in \mathbb{S}_+^{2K}$ and the weight matrix for object $j \in \mathcal{D}_i$ is $\tilde{\mathbf{W}}_{\epsilon,j} = \mathrm{diag}(\mathbf{O}_2, \mathbf{W}_{\epsilon,j}, \ldots, \mathbf{W}_{\epsilon,j}) \in \mathbb{S}_+^{2K}$. The predicted safe position sequence $\mathbf{P}_{ca,ij}$ is given by

$$\mathbf{P}_{ca,ij} = \begin{cases} \mathbf{P}_j + (r_i + r_j + d_{safe}) \overline{\mathbf{P}}_{ij}^{norm}, & j \in \mathcal{N} \\ \mathbf{P}_i + (r_i + r_j + d_{safe} - \|\mathbf{p}_{ij}\|_2) \overline{\mathbf{P}}_{ij}^{norm}, & j \in \mathcal{O} \end{cases} \quad (20)$$

where $\overline{\mathbf{P}}_{ij}^{norm} = \|\mathbf{p}_{ij}\|_2^{-1} [\mathbf{p}_{ij}^{\mathrm{T}} \, \mathbf{p}_{ij}^{\mathrm{T}} \cdots \mathbf{p}_{ij}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{2K}$ drives pursuer i away from object j and d_{safe} is the safe distance.

C. Prediction of State Sequences

Four types of objects require state sequence prediction: patrolling and pursuing pursuers, evaders, and obstacles. The future states of pursuing pursuers can be easily acquired by applying the solutions of CEO-DMPC to the linear model (12). For static obstacles, the future state sequences are comprised entirely of their respective fixed states. However, the states of patrolling pursuers and evaders need predicting as shown in Algorithm 2.

In algorithm 2, $\mathcal{G} := \bigcup_{i \in \mathcal{N}_e} \mathcal{G}_i$. π_i represents unknown APF policy if $i \in \mathcal{N}_e$ and the known PID controller if $i \in \mathcal{N}_p \setminus \mathcal{G}$. Considering that pursuers only know evaders' intention instead of specific policies, we set $c_{r,i} = c_{f,i} = 0$, $c_{g,i}$ as intermediate value of variation range, and $\mathbf{p}_{g,i} = [0, 0]^{\mathrm{T}}, \forall i \in \mathcal{N}_e$, only in the prediction process. Therefore, the prediction will help pursuers to stay between evaders and their goals for interception [27], [28], [29], [30]. Note that the actual observation $\mathcal{D}_i$ in (6) unknown for pursuers is approximated and replaced by the predicted integrated observation $\mathcal{D}[k \mid k_c], k \in \mathcal{K}$ only in the prediction process for consistency of expression. To account for the coupling between state prediction and CEO-DMPC, we assume uniform linear motion for all the evaders to establish the first prediction.

Algorithm 2: State Prediction.

Require: $\mathcal{G}$, $\mathcal{D}_i$ for all $i \in \mathcal{N}_p$, agent i's policy π_i for all $i \in \mathcal{N} \setminus \mathcal{G}$
Ensure: $\mathbf{P}_i, \mathbf{V}_i \in \mathbb{R}^{2K}$ for all $i \in \mathcal{N} \setminus \mathcal{G}$
$\mathcal{D}[0 \mid k_c] \leftarrow \bigcup_{i \in \mathcal{N}_p} \mathcal{D}_i$
for all $k \in \mathcal{K}$ **do**
  for all $i \in \mathcal{N} \setminus \mathcal{G}$ **do**
    $\mathbf{u}_i[k \mid k_c] \leftarrow \pi_i(\mathcal{D}[k \mid k_c])$
    $\mathbf{p}_i[k+1 \mid k_c], \mathbf{v}_i[k+1 \mid k_c] \leftarrow \mathrm{Dynamics}(\mathbf{u}_i[k \mid k_c])$
  end for
  $\mathcal{D}[k+1 \mid k_c] \leftarrow \mathrm{FuseNewPredictions}(\mathcal{N})$
end for
for all $i \in \mathcal{N}_p \setminus \mathcal{G}$ **do**
  $\mathbf{P}_i \leftarrow [\mathbf{p}_i^{\mathrm{T}}[1 \mid k_c] \, \mathbf{p}_i^{\mathrm{T}}[2 \mid k_c] \cdots \mathbf{p}_i^{\mathrm{T}}[K \mid k_c]]^{\mathrm{T}}$
  $\mathbf{V}_i \leftarrow [\mathbf{v}_i^{\mathrm{T}}[1 \mid k_c] \, \mathbf{v}_i^{\mathrm{T}}[2 \mid k_c] \cdots \mathbf{v}_i^{\mathrm{T}}[K \mid k_c]]^{\mathrm{T}}$
end for
return $\mathbf{P}_i, \mathbf{V}_i$ for all $i \in \mathcal{N} \setminus \mathcal{G}$

D. The Final Optimization Problem

According to the dynamic constraints (2), we define boundary vectors $\mathbf{V}_b, \mathbf{U}_b \in \mathbb{R}^{2K}$ as

$$\mathbf{V}_b := [v_{p,\max} v_{p,\max} \cdots v_{p,\max}]^{\mathrm{T}}$$
$$\mathbf{U}_b := [u_{p,\max} u_{p,\max} \cdots u_{p,\max}]^{\mathrm{T}}. \quad (21)$$

Then, the dynamic limits can be written as

$$-\mathbf{V}_b - \mathbf{A}_v \mathbf{X}_{c,i} \le \mathbf{B}_v \mathbf{U}_i \le \mathbf{V}_b - \mathbf{A}_v \mathbf{X}_{c,i}$$
$$-\mathbf{U}_b \le \mathbf{U}_i \le \mathbf{U}_b \quad (22)$$

the solution space of which can be expressed in a more succinct way as $\mathbf{A}_{bdy} \mathbf{U}_i \le \mathbf{b}_{bdy}$, $\mathbf{A}_{bdy} \in \mathbb{R}^{4K \times 2K}$, $\mathbf{b}_{bdy} \in \mathbb{R}^{4K}$.

The concentrated expression of the final optimization problem for CEO-DMPC is given by

$$\min_{\mathbf{U}_i} \; F_i(\mathbf{U}_i)$$
$$\text{s.t. } \mathbf{A}_{bdy} \mathbf{U}_i \le \mathbf{b}_{bdy} \quad (23)$$

Finally, the complete pipeline of CEO-DMPC for TGP is presented in algorithm 3.

VI. Experiments

CEO-DMPC combined with the grouping strategy is implemented in Python3 environment to validate its effectiveness. A quadratic programming (QP) solver, namely Quadprog, is applied to numerically solve the final optimization problem. In addition, the linear solver module pywraplp in OR-Tools is applied to solve MILP problem.

A. Numerical Simulations

We first validate the CEO-DMPC through numerical simulations with different parameter sets. CEO-DMPC is compared with the traditional non-cooperative PID controller combined

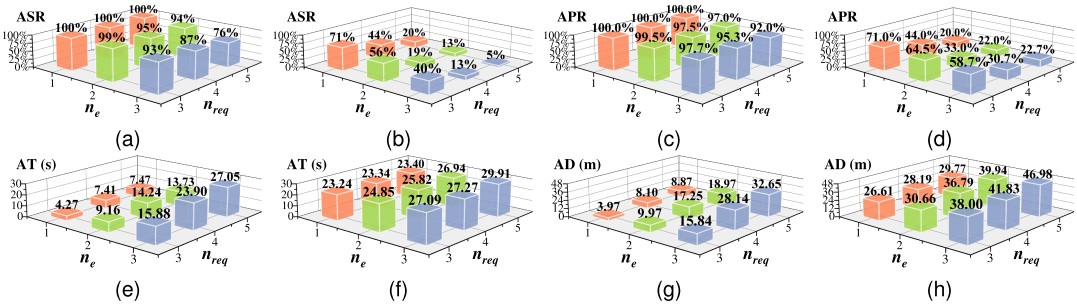

Fig. 3. Comparison of different statistics of CEO-DPMC and PID. There are four record indices: the average success rate (ASR), the average progress rate (APR), the average time (AT) consumed, and the average distance (AD) traveled by all the pursuers. APR represents the average proportion of captured evaders in total. (a) ASR of CEO-DMPC. (b) ASR of PID. (c) APR of CEO-DMPC. (d) APR of PID. (e) AT of CEO-DMPC. (f) AT of PID. (g) AD of CEO-DMPC. (h)AD of PID.

TABLE I
SIMULATION PARAMETERS

Parameters	Values	Parameters	Values	Parameters	Values
n_p	15	n_e	1-3	r_{patrol}	20 m
r_p	0.45 m	r_e	0.45 m	r_{pol}	3.2 m
$r_{d,p}$	8.0 m	$r_{d,e}$	8.0 m	K	50
$v_{p,\max}$	2.0 m/s	$v_{e,\max}$	2.0 m/s	h	0.05 s
$u_{p,\max}$	2.0 m/s^2	$u_{e,\max}$	2.1 m/s^2	d_{safe}	5.0 m
n_{req}	3-5	$d_{pursuit}$	4.0 m		

Algorithm 3: CEO-DMPC Pipeline for TGP.

Require: All information
Ensure: $\mathbf{U}_i \in \mathbb{R}^{2\,K}$ for all $i \in \mathcal{G}$
$\mathcal{G}_j$ for all $j \in \mathcal{N}_e \leftarrow$ OrdinaryGrouping($\mathcal{N}_p, \mathcal{N}_e, n_{req}$)
for all $j \in \mathcal{N}_e$ **do**
  $\mathcal{F}_j \leftarrow$ GetFormationPosition(n_{req}, j)
  $\mathcal{F}_{result,j} \leftarrow$ PositionAllocation($\mathcal{F}_j, \mathcal{G}_j, j$)
end for
$\mathbf{P}_i, \mathbf{V}_i$ for all $i \in \mathcal{N} \leftarrow$ StatePrediction($\mathcal{N}$)
for all $j \in \mathcal{N}_e$ **do**
  $\mathbf{P}_{g,i}, \mathbf{V}_{g,i}$ for all $i \in \mathcal{G}_j \leftarrow$ LeaderFollowerFramework($\mathcal{F}_{result,j}, \mathcal{G}_j, j$)
end for
for all $i \in \mathcal{G}$ **do**
$\mathbf{U}_i \leftarrow$ CEO-DPMC($\mathbf{P}_{g,i}, \mathbf{V}_{g,i}, \mathcal{D}_i$)
end for
return $\mathbf{U}_i$ for all $i \in \mathcal{G}$

with APF (abbreviated as PID below) to avoid collision. To ensure that successful encirclement can be achieved, a necessary condition $n_p \geq n_e \times n_{req}$ needs satisfying. Furthermore, we set $r_{patrol} \leq n_p r_{d,p}/\pi$ to avoid blind intervals between patrolling pursuers and assume full real-time communication for the pursuing side. Table I presents the qualified parameters. We conduct 100 rounds of the game for each set of parameters specified in Table I. For pursuers, we define $\mathbf{W}_{(\cdot)} := \mathrm{diag}(w_{(\cdot)}, w_{(\cdot)})$ in Section V-B. Then $w_{pos} : w_{vel} : w_u : w_\delta : w_\epsilon$ for pursuer $i \in \mathcal{G}$ has two values: 1) $8 : 1.5 : 1 : 1 : 1$ for $\|\mathbf{p}_{ij}\|_2 \geq r_i + r_j + 2.0, \forall j \in \mathcal{D}_i$, 2) $8 : 1.5 : 1 : 1 : 16$ otherwise. Considering $\sum_{j\in\mathcal{D}_i} w_{\epsilon,i} = w_\epsilon$, we set $w_{\epsilon,i} \propto (\|\mathbf{p}_{ij}\|_2 - r_i - r_j)^{-2}$. As for evaders, the weighting factors $c_{r,i}$, $c_{g,i}$, $c_{f,i}$, $\forall i \in \mathcal{N}_e$, set as random variables, are sampled once before each round. Specifically, $c_{r,i}$, $c_{g,i} \sim U(10, 40), \forall i \in \mathcal{N}_e$, and $P(c_{f,i} = 0, \forall i \in \mathcal{N}_e) = P(c_{f,i} \in [10, 40], \forall i \in \mathcal{N}_e) = 0.5$ with its probability density function $f(c_{f,i}) = 1/60, c_{f,i} \in [10, 40], \forall i \in \mathcal{N}_e$. This distribution implies heterogeneous APF policies and an equal probability of non-cooperation and cooperation for the evader side. Besides, evaders are initially positioned randomly outside pursuers' circular patrolling trajectory and detection range. However, non-cooperative evaders are mutually independent, while cooperative evaders have fixed relative positions to facilitate formation process.

Fig. 3 illustrates the four record indices defined in its caption. Compared with PID, CEO-DMPC realized higher ASR and APR with lower AT and AD, indicating its superior effectiveness, efficiency, and economy. Besides, mission success is substantially guaranteed with $n_e = 1$ and $n_{req} \in [3, 4, 5]$. CEO-DMPC is also more robust than PID in terms of n_e and n_{req} for less deterioration in indices in more challenging scenarios. In the most demanding scenarios with $n_e = 3$, CEO-DMPC still maintains satisfactory ASR and APR no less than 76% and 92.0% respectively, while PID suffers poor ASR and APR as low as 5% and 22.7% respectively.

Fig. 4 shows the simulation results of the most difficult scenarios involving cooperative evaders. As aforementioned, the evaders have fixed initial relative positions, resulting in challenging dense distribution as explained in Section VI-C. The three enlarged parts in Fig. 4 highlight that CEO-DMPC effectively ensures the presence of a pursuer positioned between its target evader and the protected area. Consequently, evaders cannot infiltrate easily, leading to improved success rates. These details further prove the adaptability of CEO-DMPC across diverse initial conditions.

B. Physical Simulations

Physical simulations are conducted in Gazebo on a laptop computer running Ubuntu 20.04 with an AMD Ryzen 7 5800F CPU and a NVIDIA GeForce GTX 3060 SUPER GPU.

The parameters in the physical simulations are similar to the numerical simulations. The only main difference lies in the reduced number of agents due to insufficient computer performance. The utilized holonomic quad-rotor UAVs belong to a PX4 mode named IRIS. IRIS has an approximately rectangular shape, with a width of 0.52 m and a length of 0.66 m, similar to the size of agents in the numerical simulations.

Fig. 5 illustrates the physical simulation results. All the agents start on the ground and then ascend to and maintain a fixed altitude of 5 m throughout the simulation. In Fig. 5, curved trajectories of the evaders signify their evasive actions, while pursuers accomplish the task in different cases. The results aptly display the versatility of CEO-DMPC in adapting to various initial conditions.

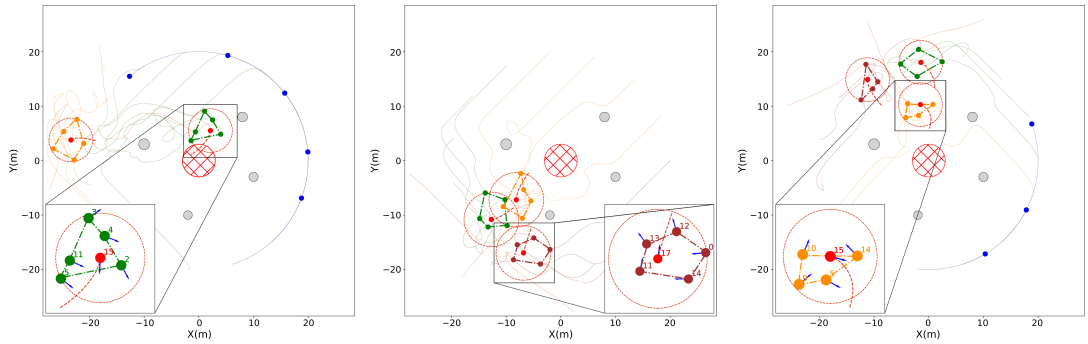

Fig. 4. Numerical simulation results with the most challenging n_e and n_{req} involving cooperative evaders. The dotted circle around each red evader represents the range satisfying (4). The polygons formed by group members are visualized in dotted segments. The blue arrows indicate current speed directions, and the red dashed lines are evaders' predicted trajectories. (a) $n_e = 2$, $n_{req} = 5$. (b) $n_e = 3$, $n_{req} = 5$. (c) $n_e = 3$, $n_{req} = 4$.

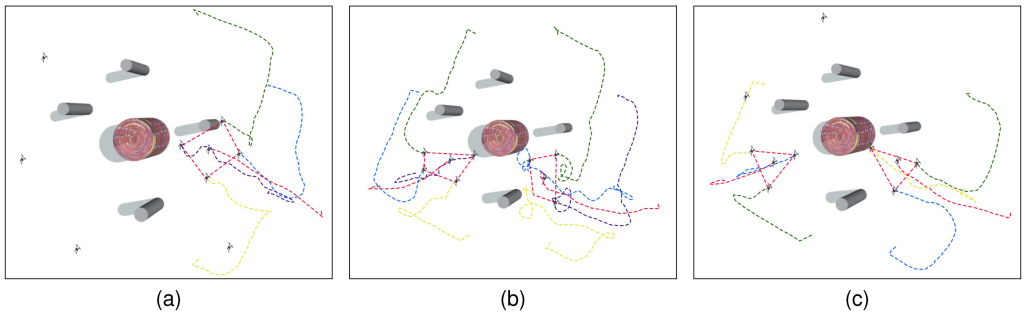

Fig. 5. Physical simulation results in Gazebo with $n_p = 8$. Evaders' trajectories are red dotted curves, while pursuers' are in other colors. The trajectories of patrolling pursuers are omitted for clearness. The final encirclement polygons formed by each group are visualized in red dotted segments. The cylinders represent inaccessible buildings. Specifically, the brick one is the protected target with a height of 10 m and a radius of 3 m, and the gray ones are ordinary obstacles. The UAVs are visualized as larger images of IRIS for obviousness. (a) $n_e = 1$, $n_{req} = 4$. (b) $n_e = 2$, $n_{req} = 4$. (c) $n_e = 2$, $n_{req} = 3$.

TABLE II
AVERAGE TIME CONSUMPTION STATISTICS

Items \ Scenarios	$n_e = 1$, $n_{req} = 4$	$n_e = 2$, $n_{req} = 4$	$n_e = 2$, $n_{req} = 3$
Ordinary Grouping	1.35 *ms*	1.43 *ms*	1.36 *ms*
Position Allocation	1.30 *ms*	1.29 *ms*	0.93 *ms*
CEO-DMPC	5.35 *ms*	5.51 *ms*	4.91 *ms*
Control Delay	55.5 *ms*	67.0 *ms*	56.6 *ms*
Control Frequency	18.0 *Hz*	14.9 *Hz*	17.7 *Hz*

Table II presents the average time consumption for different stages. The time consumption of the ordinary grouping subproblem remains stable, whereas that of the position allocation subproblem is sensitive to n_{req}. QP in CEO-DMPC requires more time than grouping and is more time-consuming with bigger n_{req}, which can lead to more complex local interaction and goal state sequences. The control delay, equivalent to the control period, is influenced by both n_e and n_{req}.

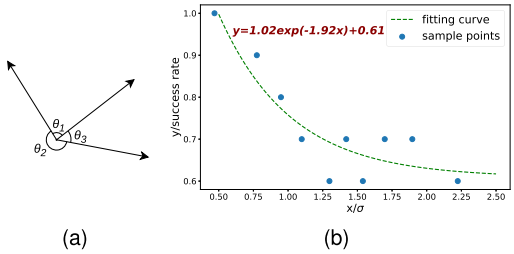

Fig. 6. Analysis of influence from distribution density. We conduct 100 rounds of the game with $n_e = 3$, $n_{req} = 5$, and the non-cooperative evading side. The indicator σ is calculated from the included angles, which represent the differences between evaders' initial polar angles. The included angles satisfy $\sum_{i=1}^{n_e} \theta_i \equiv 2\pi$. Then, the raw samples are arranged in ascending order according to σ and divided into ten groups sequentially. We computed the success rate for each group, resulting in 10 processed samples shown as blue dots. The processed samples are fitted with a green dashed line representing the exponential function in dark-red bold font. (a) Definition of the included angles. (b) Processed samples and fitting results.

C. Analysis of Influencing Factors

This section will identify several key factors that influence the mission results.

One important factor is the initial density of evaders. A dense distribution of evaders leads to frequent adjustments in the grouping, a delicate balance between collision avoidance and capture, longer travel distances for pursuers located far from the evaders, and unpredictable interactions between groups. To quantify the degree of density, we design an indicator $\sigma := \sqrt{\sum_{i=1}^{n_e}(\theta_i - 2\pi/n_e)^2}$, resembling a standard deviation. In Fig. 6, the green dashed line indicates that the success probability initially decreases rapidly and then stabilizes as σ increases. This observation is further supported by a strong positive correlation, with a correlation coefficient of $\rho = 0.914$, between the success rate and the exponential term of σ.

Another crucial aspect pertains to the weighting factors of CEO-DMPC. w_{pos} has the most direct influence in guiding pursuers into formation and encirclement, hence it is biggest. The second biggest w_{vel} plays a vital role in maintaining the encirclement trend and the formation by synchronizing the velocities of pursuers and their reference agents, especially when

$F_{pos,i}$ is minimal. w_u and w_δ smooth the trajectories but restrict the flexibility of pursuers, which is important for effective encirclement. Consequently, their ratios are deliberately kept low to maintain a fundamental level of smoothness. w_ϵ facilitates collision avoidance but can lead to increased $F_{pos,i}$. Thereby, w_ϵ is initially relatively small but becomes the largest when pursuers are close to surrounding objects with a risk of collision. Furthermore, we prioritize the avoidance of nearby objects with a higher probability of collision by assigning larger $w_{\epsilon,j}$ to them. The final proportions are manually adjusted according to the performance of the simulation.

Several other factors also affect the outcomes of the task. Cooperation for evaders that contributes to dense distribution makes the mission more challenging. Densely-distributed and large-sized static obstacles can reduce the success rate. Additionally, smaller r_{pol} facilitates the encirclement trend but also increases the risk of collisions. Shorter K and h weaken the prediction ability but enhance the frequency of control by reducing the computational burden.

VII. Conclusion

This paper proposes CEO-DMPC for multi-pursuer multi-evader TGP with varying initial conditions, including the numbers of evaders and the criteria for successful pursuit. The whole framework consisting of the cooperative grouping strategy and CEO-DMPC is adaptive to different initial conditions. Based on grouping results, CEO-DMPC produces trajectories for each group member to realize cooperative capture. Instead of a final state, sequences of predicted states are utilized in the objective function to react to evaders' motions ahead of time. The high performance, adaptability, and generalization of CEO-DMPC are validated in both numerical and physical simulations close to real applications.

There still exist future research opportunities to enhance CEO-DMPC. The two grouping strategies can incorporate dynamic models and states in addition to distances to achieve more reasonable assignments. Exploring more elaborate and realistic dynamic models and constraints is warranted. Data-driven methods can replace manual tuning to adjust the weighting matrices in CEO-DMPC. Moreover, to achieve more precise predictions, advanced methods including the Monte Carlo method, support vector machine (SVM), neural network (NN), and inverse reinforcement learning (IRL) [31] can be employed. Further investigation can also be conducted through real-world experiments, deeper theoretical analysis, and the inclusion of more realistic patrolling phase.

References

[1] I. E. Weintraub, M. Pachter, and E. Garcia, "An introduction to pursuit-evasion differential games," in *Proc. Amer. Control Conf.*, 2020, pp. 1049–1066.

[2] A. Manoharan and P. B. Sujit, "NMPC-Based cooperative strategy to lure two attackers into collision by two targets," *IEEE Contr. Syst. Lett.*, vol. 7, pp. 496–501, 2023.

[3] J. A. Bather and R. Isaacs, "Differential games: A mathematical theory with applications to warfare and pursuit, control and optimization," *J. Roy. Stat. Soc. Ser. A (General)*, vol. 129, no. 3, pp. 474–475, 1966.

[4] R. H. Venkatesan and N. K. Sinha, "The target guarding problem revisited: Some interesting revelations," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 1556–1561, Jan. 2014.

[5] J. Mohanan, S. R. Manikandasriram, R. H. Venkatesan, and B. Bhikkaji, "Toward real-time autonomous target area protection: Theory and implementation," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 1293–1300, May 2019.

[6] H. Fu and H. H.-T. Liu, "Guarding a territory against an intelligent intruder: Strategy design and experimental verification," *IEEE/ASME Trans. Mechatron.*, vol. 25, no. 4, pp. 1765–1772, Aug. 2020.

[7] G. Merkulov, M. Weiss, and T. Shima, "Virtual target approach for multi-evader intercept," in *Proc. Eur. Control Conf.*, 2022, pp. 1491–1496.

[8] D. Shishika, D. Maity, and M. Dorothy, "Partial information target defense game," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 8111–8117.

[9] D. W. Oyler, P. T. Kabamba, and A. R. Girard, "Pursuitevasion games in the presence of obstacles," *Automatica*, vol. 65, pp. 1–11, Mar. 2016.

[10] R. Yan, Z. Shi, and Y. Zhong, "Guarding a subspace in high-dimensional space with two defenders and one attacker," *IEEE Trans. Cybern.*, vol. 52, no. 5, pp. 3998–4011, May 2022.

[11] J. Mohanan, N. Kothuri, and B. Bhikkaji, "The target guarding problem: A real time solution for noise corrupted measurements," *Eur. J. Control*, vol. 54, pp. 111–118, Jul. 2020.

[12] Y. Lee and E. Bakolas, "Optimal strategies for guarding a compact and convex target set: A differential game approach," in *Proc. IEEE 60th Conf. Decis. Control*, 2021, pp. 4320–4325.

[13] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier for safety-critical control," in *Proc. IEEE 60th Conf. Decis. Control*, 2021, pp. 6814–6821.

[14] M. Sani, B. Robu, and A. Hably, "Pursuit-evasion games based on game-theoretic and model predictive control algorithms," in *Proc. Int. Conf. Control, Automat. Diagnosis*, 2021, pp. 1–6.

[15] T. Qiu, H. Zhang, and J. Wang, "Nash pursuit strategy for nonzero-sum MPC game via inverse optimal control," in *Proc. 13th Asian Control Conf.*, 2022, pp. 2354–2359.

[16] N. T. Hung, N. Crasta, D. Moreno-Salinas, A. M. Pascoal, and T. A. Johansen, "Range-based target localization and pursuit with autonomous vehicles: An approach using posterior CRLB and model predictive control," *Robot. Auton. Syst.*, vol. 132, Oct. 2020, Art. no. 103608.

[17] C. Wang, H. Chen, J. Pan, and W. Zhang, "Encirclement guaranteed cooperative pursuit with robust model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 1473–1479.

[18] J. Chen, W. Zha, Z. Peng, and D. Gu, "Multi-player pursuitevasion games with one superior evader," *Automatica*, vol. 71, pp. 24–32, Sep. 2016.

[19] G. A. Ribeiro, N. Mahmoudian, and M. Rastgaar, "Time-efficient target guarding using constrained optimization in 3D space," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot.*, 2018, pp. 1–6.

[20] Y. Cui, C. Zheng, J. Liu, H. Wang, R. Hu, and Z. Wang, "The research of aircraft pursuit-evasion game based on improved DQN," in *Proc. 3rd Int. Conf. Unmanned Syst.*, 2020, pp. 857–862.

[21] Y. Luo, T. Gang, and L. Chen, "Research on target defense strategy based on deep reinforcement learning," *IEEE Access*, vol. 10, pp. 82329–82335, 2022.

[22] J. Yoo, H. Seong, D. H. Shim, J. H. Bae, and Y.-D. Kim, "Deep reinforcement learning-based intelligent agent for autonomous air combat," in *Proc. IEEE/AIAA 41st Digit. Avionics Syst. Conf.*, 2022, pp. 1–9.

[23] Y. Wu, Y. Lei, Z. Zhu, X. Yang, and Q. Li, "Dynamic multitarget assignment based on deep reinforcement learning," *IEEE Access*, vol. 10, pp. 75998–76007, 2022.

[24] A. Asgharnia, H. M. Schwartz, and M. Atia, "Deception in a multi-agent adversarial game: The game of guarding several territories," in *Proc. IEEE Symp. Ser. Comput. Intell.*, 2020, pp. 1321–1327.

[25] A. Asgharnia, H. M. Schwartz, and M. Atia, "Multi-invader multi-defender differential game using reinforcement learning," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2022, pp. 1–8.

[26] W. Bian, C. Wang, K. Huang, Y. Jia, C. Liu, and Y. Mi, "Cooperative strike target assignment algorithm based on deep reinforcement learning," in *Proc. Int. Conf. Cyber- Phys. Social Intell.*, 2022, pp. 659–663.

[27] J. Sgall, "Solution of david Gale's lion and man problem," *Theor. Comput. Sci.*, vol. 259, no. 1-2, pp. 663–670, May 2001.

[28] M. Casini and A. Garulli, "A novel family of pursuit strategies for the lion and man problem," in *Proc. IEEE 56th Annu. Conf. Decis. Control*, 2017, pp. 6436–6441.

[29] M. Casini and A. Garulli, "An improved lion strategy for the lion and man problem," *IEEE Contr. Syst. Lett.*, vol. 1, no. 1, pp. 38–43, Jul. 2017.

[30] E. Garcia, D. W. Casbeer, A. V. Moll, and M. Pachter, "Pride of lions and man differential game," in *Proc. IEEE 59th Conf. Decis. Control*, 2020, pp. 5380–5385.

[31] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 652–674, Sep. 2022.