Proceedings of the 39th Chinese Control Conference
July 27-29, 2020, Shenyang, China

# 3-D Multi-player Pursuit-evasion Game with a Faster Evader

Xu Fang[1], Cheng Cheng[2,3], Lihua Xie[1,2,3]

1. School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore
E-mail: fa0001xu@e.ntu.edu.sg; elhxie@ntu.edu.sg

2. Department of Control Science and Engineering, Tongji University, Shanghai 201804, P. R. China
E-mail: fengyingcc@tongji.edu.cn

3. Shanghai Research Institute for Intelligent Autonomous System, Tongji University, Shanghai 201203, P. R. China

**Abstract:** This paper studies two kinds of multi-player pursuit-evasion games in 3-D space: Two-pursuer one-evader game and multi-pursuer one-evader game. The main challenges are that the evader has faster speed than a group of pursuers and the moving directions of the pursuers and faster evader are unknown to each other. First, the concept of occupied angle is introduced to represent the dominated spaces by the pursuers. Then, escape algorithm and pursuit algorithm are designed for the evader and pursuers, respectively. Finally, escape condition and capture condition of the faster evader and pursuers are presented. Some simulation examples are given to illustrate the theoretical results.

**Key Words:** 3-D Space, Pursuit-evasion Game, Faster Evader, Unknown Moving Direction

## 1 Introduction

In recent years, artificial intelligence and multiagent systems are gaining increasing attention from researchers [1, 2]. The multiagent systems are usually related to controlling multiple mobile agents to implement some tasks in industrial and civil domain. For example, in a pursuit-evasion game, the evaders try to escape from capture, while a group of pursuers aim to capture the evaders by cooperation. In this paper, we focus on 3-D multi-pursuer one-evader game.

The multiagent pursuit-evasion game is usually modeled as a multiagent differential game [3], which is formulated by a Hamilton-Jacobi-Isaacs (HJI) partial differential equation. The two-player or three-player differential game in different scenarios can be defined through a value function using HJI equation. The optimal pursuit strategy for the pursuers is obtained by minimizing the value function, while the optimal escape strategy for the evader is derived by maximizing the value function. However, when it comes to the case of multiple players, the HJI equations may suffer from the curse of dimensionality, and the terminal manifold for a large number of agents is difficult to be determined. The solutions to these problems can be divided into three categories.

1) The pursuit-evasion game is divided into several subgames, and then the optimal strategy for each subgame is derived independently. The multiplayer pursuit-evasion game with an equal number of attackers and defenders is decomposed into a pair of subgames [4], where each subgame involving at most two players is solved by the HJI framework independently [5]. The pursuit-evasion differential game can also be formulated as multistage problem [6], where a time-optimal path through a sequence of goal sets is obtained.
2) The explicit policy in a relatively simple game environment is used to avoid solving HJI equations. Inspired by collective behaviors in various fish species, the multiphase cooperative strategy for pursuers [7] confines the evader to a bounded region. Foraging swarm behavior is modeled as a noncooperative game that has a unique Nash equilibrium described by explicit expressions [8].
3) Calculating the optimal trajectories by numerical approximation based on a known terminal manifold. A discrete semi-Lagrangian scheme is utilized to compute Kruzkov transform of value function numerically. The single-network adaptive dynamic programming (ADP) is used to solve the differential game [9], and the uniform boundedness of the game is demonstrated by Lyapunov theory. In addition, ADP with experience relay algorithm is proposed [10] to solve nonlinear HJI equations for game systems with unknown dynamics.

However, the aforementioned work mostly assume that the maximum speed of the pursuers is no smaller than that of the evader, which render their approaches impracticable to the case that the evader has faster speed than the pursuers. For the case that the evader moves faster than the pursuers, the existing literature [11–16] mainly focus on 2-D pursuit-evasion game. Little has been done for the pursuit-evasion game with a faster evader in 3-D space. Rodriguez and Janosov [17, 18] require that each slower pursuer move toward to the faster evader directly, which does not consider cooperation among the pursuers.

In this paper, we study two kinds of multiplayer pursuit-evasion games in 3-D space: two-pursuer one-evader and multi-pursuer one-evader game, where the evader has faster speed than the pursuers, and their moving directions are unknown to each other. For each kind of game, pursuit algorithm and escape algorithm are designed for the pursuers and faster evader to maximize their chance of success. Moreover, capture condition and escape condition of the pursuers and faster evader are derived.

This paper is organized as follows: Preliminaries are given in Section 2. In section 3, two-pursuer one-faster-evader game with point capture in 3-D space is addressed. Section 4 solves 3-D multi-pursuer one-faster-evader game with non-zero capture radius. Some numerical examples are given in Section 5 to illustrate the theoretical results. Section 6 ends this paper with some conclusions.

This work is partially supported by ST Engineering-NTU Corporate Lab under the NRF Corporate Lab @ University Scheme, National Key R&D Program of China, No. 2018YFE0105000, and National Natural Science Foundation of China (NSFC) under Grant 51475334.

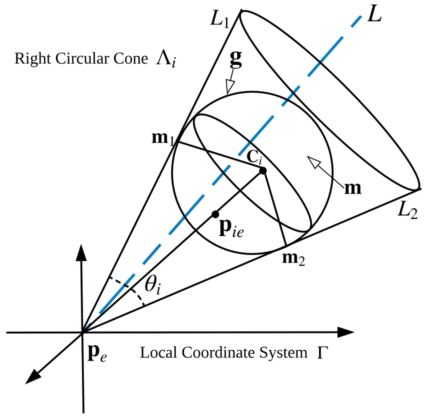


Fig. 1: The occupied angle

# 2 Preliminaries

## 2.1 Kinematic Model of the Pursuers and Evader

Consider a group of $n$ pursuers and a faster evader with single-integrator dynamics, described by

$$\dot{\mathbf{p}}_e = \mathbf{v}_e, \quad ||\mathbf{v}_e||_2 = V_e, \tag{1}$$

$$\dot{\mathbf{p}}_i = \mathbf{v}_i, \quad ||\mathbf{v}_i||_2 = V_i \quad i = 1, 2, \cdots, n, \tag{2}$$

where $||\cdot||_2$ is the Euclidean norm. $\mathbf{p}_i \in \mathbb{R}^3$ and $\mathbf{v}_i \in \mathbb{R}^3$ are, respectively, the position and the control input of pursuer $i$. $\mathbf{p}_e \in \mathbb{R}^3$ and $\mathbf{v}_e \in \mathbb{R}^3$ are, respectively, the position and the control input of the faster evader. $\lambda_i = \frac{V_i}{V_e} < 1$ is the speed ratio of pursuer $i$ and the evader. Next, the occupied angle, dihedral angle and Rodrigues' rotation formula are presented.

## 2.2 Occupied Angle

A local time-varying coordinate system $\Gamma$ centered at the evader' time-varying global coordinate $\mathbf{p}_e = (x_e, y_e, z_e)^T$ in $\mathbb{R}^3$ is presented in Fig. 1. The global coordinate of pursuer $i$ is denoted by $\mathbf{p}_i = (x_i, y_i, z_i)^T$, $i = 1, \cdots, n$, and its corresponding coordinate in $\Gamma$ is denoted by $\mathbf{p}_{ie} = (x_{ie}, y_{ie}, z_{ie})^T = (x_i - x_e, y_i - y_e, z_i - z_e)^T$.

Suppose that both pursuer $i$ and the faster evader begin to move along fixed directions at time instant $t_1$. In the local coordinate system $\Gamma(t_1)$, if they meet at a point $\mathbf{m}$ in a finite time $t_1 + T$, which is denoted by $\mathbf{m} = (x_m, y_m, z_m)$, the point $\mathbf{m}$ must satisfy the following equation

$$\frac{\sqrt{(x_m - x_{ie})^2 + (y_m - y_{ie})^2 + (z_m - z_{ie})^2}}{\sqrt{x_m^2 + y_m^2 + z_m^2}} = \frac{V_i \cdot T}{V_e \cdot T} = \lambda_i. \tag{3}$$

Taking the square of both sides of (3), it yields

$$\left(x_m - \frac{x_{ie}}{1-\lambda_i^2}\right)^2 + \left(y_m - \frac{y_{ie}}{1-\lambda_i^2}\right)^2 + \left(z_m - \frac{z_{ie}}{1-\lambda_i^2}\right)^2 = \lambda_i^2 \frac{x_{ie}^2 + y_{ie}^2 + z_{ie}^2}{(1-\lambda_i^2)^2}. \tag{4}$$

The point $\mathbf{m}$ is on a sphere centered at $\mathbf{C}_i = (\frac{x_{ie}}{1-\lambda_i^2}, \frac{y_{ie}}{1-\lambda_i^2}, \frac{z_{ie}}{1-\lambda_i^2})^T$ with radius $R_i = \sqrt{x_{ie}^2 + y_{ie}^2 + z_{ie}^2} \frac{\lambda_i}{1-\lambda_i^2}$ shown in Fig. 1. The sphere (4) is inscribed in a right circular cone $\Lambda_i$ with the apex at $\mathbf{p}_e$. A great circle of a sphere is the intersection of the sphere and a plane that passes through the center point of the sphere. For any great circle $\mathbf{g}$ of the sphere (4), we can draw two tangent lines $L_1$ and $L_2$ that intersect the great circle $\mathbf{g}$ at $\mathbf{m}_1$ and $\mathbf{m}_2$, respectively. The occupied angle $\theta_i$ is defined as an angle between lines $L_1$ and $L_2$. Then, we obtain

$$\sin(\frac{\theta_i}{2}) = \frac{\mathbf{C}_i\mathbf{m}_1}{\mathbf{C}_i\mathbf{p}_e} = \lambda_i, \quad \theta_i = 2\sin^{-1}(\lambda_i). \tag{5}$$

Note that the occupied angle $\theta_i$ is actually the apex angle of the right circular cone $\Lambda_i$. If the faster evader moves along a direction within the right circular cone $\Lambda_i$ such as $L$ shown in Fig. 1, pursuer $i$ can always move in a corresponding direction to capture the evader at a point on the sphere (4). Hence, the space dominated by pursuer $i$ is within the right circular cone $\Lambda_i$, and the occupied angle $\theta_i$ is used to represent the dominated area by pursuer $i$.

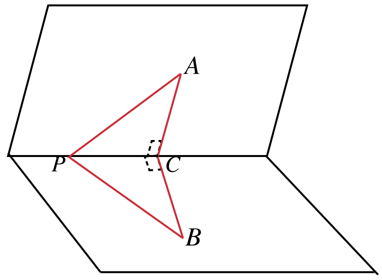


Fig. 2: The planes $\diamondsuit PAC$ and $\diamondsuit PBC$.

## 2.3 Dihedral Angle and Rodrigues' Rotation Formula

**Definition 1.** *A dihedral angle is an angle between two intersecting planes. For example, in Fig. 2, lines AC and BC are perpendicular to the line PC. $\angle ACB$ is a dihedral angle between intersecting planes $\diamondsuit PAC$ and $\diamondsuit PBC$.*

**Lemma 1.** *(**Property of Dihedral Angle**) For two planes $\diamondsuit PAC$ and $\diamondsuit PBC$ shown in Fig. 2, line PC is the intersection of two planes. Denote the angles by $\xi_a = \angle APB$, $\xi_b = \angle APC$, $\xi_c = \angle BPC$ and $\xi_d = \angle ACB$. The dihedral angle $\xi_d$ satisfies*

$$\cos\xi_a = \cos\xi_b \cos\xi_c + \sin\xi_b \sin\xi_c \cos\xi_d. \tag{6}$$

**Lemma 2.** *(**Rodrigues' Rotation Formula**) If $\mathbf{v}$ is a vector in $\mathbb{R}^3$ and $\mathbf{w}$ is a unit vector describing an axis of rotation about which $\mathbf{v}$ rotates by an angle $\alpha$ according to the right hand rule, the Rodrigues' rotation formula for the rotated vector $\mathbf{v}_{rot}$ is*

$$\mathbf{v}_{rot} = \mathbf{v}\cos\alpha + (\mathbf{w} \times \mathbf{v})\sin\alpha + \mathbf{w}(\mathbf{w} \cdot \mathbf{v})(1 - \cos\alpha), \tag{7}$$

*where $\times$ is the cross product.*

For two pursuers $i$ and $j$ shown in Fig. 3, the included angle $\alpha_{i,j}$ between two lines $\mathbf{p}_e\mathbf{p}_i$ and $\mathbf{p}_e\mathbf{p}_j$ is obtained by

$$\alpha_{i,j} = \arccos \frac{(\mathbf{p}_i - \mathbf{p}_e) \cdot (\mathbf{p}_j - \mathbf{p}_e)}{||\mathbf{p}_i - \mathbf{p}_e||_2 \cdot ||\mathbf{p}_j - \mathbf{p}_e||_2} \in [0, \pi]. \tag{8}$$

In this paper, two kinds of 3-D pursuit-evasion games with faster evader will be addressed.

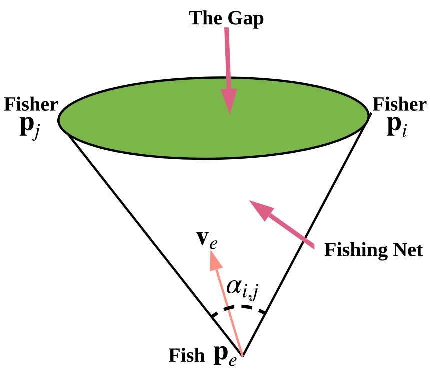


Fig. 3: Two-pursuer one-faster-evader fishing game.

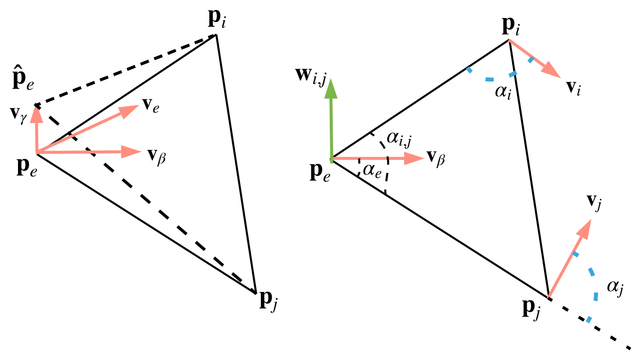


Fig. 4: The moving directions of the faster evader and two slower pursuers.

1) Two-pursuer one-faster-evader game in 3-D space. For two pursuers $\mathbf{p}_i, \mathbf{p}_j$ and one evader $\mathbf{p}_e$ shown in Fig. 3, the the evader is required to pass through the gap formed by $\mathbf{p}_i, \mathbf{p}_j$ and $\mathbf{p}_e$. If the included angle increases from $\alpha_{i,j} < \pi$ to $\alpha_{i,j} \geq \pi$ and $\min\{\|\mathbf{p}_i - \mathbf{p}_e\|_2, \|\mathbf{p}_j - \mathbf{p}_e\|_2\} > 0$, the evader wins; otherwise, the pursuers win.
2) Multi-pursuer one-faster-evader game in 3-D space. The faster evader is allowed to move freely. If there exists a moving direction along with the faster evader can escape successfully, the evader wins; otherwise, the pursuers win.

## 3 Two-pursuer One-faster-evader Game with Point Capture in 3-D Space

The two-pursuer one-faster-evader game is analogous to a natural scene that two fishers try to catch a fish with faster speed by a fishing net, while the fish tries to escape from the fishing net shown in Fig. 3. The two-pursuer one-faster-evader game represents some challenges encountered in control problems, such as cooperative besieging and capturing of multirobot and paths coordination of multiagent. In the two-pursuer one-faster-evader game, the evader aims to increase the included angle $\alpha_{i,j}$, but two pursuers try to decrease it.

### 3.1 Escape Algorithm

The velocity $\mathbf{v}_e$ of the evader can be decomposed into two velocities $\mathbf{v}_\beta$ and $\mathbf{v}_\gamma$, where $\mathbf{v}_\beta$ is on the plane $\diamondsuit\mathbf{p}_i\mathbf{p}_e\mathbf{p}_j$ and $\mathbf{v}_\gamma$ is perpendicular to the plane $\diamondsuit\mathbf{p}_i\mathbf{p}_e\mathbf{p}_j$ shown in the left figure of Fig. 4. First, we explore the relationship between $\mathbf{v}_\gamma$ and $\alpha_{i,j}$. Suppose the evader moves along a direction perpendicular to the plane $\diamondsuit\mathbf{p}_i\mathbf{p}_e\mathbf{p}_j$, i.e., $\mathbf{v}_\gamma = \mathbf{v}_e$ and $\mathbf{v}_\beta = 0$. The evader $\mathbf{p}_e$ gets to the position $\hat{\mathbf{p}}_e = \mathbf{p}_e + T\mathbf{v}_\gamma$ in a finite time interval $T$. Denote by $\alpha_{i,j} = \angle\mathbf{p}_i\mathbf{p}_e\mathbf{p}_j$, $\hat{\alpha}_{i,j} = \angle\mathbf{p}_i\hat{\mathbf{p}}_e\mathbf{p}_j$, $\xi_b = \angle\mathbf{p}_i\hat{\mathbf{p}}_e\mathbf{p}_e$, $\xi_c = \angle\mathbf{p}_j\hat{\mathbf{p}}_e\mathbf{p}_e$, $\mathbf{e}_b = \hat{\mathbf{p}}_e - \mathbf{p}_i$ and $\mathbf{e}_c = \hat{\mathbf{p}}_e - \mathbf{p}_j$. The dihedral angle between intersecting planes $\diamondsuit\mathbf{p}_i\hat{\mathbf{p}}_e\mathbf{p}_e$ and $\diamondsuit\mathbf{p}_j\hat{\mathbf{p}}_e\mathbf{p}_e$ equals $\alpha_{i,j}$. From **Lemma 1**, we obtain

$$\cos\hat{\alpha}_{i,j} = \cos\xi_b\cos\xi_c + \sin\xi_b\sin\xi_c\cos\alpha_{i,j}, \quad (9)$$

where

$$\begin{aligned} \cos\xi_b &= \frac{\mathbf{v}_\gamma\cdot\mathbf{e}_b}{||\mathbf{v}_\gamma||_2||\mathbf{e}_b||_2} = \frac{T||\mathbf{v}_\gamma||_2}{||\mathbf{e}_b||_2}, \\ \cos\xi_c &= \frac{\mathbf{v}_\gamma\cdot\mathbf{e}_c}{||\mathbf{v}_\gamma||_2||\mathbf{e}_c||_2} = \frac{T||\mathbf{v}_\gamma||_2}{||\mathbf{e}_c||_2}. \end{aligned} \quad (10)$$

Then, it yields

$$-\dot{\alpha}_{i,j}\sin\alpha_{i,j} = \frac{\partial\cos\alpha_{i,j}}{\partial T} = \lim_{T\to 0}\frac{\cos\hat{\alpha}_{i,j} - \cos\alpha_{i,j}}{T} = 0. \quad (11)$$

From (11), we get $\dot{\alpha}_{i,j} = 0$ if $\sin\alpha_{i,j} \neq 0$. When $\sin\alpha_{i,j} = 0$, $\alpha_{i,j} = 0, \pi$. Note that $\alpha_{i,j} = \pi$ means that the evader wins the game. When $\alpha_{i,j} = 0$, we have $\dot{\alpha}_{i,j} = \lim_{T\to 0}\frac{\hat{\alpha}_{i,j} - \alpha_{i,j}}{T} = 0$. Hence, if the pursuit-evasion game is not ended and the evader moves along the direction $\mathbf{v}_\gamma$, we have $\dot{\alpha}_{i,j} = 0$, i.e., $\mathbf{v}_\gamma$ will not change the included angle $\alpha_{i,j}$.

Next, we explore the relationship between $\mathbf{v}_\beta$ and $\alpha_{i,j}$. Denoted $d_i = ||\mathbf{p}_i - \mathbf{p}_e||_2$ and $d_j = ||\mathbf{p}_j - \mathbf{p}_e||_2$ as the distances between the faster evader and two pursuers shown in the right figure of Fig. 4. $\alpha_e$ is the relative heading of $\mathbf{v}_\beta$ from the vector $\overrightarrow{\mathbf{p}_e\mathbf{p}_j}$. Denote $\mathbf{v}_i, \mathbf{v}_j$ as the velocities of two pursuers $\mathbf{p}_i, \mathbf{p}_j$. If $\mathbf{v}_i = 0, \mathbf{v}_j = 0$, we have

$$\dot{\alpha}_{i,j} = \frac{||\mathbf{v}_\beta||_2\sin(\alpha_{i,j} - \alpha_e)}{d_i} + \frac{||\mathbf{v}_\beta||_2\sin\alpha_e}{d_j}. \quad (12)$$

To maximize $\dot{\alpha}_{i,j}$, the optimal relative heading $\alpha_e^*$ is given by

$$\alpha_e^* = \arctan\frac{d_i - d_j\cos\alpha_{i,j}}{d_j\sin\alpha_{i,j}}. \quad (13)$$

Note that $\dot{\alpha}_{i,j}$ is also related to $\|\mathbf{v}_\beta\|_2$. To maximize $\dot{\alpha}_{i,j}$ (12), we have $\mathbf{v}_\beta = \mathbf{v}_e$ and $\mathbf{v}_\gamma = 0$. Then, from **Lemma 2**, we have

$$\mathbf{v}_e = V_e(\mathbf{c}_j\cos\alpha_e^* + \mathbf{w}_{i,j}\times\mathbf{c}_j\sin\alpha_e^*), \quad (14)$$

where $\mathbf{w}_{i,j} = \frac{(\mathbf{p}_j - \mathbf{p}_e)\times(\mathbf{p}_i - \mathbf{p}_e)}{||(\mathbf{p}_j - \mathbf{p}_e)\times(\mathbf{p}_i - \mathbf{p}_e)||_2}$ and $\mathbf{c}_j = \frac{\mathbf{p}_j - \mathbf{p}_e}{||\mathbf{p}_j - \mathbf{p}_e||_2}$.

### 3.2 Pursuit Algorithm

Similarly, we can know that the optimal moving directions $\mathbf{v}_i, \mathbf{v}_j$ of two pursuers for decreasing the included angle $\alpha_{i,j}$ are also on the plane $\diamondsuit\mathbf{p}_i\mathbf{p}_e\mathbf{p}_j$ shown in the right figure of Fig. 4. Denote $\alpha_i$ and $\alpha_j$ as the relative headings of $\mathbf{p}_i$ and $\mathbf{p}_j$ from the vectors $\overrightarrow{\mathbf{p}_i\mathbf{p}_e}$ and $\overrightarrow{\mathbf{p}_e\mathbf{p}_j}$. If $\mathbf{v}_e = 0$, we have

$$\dot{\alpha}_{i,j} = -\frac{V_i}{d_i}\sin\alpha_i - \frac{V_j}{d_j}\sin\alpha_j. \quad (15)$$

To minimize $\dot{\alpha}_{i,j}$, the optimal relative headings $\alpha_i^*$ and $\alpha_j^*$ are given by

$$\alpha_i^* = \frac{\pi}{2}, \quad \alpha_j^* = \frac{\pi}{2}. \quad (16)$$

Then, from **Lemma 2**, we have

$$\mathbf{v}_i = -V_i \mathbf{w}_{i,j} \times \mathbf{c}_i, \tag{17}$$

$$\mathbf{v}_j = V_j \mathbf{w}_{i,j} \times \mathbf{c}_j, \tag{18}$$

where $\mathbf{c}_i = \frac{\mathbf{p}_i - \mathbf{p}_e}{||\mathbf{p}_i - \mathbf{p}_e||_2}$.

### 3.3 Sufficient Escape Condition

From Section 2.2, we can know that each pursuer can occupy a right circular cone. If the evader moves along a direction that is not within the right circular cone of pursuer $\mathbf{p}_i$ or $\mathbf{p}_j$, the evader will escape successfully. Hence, the evader must win the game if the included angle $\alpha_{i,j}$ satisfies

$$\alpha_{i,j} > \frac{\theta_i + \theta_j}{2}, \tag{19}$$

where $\theta_i$ and $\theta_j$ are occupied angles of pursuer $i$ and pursuers $j$, respectively.

### 3.4 Sufficient Capture Condition

Denote $d_{i,j}$ as the relative distance between pursuer $\mathbf{p}_i$ and $\mathbf{p}_j$. Under velocity control laws (14), (17), (18) and combining (12) and (15), we have

$$\dot{\alpha}_{i,j} = -\frac{V_i}{d_i} - \frac{V_j}{d_j} + \frac{V_e d_{i,j}}{d_i d_j}. \tag{20}$$

Hence, the pursuers must win the game if $\dot{\alpha}_{i,j} \leq 0$, i.e.,

$$d_{i,j} \leq d_j \lambda_i + d_i \lambda_j, \tag{21}$$

where $\lambda_i = \frac{V_i}{V_e}$ and $\lambda_j = \frac{V_j}{V_e}$ are speed ratios. If the condition (21) holds at time instant $t_0$, we have $\dot{\alpha}_{i,j}(t) < 0, t \geq t_0$. The proof follows from that of **Theorem 4.1** in the work [11].

## 4 Multi-pursuer one-faster-evader game with Non-Zero capture radius in 3-D space

It is proved that the faster free-moving evader can avoid point capture [19], but it may be captured by a group of slower pursuers with non-zero capture radius $d_c$.

**Definition 2.** *The evader is said to be captured by a pursuer with non-zero capture radius $d_c > 0$ if the distance between the evader and the pursuer is less than $d_c$.*

**Assumption 1.** *Each pursuer can access the position information of the faster evader, itself and its neighbors. The evader can access the position information of itself and its neighbors. The position information of each player can be obtained by the range-based or vision-based localization methods [20–22]. The moving directions of the evader and pursuers are unknown to each other.*

For a group of $n$ pursuers, each pursuer has a non-zero capture radius $d_c$ and the faster evader can access the relative position of $m \leq n$ pursuers. Denote $\mathbf{p}_{ie} = \mathbf{p}_i - \mathbf{p}_e, i = 1, \cdots, m$ as the relative position between pursuer $i$ and evader. Define function $\chi_i \in \mathbb{R}$ as

$$\chi_i = \arccos \frac{\mathbf{v}_e^T \cdot \mathbf{p}_{ie}}{||\mathbf{v}_e||_2 \cdot ||\mathbf{p}_{ie}||_2} - \arcsin \frac{d_c}{||\mathbf{p}_{ie}||_2}, \tag{22}$$

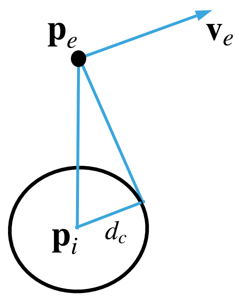


Fig. 5: The non-zero capture radius $d_c$.

where $\mathbf{v}_e$ is the velocity of the evader shown in Fig. 5. From Section 2.2, we can know that pursuer $i$ cannot capture the evader if $\chi_i > \frac{\theta_i}{2}$. Define escape function $\digamma(\mathbf{v}_e) \in \mathbb{R}^m$ as

$$\digamma(\mathbf{v}_e) = (\chi_1 - \frac{\theta_1}{2}, \chi_2 - \frac{\theta_2}{2}, \cdots, \chi_m - \frac{\theta_m}{2})^T, \tag{23}$$

where $\theta_i, i = 1, \cdots, n$ is the occupied angle of pursuer $i$.

### 4.1 Sufficient Escape Condition

The faster evader must win the game if it can access the relative position of all pursuers and there exists a moving direction $\mathbf{v}_e$ such that

$$\digamma(\mathbf{v}_e) > \mathbf{0}. \tag{24}$$

$\digamma(\mathbf{v}_e) > \mathbf{0}$ means that $\chi_i > \frac{\theta_i}{2}, i = 1, \cdots, n$, i.e., no pursuer can capture the faster evader.

### 4.2 Escape Algorithm

The evader should try to increase $\digamma(\mathbf{v}_e)$. Hence, the velocity control law $\mathbf{v}_e$ of the faster evader is given by

$$\mathbf{v}_e = \arg \max_{||\hat{\mathbf{v}}_e||_2 = V_e} \digamma(\hat{\mathbf{v}}_e). \tag{25}$$

Particle swarm optimization (PSO) algorithm [23] is used to solve (25). PSO algorithm is initialized with a group of $n_e$ particles. Denote $\beta_i$ and $\varphi_i$ as the position and velocity of particle $i$. At $k$th iteration, velocity and position of particle $i$ are updated by

$$\varphi_i(k+1) = w\varphi_i(k) + \zeta_1\eta_1(\phi_i(k) - \beta_i(k)) + \zeta_2\eta_2(\phi_g(k) - \beta_i(k)), \tag{26}$$

$$\beta_i(k+1) = \beta_i(k) + \varphi_i(k+1), i = 1, \cdots, n_e, \tag{27}$$

where $w$ is the weight. $\zeta_1, \zeta_2$ are learning factors. $\eta_1, \eta_2$ are uniformly distributed random numbers between $(0, 1)$. $\phi_i(k)$ is the fittest solution achieved by particle $i$ before the $k+1$ iteration. $\phi_g(k)$ is the fittest solution achieved by all particles before the $(k+1)$th iteration. The cost function of particle $i$ is designed as

$$J_i(k+1) = \max \digamma(\beta_i(k+1)). \tag{28}$$

The optimization process (26)-(27) continues until the maximum iteration number is reached. Denote $m_e$ as the maximum iteration number. Based on $\phi_g(m_e)$, the velocity $\mathbf{v}_e$ is designed as

$$\mathbf{v}_e = V_e \frac{\phi_g(m_e)}{||\phi_g(m_e)||_2}. \tag{29}$$

### 4.3 Pursuit Algorithm

The group of pursuers should not only try to trap the evader within the capture domain, but also approach it. From Section 2.2, we can know that each pursuer dominates a right circular cone in $\mathbb{R}^3$. For two pursuers $\mathbf{p}_i$ and $\mathbf{p}_j$, their corresponding right circular cones with the same apex $\mathbf{p}_e$ intersect if the included angle $\alpha_{i,j}$ (8) satisfies

$$\alpha_{i,j} < \frac{\theta_i + \theta_j}{2}, \tag{30}$$

where $\theta_i$ and $\theta_j$ are occupied angles of pursuer $i$ and pursuer $j$, respectively. To occupy more space, pursuer $i$ and pursuer $j$ should try to increase the include angle $\alpha_{i,j}$. From section 3.2, we can know that the optimal velocity control law of pursuer $i$ for increasing the included angle $\alpha_{i,j}$ is

$$\mathbf{v}_i = V_i \mathbf{w}_{i,j} \times \mathbf{c}_i, \tag{31}$$

where $\mathbf{w}_{i,j} = \frac{(\mathbf{p}_j - \mathbf{p}_e)\times(\mathbf{p}_i - \mathbf{p}_e)}{||(\mathbf{p}_j - \mathbf{p}_e)\times(\mathbf{p}_i - \mathbf{p}_e)||_2}$ and $\mathbf{c}_i = \frac{\mathbf{p}_i - \mathbf{p}_e}{||\mathbf{p}_i - \mathbf{p}_e||_2}$.

Each pursuer $i$ may have many neighboring pursuers. $\Omega_i$ is denoted as the set of neighboring pursuers of pursuer $i$. To occupy more space, the cooperative surrounding algorithm $\mathbf{v}_{is}$ of pursuer $i$ is designed as

$$\mathbf{v}_{is} = k_i \frac{\sum_{j\in\Omega_i} \alpha_{i,j} \cdot \mathbf{v}_{i,j}}{||\sum_{j\in\Omega_i} \alpha_{i,j} \cdot \mathbf{v}_{i,j}||_2}, \tag{32}$$

where $k_i \in [0, V_i]$ is a surrounding coefficient. $\mathbf{v}_{i,j}$ is a unit vector for increasing the included angle $\alpha_{i,j}$

$$\mathbf{v}_{i,j} = \begin{cases} \mathbf{w}_{i,j} \times \mathbf{c}_i, & \text{if } \alpha_{i,j} \in [0, \frac{\theta_i+\theta_j}{2}), \\ 0, & \text{if } \alpha_{i,j} \geq \frac{\theta_i+\theta_j}{2}. \end{cases} \tag{33}$$

Since $\mathbf{v}_{i,j}$ is a unit vector for increasing the included angle $\alpha_{i,j}$, $\alpha_{i,j}\mathbf{v}_{i,j}$ in (32) can be regarded as a 'repulsive force' between pursuer $i$ and pursuer $j$. Then, $\sum_{j\in\Omega_i} \alpha_{i,j} \cdot \mathbf{v}_{i,j}$ is the 'resultant of repulsive forces'. To approach the faster evader, the hunting algorithm $\mathbf{v}_{ih}$ is designed as

$$\mathbf{v}_{ih} = -h_i \mathbf{c}_i, \tag{34}$$

where $h_i \in [0, V_i]$ is a hunting coefficient. Combining (32) and (34), the pursuit algorithm $\mathbf{v}_i$ for pursuer $i$ is designed as

$$\mathbf{v}_i = \mathbf{v}_{is} + \mathbf{v}_{ih} = k_i \frac{\sum_{j\in\Omega_i} \alpha_{i,j} \cdot \mathbf{v}_{i,j}}{||\sum_{j\in\Omega_i} \alpha_{i,j} \cdot \mathbf{v}_{i,j}||_2} - h_i \mathbf{c}_i. \tag{35}$$

Since $\mathbf{v}_{is} \cdot \mathbf{v}_{ih} = 0$, $\mathbf{v}_{is}$ is perpendicular to the $\mathbf{v}_{ih}$. i.e., $\|\mathbf{v}_i\|_2 = k_i^2 + h_i^2 = V_i$. Note that there is a trade-off between cooperative surrounding and hunting, i.e., how to choose the values of $k_i$ and $h_i$. Denote $\psi_i = \arctan \frac{k_i}{h_i}$ as the trade-off coefficient. If a pursuer has larger maximum speed than other pursuers, it should focus on the hunting because it has a better chance to capture the evader. Denote $V_{\max} = \max\{V_1, \cdots, V_n\}$ as the maximum speed among a group of $n$ pursuers. Hence, the parameters $k_i$ and $h_i$ are designed as

$$k_i = V_i \sin\psi_i, \quad h_i = V_i \cos\psi_i, \quad \psi_i = \frac{\pi}{2}(1 - \frac{V_i}{V_{\max}}). \tag{36}$$

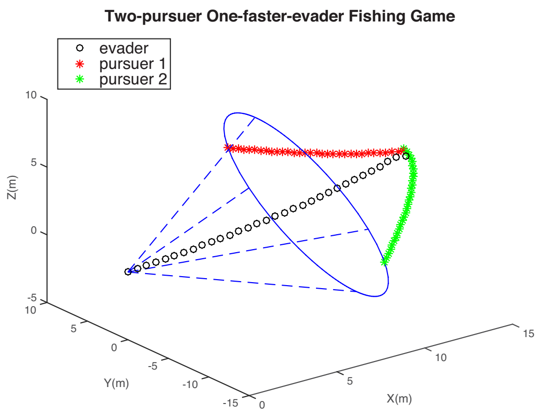


Fig. 6: Two-pursuer one-faster-evader fishing game.

## 5 Simulation

Two simulation examples are presented to illustrate the theoretical results. In the two-pursuer one-faster-evader fishing game shown in Fig. 6, $\mathbf{p}_i, \mathbf{p}_j$ try to decrease the included angle $\alpha_{i,j}$ under optimal pursuit control laws (17) and (18), while the faster evader $\mathbf{p}_e$ aims to increase the included angle $\alpha_{i,j}$ under optimal escape control law (14). The speed ratios of two slower pursuers and the faster evader are set as $\lambda_i = \frac{V_i}{V_e} = 0.65$ and $\lambda_j = \frac{V_j}{V_e} = 0.7$, respectively. The initial positions of two pursuers are set as $\mathbf{p}_i = (8, 5, 5)$ and $\mathbf{p}_j = (10, -10, 0)$, and the initial position of the faster evader is set as $\mathbf{p}_e = (0, 0, 0)$. Since this initial spatial distribution satisfies the sufficient capture condition (21), the evader will be finally captured by the pursuers. In the ten-pursuer one-faster-evader game shown in Fig. 7, the faster evader is assumed to obtain at most eight closer pursuers' relative positions $m = 8$. The speed ratio of the pursuer and evader is set as $\lambda_i = \frac{V_i}{V_e} = 0.8, i = 1, \cdots, 10$. In this example, the escape function in (23) satisfies $F(\mathbf{v}_e)(t) < 0, t \in (0, 38]$, which indicates that the sufficient escape condition (24) cannot be satisfied. The initial positions of ten pursuers are set as $\mathbf{p}_1 = (5, -6, -5)$, $\mathbf{p}_2 = (6, 4, -6)$, $\mathbf{p}_3 = (-5, -5, -4)$, $\mathbf{p}_4 = (-4, 5, -6)$, $\mathbf{p}_5 = (5, -5, 5)$, $\mathbf{p}_6 = (6, 4, 4)$, $\mathbf{p}_7 = (-5, -5, 5)$, $\mathbf{p}_8 = (-6, 6, 4)$, $\mathbf{p}_9 = (8, 8, 8)$, $\mathbf{p}_{10} = (-8, -8, -8)$. The initial position of the faster evader is set as $\mathbf{p}_e = (0, 0, 0)$. The faster evader is finally captured by a pursuer (the winner) at time instant $t = 38$.

## 6 Conclusion

This work proposes a pursuit algorithm and an escape algorithm for slower pursuers and faster evader to maximize their chance of success in two kinds of pursuit-evasion games in $\mathbb{R}^3$. The corresponding capture conditions and escape conditions are derived. The future work will be on the capture conditions of the multi-pursuer one-evader game.

## Acknowledgement

The authors would like to thank Dr. Chen Wang (`https://chenwang.site`) for refining the algorithm.

## References

[1] C. Wang, J. Yang, L. Xie, and J. Yuan, "Kervolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 31–40.

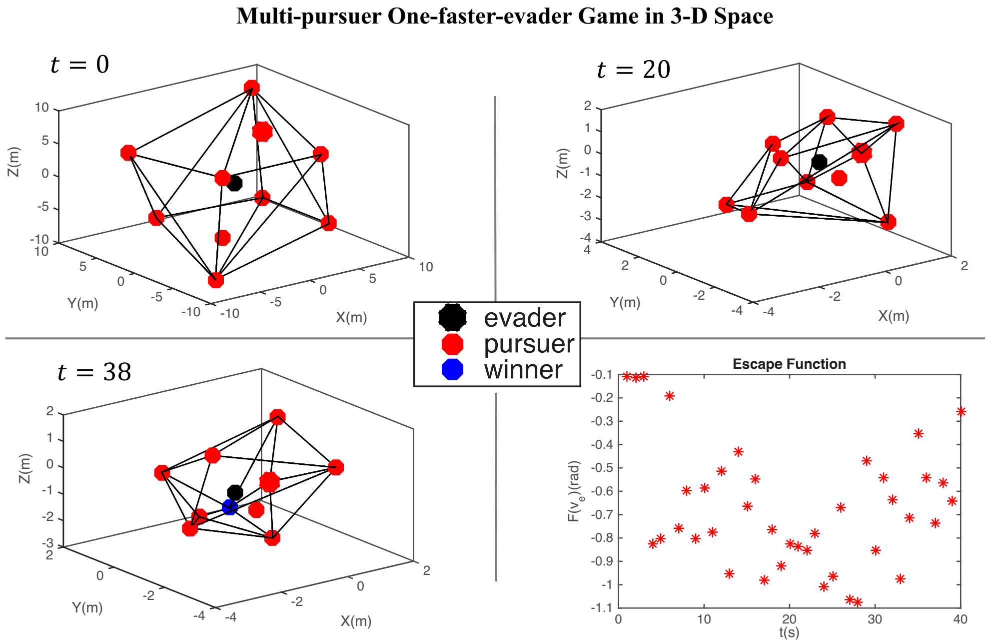


Fig. 7: Multi-pursuer one-faster-evader game.


[2] C. Wang, L. Zhang, L. Xie, and J. Yuan, “Kernel cross-correlator,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018, pp. 4179–4186.

[3] R. Isaacs, *Differential Games*. New York, USA:Wiley, 1965.

[4] Z. Zhou, R. Takei, H. Huang, and C. J. Tomlin, “A general, open-loop formulation for reach-avoid games,” in *International Conference on Decision and Control*, 2014, pp. 6501–6506.

[5] M. Chen, Z. Zhou, and C. J. Tomlin, “Multiplayer reach-avoid games via low dimensional solutions and maximum matching,” in *American Control Conference*, 2014, pp. 1444–1449.

[6] R. Takei, H. Huang, D. Jerry, and T. Claire J, “Time-optimal multi-stage motion planning with guaranteed collision avoidance via an open-loop game formulation,” in *IEEE International Conference on Robotics and Automation*, 2012, pp. 323–329.

[7] B. Shaunak D, B. Francesco, and H. Joao P, “A cooperative Homicidal Chauffeur game,” *Automatica*, vol. 45, no. 7, pp. 1771–1777, 2009.

[8] A. B. Oller and A. Yildiz, “Foraging swarms as Nash Equilibria of dynamic games,” *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 979–987, 2013.

[9] H. Zhang, L. Cui, and Y. Luo, “Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network ADP,” *IEEE Transactions on Cybernetics*, vol. 43, no. 1, pp. 206–216, Dec. 2012.

[10] D. Zhao, Q. Zhang, D. Wang, and Y. Zhu, “Experience replay for optimal control of nonzero-sum game systems with unknown dynamics,” *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 854–865, Jan. 2016.

[11] W. Zha, J. Chen, Z. Peng, and D. Gu, “Construction of barrier in a fishing game with point capture,” *IEEE Transactions on Cybernetics*, vol. 47, no. 6, pp. 1409–1422, 2016.

[12] C. Wang, Z. Ting, W. Kai, L. Shuaiyi, and M. Hongbin, “A new approach of multi-robot cooperative pursuit,” in *Proceedings of the 32nd Chinese Control Conference*, 2013, pp. 7252–7256.

[13] X. Li, Z. Peng, W. Zha, and J. Chen, “Construction of barrier in a three-player pursuit-evasion game,” *Unmanned Systems*, vol. 4, no. 01, pp. 41–49, 2016.

[14] J. Chen, W. Zha, Z. Peng, and D. Gu, “Multi-player pursuit–evasion games with one superior evader,” *Automatica*, vol. 71, pp. 24–32, Sep. 2016.

[15] H. Wang, Q. Yue, and J. Liu, “Research on pursuit-evasion games with multiple heterogeneous pursuers and a high speed evader,” in *27th Chinese Control and Decision Conference*, 2015, pp. 4366–4370.

[16] X. Fang, C. Wang, L. Xie, and J. Chen, “Cooperative pursuit with multi-pursuer and one faster free-moving evader,” *IEEE Transactions on Cybernetics, Accepted*, 2020.

[17] S. Rodriguez, J. Denny, J. Burgos, A. Mahadevan, K. Manavi, L. Murray, A. Kodochygov, T. Zourntos, and N. M. Amato, “Toward realistic pursuit-evasion using a roadmap-based approach,” in *IEEE International Conference on Robotics and Automation*, 2011, pp. 1738–1745.

[18] M. Janosov, C. Virágh, G. Vásárhelyi, and T. Vicsek, “Group chasing tactics: how to catch a faster prey,” *New Journal of Physics*, vol. 19, no. 5, p. 053003, 2017.

[19] F. Chernous’ ko, “A problem of evasion from many pursuers: Pmm,” *Journal of Applied Mathematics and Mechanics*, vol. 40, no. 1, pp. 11–20, 1976.

[20] C. Wang, H. Zhang, T.-M. Nguyen, and L. Xie, “Ultra-wideband aided fast localization and mapping system,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 1602–1609.

[21] X. Fang, C. Wang, T.-M. Nguyen, and L. Xie, “Graph optimization approach to range-based localization,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems, Accepted*, 2020.

[22] C. Wang, J. Yuan, and L. Xie, “Non-iterative SLAM,” in *18th International Conference on Advanced Robotics (ICAR)*, 2017, pp. 83–90.

[23] J. Kennedy, “Particle swarm optimization,” *Encyclopedia of machine learning*, pp. 760–766, 2010.