Bearing Angle Measurement Based Cooperative Pursuit-Evasion Game in Non-convex Environments

Di Guo *, Gangfeng Yan *, Zhiyun Lin *
* Asus Intelligent Systems Lab
College of Electrical Engineering
Zhejiang University, 38 Zheda Road, Hangzhou, 310027 P. R. China
Email: emaiguo@gmail.com (Guo), ygf@zju.edu.cn (Yan), linz@zju.edu.cn (Lin)

Abstract—This paper addresses a cooperative pursuit-evasion game in non-convex environments, for which a team of pursuers are equipped with onboard bearing angle measurement sensors. Different from many existing work, our work assumes that agents have no global coordinate system and thus can not measure the absolute positions, but it is expected for the pursuers to estimate the positions of other agents in their local frames. Based on the estimated positions, a distributed computation method is developed where a series of Hamilton-Jacobi-Bellman (HJB) equations are solved cooperatively in order to determine the control input for every pursuer with the help of limited information exchange among pursuers. We demonstrate the effectiveness of the proposed scheme by simulations.

Keywords: Pursuit-evasion game, nonlinear measurement, particle filter, localization

I. INTRODUCTION

The solutions of pursuit-evasion games are useful in many control systems especially in security and military applications. This paper deals with a distributed pursuit-evasion problem for a team of pursuers to cooperatively capture a single evader in a non-convex domain.

In recent years various pursuit-evasion games have been explored by many researchers depending on sensing and motion capabilities, the tasks of pursuers, the working spaces, etc. In [1] a pursuit-evasion game is investigated with two fixed-wing autonomous aircrafts using supervisory control where the aircrafts can switch between pursuer and evader. The visibility-based pursuit-evasion game is studied in [2] where the evader is captured when it falls into the visibility region of a pursuer and combinatorial filters are designed for tracking targets. Nonzero-sum pursuit-evasion games are investigated in [3], [4]. Scenarios that players have limited information of others are considered in [3] while the work in [4] focuses on an acoustic field. In [5] target tracking with adaptive sampling is investigated in the framework of pursuit-evasion game with the purpose of optimizing the use of expensive and limited resources that autonomous vehicles have. In [6] a mixed integer linear programming and receding horizon control are applied to seek the solutions of multi pursuers problems in polygonal environments. A pursuit-evasion problem is addressed in [7] in which a team of searchers equipped with limited range sensors clear any evader from an unknown, multiply-connected planar environment. A modular hybrid system architecture is implemented in [8] for autonomous robots to pursue an evader while concurrently building a global map in an unknown environment. In [9] a cost function is designed to deal with a single-pursuer, two-evader differential game where the pursuer attempts to capture either of the evaders while minimizing its cost. A pursuit-evasion game is studied in [10] in which the pursuer tries to stop the evader from reaching a harbor. A solvable max-min problem is formulated whose solution can be used to define a receding horizon feedback control law.

Our work has been inspired by research in [11]–[13]. In [11] Voronoi partition of the game domain is introduced where the evader’s Voronoi cell is safe for him to arrive if all agents have the same maximum speed. A decentralized control scheme is employed by pursuers in order to jointly minimize the area of the evader’s Voronoi cell. The capture is guaranteed regardless of the evader’s actions. An exit is added in [12] with a similar strategy employed which also has capture guarantee. The non-convex domain is taken into consideration in [13] where Voronoi partition is not useful anymore. Thus, the idea of safe set (similar to Voronoi cell, which is safe for the evader to arrive) is introduced where the control scheme is to minimize the safe area of an evader. However, there is no guarantee of capture in this case. The accurate global positions of all agents are required to centrally compute each pursuer’s control input.

The paper extends the cooperative pursuit-evasion games in non-convex environments. The contribution is twofold. First, no global positioning device like GPS is required. The positions of all agents are estimated by each pursuer using particle filtering and angle localization technique. The only information pursuers can use is the angle measurements of other agents in local moving frames. The system model and measurement model noises are also included which make it more practical. We point out that our scheme is also suitable for other measurement models, e.g., acoustic sensing [4], range sensing [7] etc. Second, the computation of control inputs is distributed among all pursuers resulting the cooperation effort of pursuers. Both localization and safe set estimation are evaluated in simulations. Although there is lack of analytical capture guarantee, simulation results have demonstrated the effectiveness of our scheme.
This paper is organized as follows. Section II provides the system model and measurement model and formulates the pursuit-evasion problem. Localization methods are presented in Section III. A pursuit strategy is developed in Section IV on the basis of safe set computation. Simulation results are provided in Section V, which is followed by conclusions in Section VI.

II. SYSTEM SETUP AND PROBLEM FORMULATION

In this section the dynamic model of the participating agents and a nonlinear measurement model are introduced, which are followed by the statement of the problem studied in this paper.

A. System Setup

In this paper, we study the pursuit-evasion game involving $N$ pursuers and a single evader taking place in a non-convex domain $\Omega$ in $\mathbb{R}^2$ as depicted in Fig. 1 where the blue rectangle is an obstacle. Let $x_i \in \mathbb{R}^2$ and $x_e \in \mathbb{R}^2$ be the positions of pursuer $i$ and the evader in a global coordinate system, respectively. The dynamics of the agents are as follows:

\[
\begin{align*}
\dot{x}_i(t) &= u_i(t) + w_i(t), \quad u_i(t) \in U_i, \quad i = 1, \ldots, N \\
\dot{x}_e(t) &= u_e(t) + w_e(t), \quad u_e(t) \in U_e
\end{align*}
\]

where $u_i$ and $u_e$ are the control inputs of pursuer $i$ and the evader, respectively, $U_i$ and $U_e$ are the sets of control values, and $w_i(t)$ and $w_e(t)$ are system model noises. Here, let $U_i = \{u_i \in \mathbb{R}^2 \mid \|u_i\| \leq v_{i,\text{max}}\}$ and $U_e = \{u_e \in \mathbb{R}^2 \mid \|u_e\| \leq v_{e,\text{max}}\}$, i.e., there is a maximum speed of each agent.

Consider a sampling period $T > 0$. Then the corresponding sampled-data system can be approximated by

\[
x_i(k+1) = x_i(k) + Tu_i(k) + Tw_i(k), \quad i = 1, \ldots, N, e.
\]

As depicted in Fig. 1, the measurements taken by agent $i$ are the angles of all the other agents, namely,

\[
\theta_{i,j}(k) = \arg \left( x_j(k) - x_i(k) \right) + v_i(k)
\]

where $j = 1, \ldots, i-1, i+1, \ldots, N, e, v_i$ is the measurement noise, and the function $\arg(\cdot) : \mathbb{R}^2 \to (-\pi, \pi]$ returns the angle of the vector.

B. Problem Formulation

The goal of the pursuers is to capture the evader cooperatively, i.e., at time step $k$, there exists a pursuer, say $i$, such that $\|x_i(k) - x_e(k)\| \leq d_c$ where $d_c$ is called the capture distance.

To achieve this goal, each pursuer selects control inputs using a pursuit strategy $\mu(\Theta)$ where $\Theta$ denotes the measurement vector which can be taken by itself and received from other pursuers.

In our problem setup, no global coordinate system for the pursuers is required. Every pursuer uses the measurements to localize other agents. Then the control inputs are determined using certain strategies. System model noises and measurement noises are included so that the developed strategy is more suitable for practical applications.

III. LOCALIZATION

In this section the localization algorithms of other agents in the moving frame of each pursuer are presented. First, it estimates other pursuers using a particle filter (PF). Then, the evader is localized using an angle localization method.

A. Estimate other Pursuers

Each pursuer has a common direction information, e.g., north. Denote by $r_i = [r_{i,1}, \ldots, r_{i,i-1}, r_{i,i+1}, \ldots, r_{i,N}]^T$ the aggregated relative positions of all the other pursuers in pursuer $i$’s moving frame where

\[
r_{i,j} = x_j - x_i
\]

is the relative position of pursuer $j$. We let pursuer $i$’s position be the origin point in its own frame.

We can write the dynamic of the aggregated system as follows,

\[
r_i(k+1) = r_i(k) + Tu_i(k) + Tw_i(k)
\]

where

\[
\begin{align*}
&u_i(k) = [u_1(k) - u_i(k) \ldots u_{i-1}(k) - u(k)] \\
&u_{i+1}(k) - u_i(k) \ldots u_N - u_i(k)]^T, \\
&w_i(k) = [w_1(k) - w_i(k) \ldots w_{i-1}(k) - w_i(k)] \\
&w_{i+1}(k) - w_i(k) \ldots w_N - w_i(k)]^T.
\end{align*}
\]

The control inputs can be exchanged through communications in real time.

The measurement vector $\Theta_i(k)$ contains two parts: the angles of all other pursuer $j$’s in pursuer $i$’s frame, i.e.,

\[
\theta_{i,j}(k) = \arg (r_{i,j}(k)) + v_i(k), j \neq i
\]

and the angles of all other pursuer $j$’s in all other pursuer $m$’s frame, i.e.,

\[
\theta_{m,j}(k) = \arg (r_{i,j}(k) - r_{i,m}) + v_m(k), j \neq i, j \neq m, i \neq m
which can be received from pursuer \( m \). So we can write the measurement equation as follows,

\[
\Theta_i(k) = h_i(r_i(k), v(k))
\]

where \( v(k) = [v_1(k), \ldots, v_N(k)]^T \). A particle filter is more applicable to nonlinear measurements. We consider a particle filter for better performance in dealing with the system model noise and measurement noise here. The detailed algorithm is given below.

**Algorithm 1** Localize all the other pursuers by pursuer \( i \)

**Input:** System and measurement model, control inputs, measurements \( \Theta_i \)

**Output:** The estimate state \( \hat{r}_i(k) \)

1. Randomly generate \( M \) initial particles according to initial estimate

\[
X^+_j(0), \; j = 1, \ldots, M
\]

2. **for** \( k = 1, 2, \ldots \) **do**
3. Perform the time propagation step to obtain a priori particles,

\[
X^-_j(k) = X^+_j(k-1) + Tw_i^r(k-1) + Tw_i^v(k-1)
\]
4. Compute the relative likelihoods on the basis of the measurement equation

\[
q_j(k) = P(\Theta_i(k)|X^-_j(k))
\]
5. Scale the relative likelihoods

\[
q_j(k) = \frac{q_j(k)}{\sum_{h=1}^{M} q_h(k)}
\]
6. Generate a set of a posteriori particles \( X^+_j(k) \) on the basis of the relative likelihoods \( q_j(k) \)
7. Decide the estimate \( \hat{r}_i(k) = \frac{1}{M} \sum_{j=1}^{M} X^+_j(k) \)
8. **end for**

**B. Localize the evader**

As long as pursuer \( i \) obtains the estimation of all the other pursuers, it can localize the evader using an angle based optimization method. As depicted in Fig. 2, pursuer \( i \) solves the following optimization problem in order to obtain an estimate \( \hat{r}_{i,e} \).

\[
\min_{\hat{r}_{i,e}} \left( \text{arg}(\hat{r}_{i,e} - \theta_{i,e})^2 + \sum_{j \neq i} (\text{arg}(\hat{r}_{i,e} - \hat{r}_{i,j} - \theta_{j,e})^2, \right)
\]

where \( \theta_{j,e} \) is transmitted by pursuer \( j \) to pursuer \( i \).

**IV. PURSUIT STRATEGY**

In this section we first introduce the notion of safe set of the evader which is developed in [13] and its computation method. Based on that, the pursuit strategy is employed to minimize the area of the safe set in each step [13]. Then, we develop a distributed method for pursuers to cooperatively determine their own control inputs.

**A. Safe Set of the Evader**

A point \( y \in \Omega \) is said to be safe-reachable for the evader if there exists \( t > 0 \) and some control input sequence \( u_e(s), s < t \) such that \( x_e(t) = y \) and \( \|x_e(s) - x_i(s)\| > d_e \) for all \( s \in [0, t] \), \( u_i(t) \in U_i \), and \( i = \{1, \ldots, N\} \). The safe set of the evader is then defined as

\[
\mathcal{S}_e = \{ y \in \Omega | y \text{ is safe-reachable} \}
\]

The safe set can be found in the following way.

Define the minimum time-to-reach function \( \psi_i : \Omega \rightarrow \mathbb{R} \) for pursuer \( i \) as

\[
\psi_i(y) = \min \left\{ t \left| x_i(0) = x^0_i, x_i(t) = y, x_i(s) \in \Omega, \forall s \in [0, t] \right. \right\}
\]

where \( x^0_i \) denotes the initial position of pursuer \( i \).

The minimum time-to-reach function is the viscosity solution to the following Hamilton-Jacobi-Bellman (HJB) equation [14],

\[
- \min_{u_i \in U_i} \{ \nabla \psi_i(y) \cdot u_i \} = 1 \tag{1}
\]

with the boundary condition

\[
\psi_i(x^0_i) = 0.
\]

Eq. (1) can be solved by a standard fast marching method (FMM) [15].

A minimum time-to-capture function for pursuer \( i \) can be computed as

\[
\psi^c_i(y) = \min_{z \in \mathbb{R}} \left\{ \psi_i(z) \right\}.
\tag{2}
\]

Then a minimum time-to-capture function such that the evader is captured by at least one pursuer can be computed as

\[
\psi^c(y) = \min_{i=1, \ldots, N} \psi^c_i(y).
\]

Similarly, define the minimum time-to-capture function \( \varphi_e : \Omega \rightarrow \mathbb{R} \) for the evader, constrained to the set \( \mathcal{S}_e \) as:

\[
\varphi_e(y) = \min \left\{ t \left| x_e(t) = x^0_e, x_e(t) = y, x_e(s) \in \mathcal{S}_e, \forall s \in [0, t] \right. \right\}.
\]
where \( x^0_i \) denotes the initial position of the evader. The minimum time-to-reach function \( \varphi_e \) is also the viscosity solution to a HJB equation similar to eq. (1), i.e.,

\[
- \min_{u_e \in U_e} \{ \nabla \varphi_e(y) \cdot u_e \} = 1
\]

with the boundary conditions

\[
\varphi_e(x^0) = 0, \quad \varphi_e(y) = \infty, \quad y \in \Omega \setminus S_c.
\]

We can see that \( S_c \) depends on \( \varphi_e \) and \( \psi^c \), i.e., if \( \varphi_e(y) < \psi^c(y) \), then \( y \in S_c \). A modified fast marching method (mFMM) developed in [13] is then employed to compute \( S_e \) and \( \varphi_e \) simultaneously based on \( \psi^c \). An example of safe set is shown in Fig. 3 where \( P_1, P_2 \) and \( E \) denote pursuer 1, pursuer 2 and the evader, respectively. Here, all the agents have the same maximum speed. The contours of \( \psi^1, \psi^2 \), and \( \varphi_e \) are also depicted. Notice that the point \([6, 0]^T\) is nearer to pursuer 2 than the evader. However, \([6, 0]^T\) is safe-reachable because pursuer 2 can not go through the obstacle.

### B. Pursuit Strategy: Minimize the Safe Area

Using mFMM, we can obtain discrete values of \( \varphi_e(y_{i,j}) \) and \( \psi^c(y_{i,j}) \) where \( y_{i,j} = [ih, jh]^T \in \Omega \) with \( h \) the grid spacing. So if \( \varphi_e(y_{i,j}) < \psi^c(y_{i,j}) \), then \( y_{i,j} \) is safe-reachable, i.e., \( y_{i,j} \in S_c \). The area of \( S_c \) can thus be approximated as

\[
A \approx N_s h^2
\]

where \( N_s \) is the grid node number of \( S_c \).

A pursuit strategy for pursuer \( i \) is proposed to minimize the safe area using the gradient with respect to pursuer \( i \)'s movement which can be numerically approximated by perturbing pursuer \( i \) by \( h \) horizontally and vertically on the grid and recomputing \( A \), i.e.,

\[
u_i(t) = \mu_i(x_e, x_1, \ldots, x_N) = -v_i, \max \frac{\partial A}{\partial x_i} \left\| \frac{\partial A}{\partial x_i} \right\|
\]

### C. Cooperative Computation of Control Inputs

The method proposed above has taken centralized computation in a global coordinate system and the accurate position of each agent is required. However, in our problem, there is no global coordinate system and the positions of other agents can only be estimated in the moving frame by each pursuer. Now we propose a distributed method for pursuer \( i \) to compute its control input as follows.

- At step \( k \), pursuer \( i \) localizes other agents using the methods developed in Section III and obtains an estimation of relative positions \( \hat{r}_{i,j}(k), j = 1, \ldots, i-1, i+1, \ldots, N, e. \)
- It computes \( \psi_i(y), \forall y \in \Omega \) in its own frame where its current position is \([0, 0]^T\). Then \( \psi^c_i(y), \forall y \in \Omega \) is computed as (2) and is broadcast to other pursuers.
- It receives \( \psi^c_j(y), \forall y \in \Omega \), for every pursuer \( j \neq i \) which is computed in pursuer \( j \)'s moving frame and broadcast by pursuer \( j \).
- It estimates \( \hat{\psi}^c_i(y), \forall y \in \Omega \) in its own frame as follows

\[
\hat{\psi}^c_i(y) = \min \{ \psi_i^c(y), \min_{\text{every pursuer } j \neq i} \hat{\psi}^c_j(y - \hat{r}_{i,j}) \}.
\]

- It estimates the safe set \( \hat{S}_c \) using mFMM on the basis of \( \hat{\psi}^c_i(y), \hat{r}_i \) and \( \hat{r}_{i,c} \). Then the safe area \( A \) and the gradient of \( A \) are estimated as described in Section IV-B. Finally the control input is determined as eq. (4) in order to minimize the safe area of the evader.

Each pursuer determines its own control input using the above method and the computation is distributed among all pursuers.

### V. Simulation

This section provides the simulation results of the pursuit-evasion game. The performance of both localization and safe set estimation is analyzed.

As shown in Fig. 3 we consider 2 pursuers vs 1 evader taking place in a working area of \( 10m \times 10m \) with a rectangle obstacle. The maximum speed of each agent is 1m/s. Each pursuer maintains 50 particles to estimate another pursuer. The model noise and measurement noise are assumed to be white Gaussian with zero means. The variance of model noise \( w_i \) satisfies \( \sigma^2_{w_i} = 0.1^2 \) and the variance of measurement noise \( v_i \) satisfies \( \sigma^2_{v_i} = 0.01^2 \), \( i = 1, 2 \). The control input of the evader is determined manually at each time step. The capture distance is \( d_c = 0.5m \). The grid spacing is \( h = 0.1m \) and the sampling time is \( T = 0.1s \). The initial position of pursuer 1 and pursuer 2 are \([2, 9]^T\) and \([3, 3]^T\), respectively. The initial position of the evader is \([5, 6]^T\). The trajectories of all agents are shown in Fig. 4. At \( t = 8.2s \), the evader is captured by pursuer 2. We can see that pursuer 2 is finally within the capture distance which is depicted by a dashed circle in Fig. 4.

### A. The Performance of Localization

Now we evaluate the localization performance of each pursuer as follows. For pursuer \( i \), two metrics are adopted, i.e.,

\[
e_i^2(k) = \sum_{j \neq i} ||r_{i,j}(k) - \hat{r}_{i,j}(k)||^2.
\]
\[ e^e_i(k) = \| r_{i,e}(k) - \hat{r}_{i,e}(k) \| \]

where \( e^p_i(k) \) and \( e^e_i(k) \) denote the estimation error of another pursuer and the evader at time step \( k \), respectively. Fig. 5 shows the localization results estimated by pursuer 1 during the whole game. The time evolution of \( e^p_1 \) and \( e^e_1 \) are shown in Fig. 6 where we can see that during most time both \( e^p_1 \) and \( e^e_1 \) are less than one grid size 0.1m.

\[ A = h^2 \sum_i \sum_j S_e(i,j) \]

with the unit m². A metric called safe set estimation error \( e^s_i \) for pursuer \( i \) is defined as

\[ e^s_i = \sum_i \sum_j |\hat{S}_e(i,j) - S_e(i,j)| \]

where \( \hat{S}_e \) is estimated by pursuer \( i \) in its moving frame at each step. As depicted in Fig. 7, during most time both \( e^s_1 \) and \( e^s_2 \) are less than 10%. However, when agents get closer, the error becomes larger and unstable. This is because the numerical solutions obtained by mFMM differ much even with a small position estimation error if agents are very close. An illustration of difference between the actual safe set and the estimated safe set is given in Fig. 8.

**VI. CONCLUSION**

This paper studies a pursuit-evasion game without a global coordinate system. By using particle filtering, each pursuer estimates other pursuers’ positions in its own frame. Then an
angle based localization method is adopted to estimate the position of the evader. Distributed safe set computation as well as its gradient with respect to the pursuer’s movement are performed to determine their control inputs with the help of a modified fast marching method. Pursuers are not required to know any prior information of the evader but its maximum speed. Future work includes considering more complicated dynamics of agents and limited range of sensors, combing with Simultaneous Localization and Mapping (SLAM) in an unknown environment, and providing analysis for capture conditions.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation (NNSF) of China under Grant 61273113.

REFERENCES


