Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

A Cooperative Pursuit-Evasion Game for Non-holonomic Systems

Mangal Kothari * **Joel G. Manathara** ** **Ian Postlethwaite** ***

* *M. Kothari is with the Faculty of Engineering and Environment, Northumbria University, Newcastle upon Tyne, UK, NE1 8ST, (email: mangal.kothari@northumbria.ac.uk)*
** *J. G. Manathara is with the Department of Aerospace Engineering, Indian Institute of Technology Madras, Chennai, India, (email: joel@iitm.ac.in)*
*** *I. Postlethwaite is Deputy Vice-Chancellor at Northumbria University, Newcastle Upon Tyne, UK, NE1 8ST, (email: ian.postlethwaite@northumbria.ac.uk)*

Abstract-This paper considers a pursuit-evasion game for non-holonomic systems where a number of pursuers attempt to capture a single evader in a bounded connected domain. The problem is challenging because all vehicles have the same manoeuvring capability and are subject to turn radius constraints making them non-holonomic systems. The paper initially presents simple and alternate proofs for results existing in the literature that guarantee capture for holonomic systems. These results that are based on the minimization of safe-reachable area (the set of points where an evader can travel without being caught) are then extended to non-holonomic systems. However, solving such a problem exactly is computationally intractable. Therefore, the paper proposes a computationally efficient algorithm to obtain an approximate solution to the safe-reachable area minimization problem where the pursuers aim to minimize the safe-reachable area of the evader, while the evader chooses control actions to maximize it. Also proposed is an alternative approach that uses a cooperative strategy based on a pure proportional navigation law to capture the evader. In the process, an evader strategy which is superior to those based on the minimization of safe-reachable area is identified. The paper evaluates the proposed algorithms through numerical simulations.

Keywords: Pursuit-evasion games, safe-reachable sets, pursuit and pure proportional navigation guidance laws, capaturability.

1. INTRODUCTION

This paper considers pursuit-evasion games for systems of unmanned aerial vehicles (UAVs) in which multiple pursuers try to capture a single evader. Solutions of pursuit-evasion games are useful in many civilian and military applications. Broadly speaking, the capture problem can be defined as two sub-problems: (i) encircle, and (ii) grasp. In the encircling problem, a group of UAVs is required to make a formation around an evader. In the grasping problem, UAVs are required to get close enough to an evader to perform some sophisticated task. The focus of this paper is cooperative capture of an evader, where the pursuers and the evaders have the same speed and manoeuvring capabilities.

The problem of the pursuit-evasion game can be posed as a differential game and the solution can be obtained by solving the associated Hamilton-Jacobi-Isaacs (HJI) partial differential equations (Isaacs, 1965). A single pursuit and a single evader problem where the evader and pursuer have equal speeds is treated from this point of view in Berkovitz (1975). However, as is well known, HJI suffers from the *curse of dimensionality* and is extremely difficult to solve, especially in real-time when the number of players increases.

The pursuit-evasion game for holonomic systems in a polygonal domain, where the pursuers and the evader have same maximum speed constraint, was studied in Isler et al. (2005). In Huang et al. (2011), the authors proposed a minimization of the safe-reachable area of the evader in order to capture an evader confined to move in a convex domain in finite-time, when the pursuers and the evader are assumed to move with same speed. To achieve this, they computed a gradient of the safe-reachable area using geometric properties with a Voronoi partition approach, and made each pursuer move in the negative gradient direction. Inspired by their work, we first propose a similar strategy for non-holonomic systems. However, computing safe-reachable area for a non-holonomic vehicle with minimum radius of turn is computationally intensive. In such situations, the philosophy of Model Predictive Control (MPC) theory can be used to reduce the computational burden. For computing a safe-reachable set/area for non-holonomic systems, we discretize the given domain into a fixed number of cells and determine the number of cells that are safe to visit using Dubins distance (Dubins, 1957) – minimum path length of a vehicle, with a constant speed and a minimum turn radius constraint, from the current position to a cell center. Then we propose an algorithm based on MPC to obtain computationally efficient solutions. The algorithm tries to provide a pursuer strategy that results in the minimization of the safe-reachable area of the evader. The algorithm is expected to perform well if we use a good approximation to the safe-reachable area and have a reasonable model of the evader's strategy.

Alternatively, the problem of target or evader capture can be studied using tools from the missile guidance literature (Zarchan, 2002; Shneydor, 1998). In missile guidance, using geometric properties and kinematics equations, simple laws are derived to intercept a target. The pursuit and the pure proportional navigation (PPN) laws are very popular in this respect. The idea behind the PPN guidance law is to keep the rate of rotation of the line-of-sight (LOS), the vector between missile and target, at zero and then to follow a collision course for interception. The PPN guidance law applies lateral acceleration proportional to the rate of rotation of the LOS. There are works in the literature that have used variants of pursuit (Belkhouche et al., 2005) and PPN (Jeon et al., 2010) guidance laws for cooperative capture of target/evader. However, these works assume that the pursuers have a speed advantage over the evader. As missiles usually have higher manoeuvrability (and speed) capabilities compared to a target, interception can be guaranteed. In this paper however, the pursuers are subject to the same speed constraint as the evader, and therefore, they cannot always satisfy the conditions required by PPN guidance law to achieve interception. However, because the game is being played in a closed domain, we can take advantage of this and try to attain favourable conditions by employing a cooperative PPN law. The cooperative PPN law ensures that one of the UAVs satisfies the collision triangle conditions, and hence that capture will occur. In order to escape, the evader reacts to the closest pursuer by applying the inverse-PPN law. We show empirically that the proposed strategy performs better than those based on the minimization of safe-reachable areas.

The rest of the paper is organized as follows. Section 2 formally describes the pursuit-evasion problem. Section 3 presents a capturing strategy for holonomic systems based on the minimization of safe-reachable areas. Section 4 proposes two strategies for non-holonomic systems. The first one extends the work of Huang et al. (2011) and the second one uses an idea from guidance theory. Numerical results are presented in Section 5 and concluding remarks are presented in Section 6.

2. PROBLEM STATEMENT

Consider a game of N pursuers and one evader taking place in the interior of a polytope $\Omega \in \mathbf{R}^2$. The aim of the pursers is to capture the evader by having one pursuer within Euclidean distance r_c of the evader during the game. The equations of motion of any player (pursuer or evader) are given as

$$\begin{aligned} \dot{x}_i &= v\cos\psi_i \\ \dot{y}_i &= v\sin\psi_i \\ \dot{\psi}_i &= \frac{u_i}{v} \end{aligned} \tag{1}$$

where $i \in (\mathbb{P}, e)$, $\mathbb{P} = \{1, \ldots, n\}$ is the set of pursuers and e denotes the evader. $[x_i,\ y_i] \in \Omega$ is the position of the i^{th} player, v is the speed of the i^{th} player (we assume the same speed for all players), ψ_i is the heading angle of the i^{th} player, and $u_i \in \mathbf{U}$ represents lateral acceleration, which acts as a control input for the i^{th} player. The set $\mathbf{U}$ represents a set of feasible control inputs and we assume that all players are subject to identical input constraints. The minimum distance from the evader to a pursuer at time t is defined as follows

$$r_{min}(t) \triangleq \min_i \|(e(t) - p_i(t))\| \tag{2}$$

Assuming that a pursuer can capture at distance r_c, the capture condition for the pursuers is then given by

$$r_{min}(t_c) \leq r_c,\ \text{for some } t_c \geq 0 \tag{3}$$

With this, the capture problem for a group of pursuers can now be defined.

Problem 1. Given the initial configuration $e(0), p_i(0) \in \Omega$ with $r_{min}(0) > r_c$, find a feasible set of pursuit inputs that satisfies the capturability condition (3) in finite time.

3. CAPTURING FOR A HOLONOMIC SYSTEM

As pursuers and evader have the same manoeuvrability capabilities, the problem is challenging. For holonomic systems, Huang et al. (2011) proposed a solution based on minimizing the safe-reachable area. In this section, we present a solution based on a pursuit strategy and derive the conditions for capturability.

The Voronoi partition approach is an efficient tool for computing safe-reachable areas for holonomic systems if all payers have the same speed Gavrilova (2008). A partition/cell is computed for each player in the domain Ω based on the positions of all players. In a particular Voronoi partition, the associated player can reach any point in the partition before any of the other players. Let $\mathbb{V}(\Omega) = \{V_e, V_1, \ldots, V_n\}$ be the Voronoi partition of Ω generated by the points $\{e, p_1, \ldots, p_n\}$:

$$\begin{aligned} V_e &= \{p \in \Omega \parallel p - p_e \| \leq \| p - p_i \|, \forall, i \in \mathbb{P}\} \\ V_i &= \{p \in \Omega : \| p - p_i \| \leq \\ &\quad \min(\| p - p_e \|, \| p - p_j \|), \forall, i, j \in \mathbb{P}, i \neq j\} \end{aligned}$$

From the above definition, it can be seen that the evader cell V_e is a regular polygon and the area A of V_e can be calculated using the coordinates of the vertices of V_e. Note that the area A depends on the locations of the evader and its neighbouring players. This is because neighbouring players can influence relative positions of shared boundaries and thus control the movement of vertices of the evader's cell. Observing this fact, the rate of change in A can be written as

$$\frac{dA}{dt} = \frac{\partial A}{\partial e}\dot{e} + \sum_{i \in \mathbb{N}_e} \frac{\partial A}{\partial p_i}\dot{p}_i \tag{4}$$

where $\mathbb{N}_e \subset \mathbb{P}$ is the set of the evader's neighbours. In order to make $\frac{dA}{dt}$ negative, we seek a strategy that enables each player to move in a direction that is opposite to $\frac{\partial A}{\partial p_i}$, that means $\dot{p}_i = -\frac{\partial A}{\partial p_i}$. Towards this, we propose a pursuit strategy in which each player follows the command

$$\psi_i^* = tan^{-1}\left(\frac{y_{ci} - y_i}{x_{ci} - x_i}\right) \tag{5}$$

where (x_{ci}, y_{ci}) is the centre point of the shared boundary (line of control) and ψ_i^* is the commanded heading. The above strategy guides a pursuer towards the centre point of the shared boundary. Next we prove that with the above mentioned strategy, capture will occur. We first recall the following lemma from Huang et al. (2011).

Lemma 3.1. For any $i \in \mathbb{N}_e$, the gradient vectors in the local frame are as follows.

$$\frac{\partial A}{\partial \mathbf{p_i}} = [\alpha_{h_i} \ \alpha_{v_i}]$$
$$\frac{\partial A}{\partial \mathbf{e}} = [\alpha_{h_i} \ -\alpha_{v_i}]$$

where $\alpha_{h_i} = -\frac{L_i}{2}$ and $\alpha_{v_i} = \frac{l_i^2-(L_i-l_i)^2}{2\|r_i\|}$. Here, L_i is the length of the line of control B_i, l_i is the length of the segment below the LOS, and $\|r_i\|$ is the length of the LOS. These are also shown in a schematic given in Figure 1.

Proof. The proof of the above lemma is given in Huang et al. (2011).

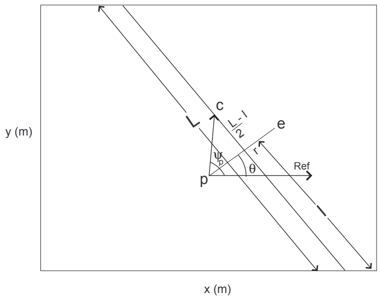

Figure 1. Pursuer-evader engagement geometry

For simplicity, we will drop subscript i from here onwards in this section, and show that capture is possible even with just one pursuer for a holonomic system.

Lemma 3.2. The pursuit guidance points in the negative of the gradient direction $\frac{\partial A}{\partial \mathbf{p}}$.

Proof. Let α be the angle of the gradient vector $\frac{\partial A}{\partial \mathbf{p}}$, then

$$\tan\alpha = \frac{\alpha_v}{\alpha_h} = \frac{\frac{l^2-(L-l)^2}{2\|r\|}}{\frac{-L}{2}} = -\frac{2l-L}{\|r\|}$$

Let β be the angle in the relative frame which point towards the mid point of the line of control B, then

$$\tan\beta = \frac{(L/2)-l}{(\|r\|/2)} = \frac{L-2l}{\|r\|}$$

As $\alpha_h \leq 0$ and $-\frac{\pi}{2} \leq \alpha \leq \frac{\pi}{2}$, we have

$$\beta = \pi + \alpha$$

Hence, the proposed strategy points in the direction of $-\frac{\partial A}{\partial \mathbf{p}}$.

Lemma 3.3. Under the proposed pursuit strategy, the area A satisfies $\frac{dA}{dt} \leq 0$ for any admissible evader control input. Furthermore, $\frac{dA}{dt} = 0$ if, and only if, the evader follows the following command

$$\psi_e^* = \pi + tan^{-1}\left(\frac{y_{ci}-y_i}{x_{ci}-x_i}\right)$$

Proof. The rate of change in A is given by

$$\begin{aligned}\frac{dA}{dt} &= \frac{\partial A}{\partial e}\dot{e} + \frac{\partial A}{\partial p}\dot{p} \\ &= [\alpha_h \ \alpha_v]\begin{bmatrix} v\cos(\psi_p-\theta) \\ v\sin(\psi_p-\theta)\end{bmatrix} + [\alpha_h \ -\alpha_v]\begin{bmatrix} v\cos(\psi_e-\theta) \\ v\sin(\psi_e-\theta)\end{bmatrix}\end{aligned}$$

where θ is the LOS angle and is depicted in Figure 1.

$$\begin{aligned}\frac{dA}{dt} &= [\alpha_h \ \alpha_v]\begin{bmatrix} v\cos\beta \\ v\sin\beta\end{bmatrix} + [\alpha_h \ -\alpha_v]\begin{bmatrix} v\cos(\psi_e-\theta) \\ v\sin(\psi_e-\theta)\end{bmatrix} \\ &= [\alpha_h \ \alpha_v]\begin{bmatrix} -v\frac{\alpha_h}{\sqrt{\alpha_h^2+\alpha_v^2}} \\ -v\frac{\alpha_v}{\sqrt{\alpha_h^2+\alpha_v^2}}\end{bmatrix} + [\alpha_h \ -\alpha_v]\begin{bmatrix} v\cos(\psi_e-\theta) \\ v\sin(\psi_e-\theta)\end{bmatrix} \\ &= -v\sqrt{\alpha_h^2+\alpha_v^2} + [\alpha_h \ -\alpha_v]\begin{bmatrix} v\cos(\psi_e-\theta) \\ v\sin(\psi_e-\theta)\end{bmatrix} \leq 0\end{aligned}$$

where equality holds if, and only if,

$$\cos(\psi_e-\theta) = \frac{\alpha_h}{\sqrt{\alpha_h^2+\alpha_v^2}}$$
$$\sin(\psi_e-\theta) = -\frac{\alpha_v}{\sqrt{\alpha_h^2+\alpha_v^2}}$$

This can be achieved by setting

$$\psi_e^* = \pi + tan^{-1}\left(\frac{y_{ci}-y_i}{x_{ci}-x_i}\right)$$

In order to demonstrate that capture occurs, we show that the relative distance r between the pursuer and the evader is decreasing even when the area A is kept constant. For this we need the following lemma.

Lemma 3.4. If $\frac{dA}{dt} = 0$, then under the proposed pursuit strategy the following holds

$$\frac{d}{dt}r^2 = -4v\frac{r^2}{\sqrt{r^2+4s^2}} \leq 0$$

Proof.

$$\begin{aligned}\frac{d}{dt}r^2 &= 2r(\dot{e}-\dot{p}) \\ &= 2r(v\cos(\psi_e-\theta) - v\cos(\psi_p-\theta)) \\ &= -4rv\cos(\psi_p-\theta) \\ &= -4v\frac{r^2}{\sqrt{r^2+(L-2l)^2}} \leq 0\end{aligned}$$

In this section, we have shown, using mathematical tools borrowed from missile guidance theory, that capture based on a pursuit guidance law can be achieved in finite time for a holonomic pursuer-evader setting in a closed domain; see Huang et al. (2011) for more details. We will use the evading strategy presented in Lemma 3.3, with the addition of minimum turn radius constraint, to evaluate the performance of our proposed algorithm for a non-holonomic system in Section 5.

4. CAPTURING FOR A NON-HOLONOMIC SYSTEM

This section first extends the idea of minimizing the safe-reachable area to non-holonomic systems. The resultant problem is computationally intensive, and therefore we propose a computationally efficient algorithm to obtain solutions. The solution strategy involves the use of Model Predictive Control (MPC) philosophy and an efficient algorithm to approximately compute the safe reachable area. For the non-holonomic pursuer-evader problem, we also propose a cooperative strategy based on a Pure Proportional Navigation (PPN) guidance law that is simple and easy to implement.

4.1 A cooperative algorithm to minimize safe-reachable areas

The idea of safe-reachable area minimization works in two steps: (i) computing the safe-reachable area; and (ii) finding control commands for each pursuer that minimize the safe-reachable area of the evader. Towards this, we first define the safe-reachable set of an evader in a non-holonomic setting as follows

$$\mathbb{S}_e = \{p \in \Omega \mid D(p, p_i) - D(p, e) > 0\} \tag{6}$$

where D represents a function measuring the Dubins distance from an evader (or a pursuer) to a point p Dubins (1957). We now define a joint control input vector of all pursuers $u = [u_1, \ldots, u_n]^T$ and pose the following optimization problem for capturability:

$$J(u) = \min_{u \in \mathbf{U}^n} \max_{u_e \in \mathbf{U}} A(\mathbb{S}_e) \tag{7}$$

where U is the set of allowable controls. U is bounded and accounts for the non-holonomic constraint of minimum turn radius or maximum turn rate. Here $A(\mathbb{S}_e)$ represents the area of the safe-reachable set $\mathbb{S}_e$. The main difficulty in solving (7) is that there is no explicit method available to compute $A(\mathbb{S}_e)$. Secondly, any optimization algorithm would require the evaluation of $J(u)$ repeatedly to find the optimal solution $J^*(u)$. This makes the problem extremely difficult to solve, especially in real-time. To obtain approximate solutions, we make approximations to $A(\mathbb{S}_e)$ and then solve a simplified optimization problem. Towards this, we present a computationally efficient algorithm to approximately compute the safe reachable area of the evader (see Algorithm 1). Then we solve the optimization problem in (7) using MPC framework.

Algorithm 1 Safe Reachable Area Computation

Input: a set of discrete grids $\mathbb{G} \in \Omega$, p_i, $e \in \Omega$, ψ_i, ψ_e, $\mathbb{O} = \emptyset, \mathbb{C} = \emptyset$, the area of one cell A_c

1: $i_e \leftarrow Find\_Cell(e, \mathbb{G})$
2: **if** $Dubins\_Cost(i_e) \geq 0$ **then**
3: $\quad \mathbb{O} \leftarrow Include\_Cell(\mathbb{O}, i_e)$
4: **else**
5: $\quad \mathbb{N}_e \leftarrow Find\_Neighbor(i_e)$
6: $\quad$ **if** $Dubins\_Cost(j \in \mathbb{N}_e, \forall j \in \mathbb{N}_e) \geq 0$ **then**
7: $\quad\quad \mathbb{O} \leftarrow Include\_Cell(\mathbb{O}, j)$
8: $\quad$ **end if**
9: **end if**
10: **while** $\mathbb{O}$ is not empty **do**
11: $\quad \mathbb{N}_e \leftarrow Find\_Neighbor(\mathbb{O}(1))$;
12: $\quad$ **if** $Dubins\_Cost(j \in \mathbb{N}_e, \forall j \in \mathbb{N}_e) \geq 0$ **then**
13: $\quad\quad \mathbb{O} \leftarrow Include\_Cell(\mathbb{O}, j)$
14: $\quad$ **end if**
15: $\quad \mathbb{C} \leftarrow Transfer\_Element(\mathbb{O}(1))$
16: **end while**
17: Return the total area $\hat{A} = A_c \times \mid \mathbb{C} \mid$

The safe-reachable area for a non-holonomic system can be computed using Dubins distance to all the points in Ω. As Dubins distance from the i^{th} player to point p depends both on position and heading of the i^{th} player, the task is computationally expensive. Moreover, the complexity increases as the size of Ω increases. To compute the safe-reachable area in an efficient way, we propose an algorithm that discretizes Ω into a fixed number of grid cells $\mathbb{G}$ and determines the number of cells that are safe to visit. The approximate area $\hat{A}(\mathbb{S}_e)$ is obtained by multiplying the area of a single cell with the number of safe cells. Whether a cell c is safe to visit or not is determined using the following cost function

$$J_D(c) = \min_{i \in \mathbb{P}} D(c, p_i) - D(c, e), \; c \in \mathbb{G} = 1, \ldots, |\mathbb{G}| \tag{8}$$

If $J_D(c) > 0$, then the cell c is declared safe to visit; this means that the evader can reach the centroid of c before any other pursuers. To determine the number of cells that are safe to visit in $\mathbb{G}$, one may have to compute the cost $J_D(c)$ at each grid point, which is cumbersome when Ω is large. We employ a Dijkstra type algorithm to calculate efficiently the number of cells that are safe to visit without computing the cost at every grid point, but checking the safe-reachability of only the neighbouring cells of already checked cells, starting from the current cell or a known safe cell.

The algorithm first checks whether the evader's own cell is safe or not. If the evader's cell is safe to visit, then it is included in an open set called $\mathbb{O}$. If the cell is not safe to visit, then it is discarded and cells that are in the neighbourhood of the evader's cell are evaluated to decide whether they are safe to visit. Cells that are safe are included in $\mathbb{O}$ (steps 1-4). If no cell is safe to visit, this implies that the evader cannot escape if the group of pursuers employs an appropriate pursuit strategy. Next, neighbours of the first element of $\mathbb{O}$ are processed to decide whether or not to include them in $\mathbb{O}$. Those neighbours who have a positive cost are included in $\mathbb{O}$ and the rest are discarded. Then the first element of the set $\mathbb{O}$ is moved into another set $\mathbb{C}$ called the closed set. The process of evaluating neighbours of the first element of $\mathbb{O}$ (to decide whether or not to include them in $\mathbb{O}$) and moving the corresponding first element of $\mathbb{O}$ into $\mathbb{C}$ is continued until the set $\mathbb{O}$ becomes empty. At the end, the set $\mathbb{C}$ contains all the cells which have positive cost and are safe to visit. The approximated area is obtained by multiplying the number of those cells in the set $\mathbb{C}$ with the area of a single cell. Note that the accuracy of this approximation depends on the discretization of Ω. With this, we can write the simplified optimization problem from (7) as follows

$$\hat{J}(u) = \min_{u \in \mathbf{U}^n} \max_{u_e \in \mathbf{U}} \hat{A}(\mathbb{S}_e) \tag{9}$$

However, the problem still remains intractable. This is because the optimization problem is combinatorial in nature, which requires computing $\hat{J}(u)$ for all feasible control inputs and for combinations of all these inputs. For the pursuers to successfully capture the evader, we have to drive $A(\mathbb{S}_e) \approx \hat{A}(\mathbb{S}_e)$ to zero. Instead of solving (9), we aim to find control actions for each pursuer that ensure $\hat{A}(\mathbb{S}_e)(t+1) < \hat{A}(\mathbb{S}_e)(t)$. For this purpose, we use the MPC philosophy in which we look ahead for fixed steps and compute control commands sequentially. The steps of the algorithm are given in Algorithm 2.

The algorithm presented above sequentially computes control commands for each pursuer by solving the following simplified optimization problem for each pursuer

Algorithm 2 Solving optimization to compte control inputs

Input: a set of discrete grids $\mathbb{G} \in \Omega$, $p_i(t)$, $e(t) \in \Omega$, $\psi_i(t)$, $\psi_e(t)$.
1: **for** i = 1 to n **do**
2: Set $u_j(t+1:t+N-1|t) = 0, \forall j \in \{i+1,\ldots,n,e\}$
3: Generate $p_j(t+N|t), \psi_j(t+N|t), e(t+N|t), \psi_e(t+N|t)$
4: Solve $\min_{u_i \in \mathbb{U}_D} \hat{A}$
5: Set $u_i(t+1:t+N-1|t) = u_i^*$
6: Generate $p_i(t+N|t), \psi_i(t+N|t)$
7: **end for**
8: Solve $\max_{u_e \in \mathbb{U}_D} \hat{A}$
9: Return u and u_e

$$J_i(u_i(t)) = \min_{u_i \in \mathbf{U}_d} \hat{A}_{t+N}(\mathbb{S}_e) \tag{10}$$

where $\mathbf{U}_d \triangleq \{-u_{max}, -\frac{u_{max}}{2}, 0, \frac{u_{max}}{2}, u_{max}\}$ is the discrete control set, and we choose u_i from the fixed set $\mathbf{U}_d$ to speed up the algorithm. Although we have chosen five points for discretization, one can choose more or less to obtain reasonable results. When solving (10), we use a prediction horizon of N steps. The optimization process assumes that while computing control inputs for the i^{th} player, the control inputs for the remaining players are constant, and are given by

$$u_j(t+k|t) = \begin{cases} 0 & j > i \\ u_j^*(t) & j < i. \end{cases} \;\forall\, k = [1,\ldots,N-1], j \neq i \tag{11}$$

Note that $u_j^*(t)$ is the optimal control input, already obtained for the j^{th} player by solving (10). Using the control inputs, $p_j(t+N|t), \psi_j(t+N|t), e(t+N|t)$ and $\psi_e(t+N|t)$ are generated to calculate $\hat{A}_{t+N}(\mathbb{S}_e)$ at time step $(t+N)$. As the prediction horizon (N steps) and the control set $\mathbf{U}_d$ are of fixed lengths, it is possible to characterize *a priori* trajectories for each input $u_i \in \mathbf{U}_d$. We compute trajectory templates $\mathbb{T}$ off-line for each $u_i \in \mathbf{U}_d$ and use them in the optimization process to obtain $p_i(t+N|t), \psi_i(t+N|t)$ directly. This further speeds up the algorithm as the optimization process chooses a template trajectory that minimizes the cost function rather than running the whole optimization with state equations as the constraint. If no trajectory (template) satisfies $\hat{A}(\mathbb{S}_e)(t+N) < \hat{A}(\mathbb{S}_e)(t)$, then the corresponding player continues to travel in the same direction.

In this subsection, we have presented a computationally efficient algorithm to solve the pursuit-evasion game while accounting for turn radius constraints. However, the performance of the algorithm will depend on the discretization of Ω to compute the approximate area and the simplified optimization problem posed in (10). To obtain meaningful solutions, one has to choose appropriate values for the number of grid cells in Ω and the length of the prediction horizon in (10).

4.2 A cooperative proportional navigation algorithm

As mentioned in the previous subsection, the bottleneck in the approach lies in computing the safe-reachable area and solving the associated combinatorial optimization (7). In this subsection, we look at an alternative approach. To obtain simple and effective solutions, we propose a cooperative law based on a PPN law from missile guidance theory. In missile guidance, the PPN law generates control commands proportional to the rate of rotation of the line-of-sight (LOS) to ensure that the LOS does not rotate in the body frame of a missile. The rate of change in the LOS separation r_i and the rate of rotation of the LOS $\dot{\theta}_i$ for the i^{th} pursuer are given by

$$\begin{aligned} \dot{r}_i &= v\cos(\psi_e - \theta_i) - v\cos(\psi_{p_i} - \theta_i) \\ \dot{\theta}_i &= \frac{v\sin(\psi_e - \theta_i) - v\sin(\psi_{p_i} - \theta_i)}{r_i} \end{aligned} \tag{12}$$

A zero rate of LOS rotation and a reducing LOS separation will ensure interception or capture. These conditions are met in missile guidance problems through the application of PPN guidance law as missiles usually have a velocity advantage over their targets. Since the evader and the pursuers have same speed for the problem that we consider, we resort to a cooperative capture strategy that is built upon some nice properties of the PPN guidance law that we show below. The following two lemmas sheds some light into why a cooperation can help in a case where multiple pursuers are following an evader using PPN guidance law.

Lemma 4.1. If a pursuer uses the PPN law to achieve capture and the evader uses the inverse-PPN law to avoid capture, then $\psi_p = \psi_e$ are equilibrium points of (12) if $\psi_p + \psi_e - 2\theta < \pi$ (we drop the subscript 'i' here onwards for the sake of clarity).

Proof. Let $V = \frac{1}{2}(\psi_e - \psi_p)^2$ be the Lyapunov cost function. Taking the time derivative along the trajectories of ψ_p and ψ_e, we get

$$\begin{aligned} \dot{V} &= (\psi_e - \psi_p)(\dot{\psi}_e - \dot{\psi}_p) \\ &= -(\psi_e - \psi_p) 2\, N\, v\, \dot{\theta} \\ &= -\frac{2Nv^2(\psi_e - \psi_p)}{r} \sin(\psi_e - \theta) - \sin(\psi_p - \theta) \\ &= -\frac{4Nv^2(\psi_e - \psi_p)}{r} \sin\left(\frac{\psi_e - \psi_p}{2}\right) \cos\left(\frac{\psi_e + \psi_p - 2\theta}{2}\right) \end{aligned}$$

Noting that $(\psi_e - \psi_p)\sin\left(\frac{\psi_e - \psi_p}{2}\right) \geq 0$, then $\dot{V} \leq 0$ only if $\psi_p + \psi_e - 2\theta < \pi$. If the latter is satisfied, then ψ_p and ψ_e will attain the same value on reaching the equilibrium.

Lemma 4.2. If a pursuer uses the PPN law to achieve capture and an evader is travelling on a straight line, then $\psi_p = \psi_e$ are equilibrium points of (12) if $\psi_p + \psi_e - 2\theta < \pi$.

Proof. The proof remains the same as in Lemma 4.1 except $\dot{\psi}_e = 0$. Because of this, we get

$$\dot{V} = -\frac{2Nv^2(\psi_e - \psi_p)}{r} \sin\left(\frac{\psi_e - \psi_p}{2}\right) \cos\left(\frac{\psi_e + \psi_p - 2\theta}{2}\right)$$

and the argument stays the same, which implies $\dot{V} \leq 0$ only if $\psi_p + \psi_e - 2\theta < \pi$. If this is satisfied, then ψ_p and ψ_e will attain the same value to reach the equilibrium.

It can be seen from these two lemmas that if an evader applies the inverse-PPN law or travels on a straight line, then pursuers can attain the same heading if they satisfy $\psi_p + \psi_e - 2\theta < \pi$. Under the equilibrium condition $\dot{r}_i = 0$, the separation between a pursuer and the evader will be constant at a distance that depends on the initial conditions. As the evader is confined to be in a bounded region, it is possible to employ other pursuers so as to decrease the minimum of the LOS separations of the evader with the pursuers. We give an algorithm for multiple pursuers to capture an evader in a bounded region using PPN guidance law in Algorithm 3.

The algorithm presented above asks each pursuer to apply the PPN law if they are in the domain Ω. If any pursuer is about to hit the boundary, which is predicted based on its path and

Algorithm 3 Cooperative PPN algorithm

Input: $p_i(t),\ e(t) \in \Omega,\ \psi_i(t),\ \psi_e(t), \Omega$.

1: **for** i = 1 to n **do**
2:     **if** $p_i(t+t_c) \notin \Omega$ **then**
3:         Generate u_i to point towards the centroid of Ω
4:     **else**
5:         Set $u_i = N\ v\ \dot{\theta}_i$
6:     **end if**
7:     Find $p_i(t+N|t), \psi_i(t+N|t)$
8: **end for**
9: Find j that $\min_{j \in \mathbf{P}} \|(e(t), p_i(t))\|$
10: Set $u_e = -N\ v\ \dot{\theta}_j$
11: Return u and u_e

minimum turn radius (line 2), then the pursuer takes a minimum radius turn until it points towards the centroid of Ω. This ensures that no pursuers leave the region Ω. The evader chooses the inverse-PPN law to avoid capture from the closest pursuer when it is inside Ω. While performing evasive manoeuvres, the evader should also remain within the bounded region. When an evader is approaching a boundary, it can turn either clockwise or anticlockwise to stay within Ω. The evader chooses that direction of turn that maximizes its minimum distance to any of the pursuers, assuming that the pursuers were to continue in their previous paths without making any manoeuvres. It is important to emphasis that we do not know the best evader action but this particular choice seems reasonable. We believe this is because the evader reacts to the closest pursuer. This will be shown empirically in the next section.

Under the cooperative strategy as proposed in Algorithm 3, the closest pursuer will force the evader to move in a straight line as argued in Lemma 4.1. This will enable the other pursuers to get closer to the evader while it is making a turn. In this subsection, we have proposed an approach based on the PPN law to capture an evader. In this process, we have also identified that the inverse-PPN as a potential evading strategy, the efficacy of which is empirically evaluated in the next section.

5. NUMERICAL RESULTS

In this section, we demonstrate the performance of our proposed algorithms for non-holonomic systems by three examples. In the first example, the performance of minimizing the safe-reachable area approach (Algorithm 2) is evaluated. In the second example, the performance of the cooperative strategy based on the PPN guidance law is evaluated. In the third example, we present a comparative study. The equations of motion of each player are given in (1).

For the safe-reachable based approach (Algorithm 2), we consider a square region of $1km \times 1km$ and discretize it into 100 cells, the size of each cell is $0.1km \times 0.1km$. The initial positions and headings of the pursuers are chosen randomly. A snapshot of a discretized area with two pursuers and one evader is given in Figure 2. We have also plotted trajectories of each player in the figure. We consider a prediction horizon of one second ($N = 10$ steps) in computing control commands. This means that the algorithm updates control after each one second and within the horizon the control is kept constant (constant parametrization). We have considered four different scenarios by increasing the number of pursuers from 2 to 5. The number of the safe-reachable cells is plotted for each case in Figure 3. It can be seen from the figure that the algorithm tries to find the control inputs that reduce the safe-reachable area for the evader. It can also be noticed that if we increase the number of pursuers, then the time for capturing reduces significantly. However, we can see that the area increases at some instances during the game instead of decreasing. This is because there is no control input $u_i \in \mathbf{U}_d$ that minimizes the safe-reachable area. This happens due to the discretization of $\mathbb{U}$ and due to constraints on control inputs; hence, there is a sharp increase in the number of cells, especially during the endgame, when the pursuers and evader are close to each other. Note that the performance of the algorithm depends on discretization of Ω and $\mathbb{U}$ and the length of the prediction horizon. If we increase the number of cells in Ω, then with more control inputs in $\mathbf{U}_d$, especially during the endgame, there may exist control inputs that may control the sharp increase in the number of safe-reachable cells.

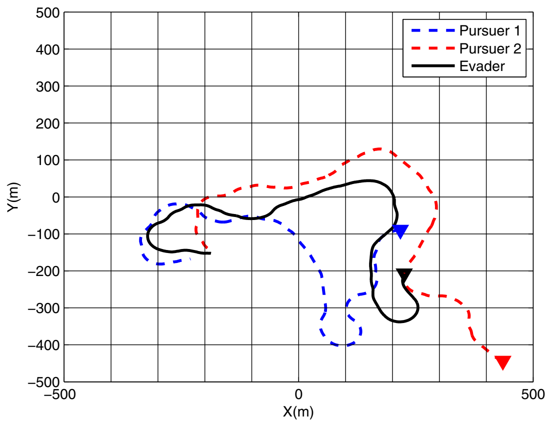

Figure 2. Pursuit evasion scenario for two pursuers and one evader. The region is discretized into 100 cells. The initial positions are shown by triangles.

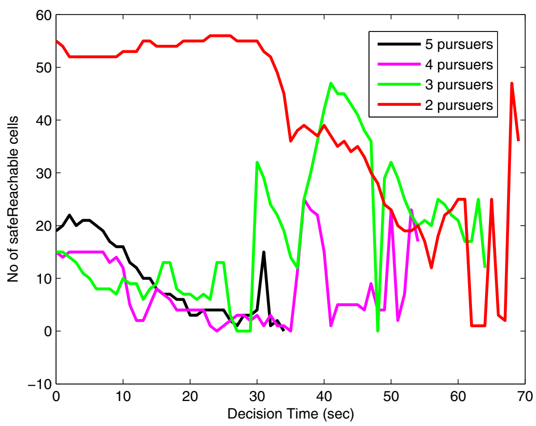

Figure 3. The number of safe-reachable cells with different scenarios. The approximated area is computed at an interval of 1 second, which is equal to the prediction horizon.

As can be seen from Figure 3, the performance of the algorithm suffers during the endgame mainly due to discretization. Hence, we may have to fine tune our strategy during the endgame to capture more quickly. In this paper, instead of switching to another strategy during the endgame, we have looked at another approach that is based on a PPN guidance law. The performance of this approach is evaluated against three different evader strategies: (i) the Voronoi partition approach, as described in Section 3; (ii) minimizing (or maximizing for an evader) the safe-reachable area, as described in Algorithm 2; and (iii) using the inverse-PPN law, as described in Algorithm 3. In each case, a group of pursuers use Algorithm 3 to capture the evader.

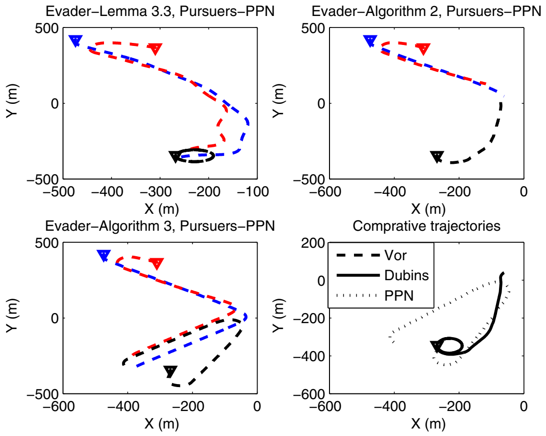

Figure 4. Pursuit evasion scenario for two pursuers and one evader.

Table 1. Simulation results, cooperative PPN strategy versus different evasion strategies

Team size	Over 20 runs								
	Voronoi			Safe-reachable area			inverse-PPN		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
2	27.7	150.7	69.9	16.5	110.5	58.6	57.5	225.8	114.4
3	6.0	76.3	42.1	10.9	76.8	37.4	25.1	245.6	70.9
4	12.2	63.3	37.2	12.5	53.4	32.8	20.4	93.9	50.3
5	5.0	60.2	29.7	5.7	51.2	32.1	11.7	135.8	49.5

Figure 4 shows trajectories of all players under different evader strategies. It can be noticed from the sub-figures that in each case a pair of pursuers capture the evader; the inverse-PPN strategy looks to be superior compared to the other two strategies. Under this strategy the pursuers take more time (86 sec) to capture the evader compared to other strategies (76.3 sec for Voronoi based approach and 43.8 sec for the safe-reachable area based approach). To investigate this claim empirically, we have conducted 20 runs for each case and summarized the results in Table 1. It can be seen that the minimum, maximum, and average times for capture are similar when the evader uses strategies based on Voronoi and safe-reachable approaches. However, it takes almost double the time for capture when the evader uses inverse-PPN approach. This is because in the cooperative PPN algorithm the evader has a superior strategy in the way it reacts to the closest neighbour. In the Voronoi based approach, the evader tries to maximize the safe-reachable area based on only the position of each player. In the safe-reachable area based approach, the evader tries to maximize the safe-reachable area based on both the position and heading of each player. In the cooperative PPN algorithm, the capture time depends on attaining the desired collision formation geometry. As we have considered the same manoeuvring capabilities for pursuers and evader, it takes a little longer time to capture. In spite of this, the proposed algorithm performs well and is able to capture the evader.

Next, we evaluate the performance of the inver-PPN (evading) strategy against the Voronoi and cooperative PPN capturing approaches. For this purpose, we again carried out 20 runs and summarized the results in Table 2. It can be seen from the table that when the number of pursers is low, the cooperative PPN performs better as a pursuer strategy. However, when the

Table 2. Simulation results, inverse-PPN versus different capturing strategies

Team size	Over 20 runs					
	Voronoi			Cooperative PPN		
	Min	Max	Avg	Min	Max	Avg
2	22.1	480.9	100.3	16.3	766.8	128.6
3	12.6	135.1	46.8	25.1	208.0	62.4
4	2.4	90.7	39.9	10.6	216.0	61.9
5	21.4	249.9	55.0	9.3	86.8	37.4

number of pursuers is set to five, the Voronoi approach becomes effective. This is because the Voronoi approach works in a decentralized manner and in a small region an evader would not have a place to escape. In the cooperative PPN approach, all agents works similarly. Hence, we need a combination of the safe-reachable area and cooperative PPN approaches to capture an evader effectively. We will follow this up in our future work.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented cooperative algorithms for a pursuit-evasion game in which multiple pursuers are trying to capture a single evader. The game is considered in a two dimensional bounded connected plane and all players are subject to the same speed and turn radius constraints. We have initially presented a simple capturing strategy for holonomic systems based on the idea of minimizing safe-reachable areas, similar to the work presented in Huang et al. (2011), and using a pursuit guidance law. Next, we have extended this idea to non-holonomic systems and proposed an efficient algorithm for capturability. Our approach discretizes the domain to compute approximate areas efficiently and then employs ideas from model predictive control to compute the solution. The control inputs are parameterized with a constant value over the horizon. As the performance of the approach depends on the discretization of the domain and the discrete control inputs, we have proposed another approach based on the pure proportional navigation (PPN) law. The cooperative PPN algorithm tries to capture the evader by keeping the rate of rotation of the line-of-sight (LOS) vector close to zero in each pursuer's body frame. The performance of the algorithm is validated in numerical simulations. In future, we will try to fuse both approaches to get a better capturing strategy. Moreover, we will extend the work to the case where the state of the evader is not directly available to all the pursuers.

REFERENCES

Belkhouche, F., Belkhouche, B., and Rastgoufard, P. (2005). Multi-robot hunting behavior. In *IEEE International Conference on Systems, Man and Cybernetics*, 2299–2304. IEEE.

Berkovitz, L.D. (1975). Two person zero sum differential games: An overview. In J.D. Grote (ed.), *The theory and Application of Differential Games*, 12–22. D. Reidel.

Dubins, L.E. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics*, 79(3), 497–516.

Gavrilova, M. (2008). *Generalized voronoi diagram: a geometry-based approach to computational intelligence*, volume 158. Springer.

Huang, H., Zhang, W., Ding, J., Stipanovic, D.M., and Tomlin, C.J. (2011). Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers. In *IEEE Conference on*

Decision and Control and European Control Conference, 4835–4840.

Isaacs, R. (1965). *Differential Games: A Mathematical Theory with Application to Warfare and Pursuit Control and Optimization*. Dover Publications Inc.

Isler, V., Kannan, S., and Khanna, S. (2005). Randomized pursuit-evasion in a polygonal environment. *IEEE Transactions on Robotics*, 21(5), 875–884.

Jeon, I.S., Lee, J.I., and Tahk, M.J. (2010). Homing guidance law for cooperative attack of multiple missiles. *Journal of Guidance, Control, and Dynamics*, 33(1).

Shneydor, N.A. (1998). *Missile Guidance and Pursuit (Kinematics, Dynamics and Control)*. Horwood Publishing Limited.

Zarchan, P. (2002). *Tactical and strategic missile guidance*. Progress in astronautics and aeronautics. AIAA.