GREEDY CONTROL FOR HYBRID PURSUIT GAMES

Maria Prandini[§] **João P. Hespanha**[†] **George J. Pappas**[‡]

[§]Dept. of Electronics for Automation, University of Brescia
Fax: +39 (030) 380014 - E-mail: prandini@ing.unibs.it

[†]Dept. of Electrical Engineering–Systems, University of Southern California
Fax: +1 (213) 821-1109 - E-mail: hespanha@usc.edu

[‡]Dept. of Electrical Engineering, University of Pennsylvania
Fax: +1 (215) 573-2068 - E-mail: pappasg@ee.upenn.edu

Keywords: pursuit games, hybrid systems, greedy control, partial information Markov games.

Abstract

We address the design of optimal strategies for a pursuer trying to catch a moving evader. When the pursuer has available two teams of agents with different capabilities –one that can "search for the evader" and the other one that can "catch the evader"–, the game can be naturally formulated as an optimal control problem on a hybrid system. We show that solving the hybrid pursuit game is equivalent to finding a Stackelberg equilibrium solution for a partial information Markov game, which can be solved using dynamic programming. Since for most realistic situations this approach is computationally very difficult, we propose a two-level suboptimal solution that uses a *greedy* control for coordinating the agents within each team, and a threshold-based logic for orchestrating the switching between teams. Simulations are included to show the feasibility of the approach.

1 Introduction

In this paper, we consider games where a pursuer is moving in some region trying to catch (or more generally handle in some way) an evader. The pursuer has a finite amount of time to accomplish its mission. Possible scenarios for such pursuit games includes search and rescue operations, localize and retrieve parts in a warehouse, localize and neutralize environmental threats, search and capture missions, etc. In some of these problems the evader is approximately moving randomly (e.g., search and rescue operations), whereas in other ones it is actively avoiding detection (e.g., search and capture missions).
There is a large body of literature dealing with pursuit games. The reader is referred for example to the textbook [1]. For a formulation of pursuit games that takes visual occlusion into account, see, e.g., [9]. The search and rescue problems [17, 5] are also closely related to the pursuit games addressed here.

We deal with 'structured' games in which the pursuer has available different resources for performing different operations. These operations cannot be executed simultaneously, therefore the pursuer must decide their sequencing to accomplish its mission. Executing a mission then involves *i)* choosing a strategy for coordinating the agents belonging to the same team during each mode of operation, and *ii)* orchestrating the switching between the different modes. In our formulation, the performance achieved by a strategy is measured by an index taking into account the operations costs and the time to accomplish the mission.

In Section 2, the game is formulated as a controller synthesis problem for a hybrid system, hence its denomination "*hybrid pursuit game*". The state of the hybrid system has two components: a *discrete state* representing the mode of operation, and a *continuous state* representing the positions of pursuer and evader. The transitions between the discrete modes are always enabled and actually occur when selected by a *discrete input* corresponding to a switching command. As for the evolution of the continuous state, the evader's motion is assumed to be open-loop, whereas the pursuing agents' motion depends on an applied *continuous input* as well as on the value of the discrete state. Solving the game means designing a feedback controller for the pursuer that, at each time instant, decides which discrete and continuous inputs should be applied to the system so that the performance index is minimized. These decisions are based on noisy measurements of the state of the system.
In general, partial information stochastic games are poorly understood and the literature is relatively sparse. Notable exceptions are games with lack of information for one of the player [10, 16] and games with particular structures such as the Rabbit and Hunter game [3], the Searchlight game [11].
Here, we avoid some of the inherent difficulties of partial information games by assuming a known open-loop policy for the evader. The resulting problem can then be viewed as a Stackelberg equilibrium in which the evader has announced its policy. In Section 3, we show how the described hybrid pursuit game can be solved using dynamic programming. This approach, however, becomes computationally unfeasible as the dimension of the problem grows ("curse of dimensionality", [2]). In Section 4, we then propose a suboptimal hierarchical solution where first the control for performing each single operation is designed, and then a switching rule between operation

modes is selected. The use of a hierarchical decomposition that partitions the problem in subproblems of smaller dimension is not new in the literature and it has been proposed, for example, in [4] and [12]. In our case, however, the decomposition is generated by the nature of the problem itself rather then being artificially imposed through a somewhat arbitrary partition of the state space.
Simulation results show the feasibility and performance of the proposed suboptimal solution.

Notation: $(\Omega, \mathcal{F})$ denotes the relevant *measurable space*. Bold face symbols are used for random variables. Given a probability measure $\mathrm{P} : \mathcal{F} \to [0,1]$ and a random variable $\boldsymbol{\xi} : \Omega \to \mathcal{Z}$, $\mathrm{P}(\boldsymbol{\xi} = z)$ is the probability of $\boldsymbol{\xi}$ taking the value $z \in \mathcal{Z}$. Moreover, $\mathrm{E}[\boldsymbol{\xi}|B]$ is the expected value of $\boldsymbol{\xi}$ conditioned to an event $B \in \mathcal{F}$. Given a set $\mathcal{C}$, we denote by $\mathcal{P}(\mathcal{C})$ the family of all probability distributions on $\mathcal{C}$, and by p_c the probability of $c \in \mathcal{C}$ in the distribution $p \in \mathcal{P}(\mathcal{C})$.

2 Hybrid 2-modes pursuit game

We consider a game where the pursuer has available two *modes of operation* to accomplish its mission: it can either *search* the evader or *capture* it once found. Depending on the particular context, "capture" may actually mean handling the evader in some particular way (for example rescuing it). The resources available to the pursuer for executing the two operations are:

- a team of n_S agents for the search operation. Typically, each *searching agent* can only move slowly but it is capable of sensing the surrounding region for the evader;
- a single agent for the capture operation. Typically, the *capturing agent* can move fast and is appropriately equipped for executing the capture operation, but it has poor sensors.

We assume that the search and capture operations cannot be done at the same time. This would happen, for example, when keeping the capturing agent moving around while the evader has not yet been detected is expensive or may cause the equipment to be damaged. The pursuer must then appropriately switch between the two operation modes. The game ends when the capture operation is successfully performed.

We describe next the game in terms of a hybrid system optimal control problem. The proposed model differs from those commonly adopted in the literature (see, e.g., [15, 18]), in that it is stochastic. The probabilistic embedding is useful for modeling different sources of uncertainty affecting the system, e.g., actuators/sensors inaccuracy.
We assume that the game is quantized both in time and in space. All events take, in fact, place on a set of equally spaced event times $\mathcal{T} := \{0, 1, \ldots, T\}$, where $T < \infty$ is the duration of the game, and the pursuit region consists of a finite collection $\mathcal{R}$ of cells. The system is hybrid in that: *i)* each player's position evolves under the influence of a control input that is applied at every time instant, whereas *ii)* the transitions between modes are determined by an event-driven switching input.

The discrete state component of the hybrid system represents the operation mode. It takes values in the set $\mathcal{Q} := \{q_S, q_C, q_{\text{over}}\}$, where q_S and q_C respectively denote the search and capture mode, and q_{over} is a mode introduced to represent the game-over condition. The continuous state component of the hybrid system corresponds to the players' position. The pursuer's position takes values in $\mathcal{R}^{n_S}$ when the mode is q_S, and in $\mathcal{R}$, when the mode is either q_C or q_{over}. The evader's position takes values in $\mathcal{R}$, irrespectively of the operation mode. For ease of reference, in the sequel we shall refer to the pursuer and the evader as U and D, respectively.

Definition 1 (probabilistic hybrid system) A *probabilistic hybrid system model* for the game is a 7-tuple $\mathcal{H}_{\mathcal{G}} = (s, w, y, p^0, e, f, h)$, where

- $s := (q, x_U, x_D)$ is the *state*, with q the discrete state variable and $x = (x_U, x_D)$ the continuous state variable. $s_U = (q, x_U)$ represents the operation mode and the pursuer's position, and x_D the evader's position. The state space is $\mathcal{S} := \mathcal{S}_U \times \mathcal{X}_D$, where $\mathcal{S}_U := \{q_S\} \times \mathcal{R}^{n_S} \cup \{q_C, q_{\text{over}}\} \times \mathcal{R}$, and $\mathcal{X}_D := \mathcal{R}$.
- $w := (u_q, u_x, d)$ is the *input*, with u_q the discrete input variable and $w_x = (u_x, d)$ the continuous input variable. $u = (u_q, u_x)$ is under the control of the pursuer and d under the control of the evader. The input set $\mathcal{W} = \mathcal{U} \times \mathcal{D}$, where $\mathcal{U} = \mathcal{U}_q \times \mathcal{U}_x$ with $\mathcal{U}_q = \{q_C, q_S\}$, is assumed finite.
- $y := (y_q, y_{x_U}, y_D)$ is the *output*, with y_q the discrete output variable and $y_x = (y_{x_U}, y_D)$ the continuous output variable. $y_U = (y_q, y_{x_U})$ represents the observation available to the pursuer on the operation mode and its own position, and y_D the observation on the evader's position. The output set $\mathcal{Y} = \mathcal{Y}_U \times \mathcal{Y}_D$ is assumed finite.
- $p^0 \in \mathcal{P}(\mathcal{S})$ is the *a-priori distribution* of the state.
- $e : \mathcal{S} \times \mathcal{U}_q \times \mathcal{D} \to \mathcal{P}(\mathcal{S})$ is the *discrete transitions probability map* governing the transitions between modes.
- $f : \mathcal{S} \times \mathcal{U}_x \times \mathcal{D} \to \mathcal{P}(\mathcal{X})$, where $\mathcal{X} := \mathcal{X}_U \times \mathcal{X}_D$ with $\mathcal{X}_U := \mathcal{R}^{n_S} \cup \mathcal{R}$, is the *continuous transitions probability map* governing the evolution of the continuous state within each mode.
- $h : \mathcal{S} \to \mathcal{P}(\mathcal{Y})$ is the *output probability distribution map.*

When $\mathcal{Y} = \mathcal{S}$ and $h_y(s) = 1$, iff $y = s$, for all $s \in \mathcal{S}$, then $\mathcal{H}_{\mathcal{G}}$ is said to be a *full-state information* hybrid system. This models the case when the pursuer has perfect state-measuring sensors. When this does not happen, the hybrid system is said to be *partial-state information.*
We have set $\mathcal{U}_q = \{q_C, q_S\}$ with the understanding that if the pursuer is in mode q_C and wants to switch to mode q_S, it applies $u_q = q_S$. Viceversa for switching from q_S to q_C. As for the transition to q_{over}, it is taken when the state belongs to a certain set $\mathcal{S}_{\text{over}} := \{q_C\} \times \mathcal{X}_{\text{over}}$,

which models a successful capturing operation. Here, we set $\mathcal{X}_{\text{over}} = \{(x_U, x_D) \in \mathcal{X} : x_D = x_U\}$.

Definition 2 (stochastic execution) A stochastic process $(\mathbf{s}, \mathbf{w}, \mathbf{y})$ is a *stochastic execution* of $\mathcal{H}_\mathcal{G}$ if, for all $t \in \mathcal{T}$,

- the random variables $\mathbf{s}(t)$, $\mathbf{w}(t)$, and $\mathbf{y}(t)$ take values in $\mathcal{S}$, $\mathcal{W}$, and $\mathcal{Y}$, respectively.
- $\mathbf{s}(t+1)$ is conditionally independent of all other random variables at times smaller or equal to t, given $\mathbf{s}(t)$ and $\mathbf{w}(t)$. Moreover, for all $s' = (q', x')$, $s = (q, x) \in \mathcal{S}$, $w = (u_q, u_x, d) \in \mathcal{W}$,

$$\mathrm{P}\left(\mathbf{s}(t+1) = s' \,|\, \mathbf{s}(t) = s, \mathbf{w}(t) = w\right) = p(s, s', w),$$

where $p : \mathcal{S} \times \mathcal{S} \times \mathcal{W} \to [0, 1]$ is given by

$$p(s, s', w) = \tag{1}$$
$$\begin{cases} 1, \ q = q_{\text{over}} \wedge s' = s \\ 1, \ s \in \mathcal{S}_{\text{over}} \wedge q' = q_{\text{over}} \wedge x' = x \\ e_{s'}(s, u_q, d), \ u_q \neq q \wedge q \neq q_{\text{over}} \wedge s \notin \mathcal{S}_{\text{over}} \\ f_{x'}(s, u_x, d), \ u_q = q = q' \wedge q \neq q_{\text{over}} \wedge s \notin \mathcal{S}_{\text{over}} \\ 0, \ \text{otherwise.} \end{cases}$$

For clarity of notation we shall write $p(s \xrightarrow{ud} s')$ for $p(s, s', (u, d))$.

- $\mathbf{s}(0)$ is independent of all the other random variables at time 0, and it has probability distribution p^0.
- $\mathbf{y}(t)$ is conditionally independent of all the other random variables at times smaller or equal to t, given $\mathbf{s}(t)$. Moreover, for all $s \in \mathcal{S}$, $y \in \mathcal{Y}$,

$$\mathrm{P}\left(\mathbf{y}(t) = y \,|\, \mathbf{s}(t) = s\right) = h_y(s). \tag{2}$$

In this paper, we assume that, when a transition between modes is taken, the continuous state component of the system is subject to a *reset condition*. In particular, the evader's position evolves irrespectively of the fact that the transition is taken, whereas the searching and capturing agents positions are reset to a fixed position $b \in \mathcal{R}$ representing, for example, the position of the base from which the pursuer coordinates the mission. This corresponds to a discrete transitions probability map of the form: For $s' = (q', x'_U, x'_D) \in \mathcal{S}$, $x = (x_U, x_D) \in \mathcal{X}$, $u_q \in \mathcal{U}_q$, $d \in \mathcal{D}$,

$$e_{s'}((q_S, x), u_q, d) = \begin{cases} \epsilon_{x'_D}(x_D, d), \ u_q = q' = q_C \wedge x'_U = b \\ 0, \ \text{otherwise;} \end{cases}$$

$$e_{s'}((q_C, x), u_q, d) =$$
$$\begin{cases} \epsilon_{x'_D}(x_D, d), \ u_q = q' = q_S \wedge x \notin \mathcal{X}_{\text{over}} \wedge x'_U = b^{n_S} \\ 0, \ \text{otherwise,} \end{cases}$$

where $\epsilon : \mathcal{X}_D \times \mathcal{D} \to \mathcal{P}(\mathcal{X}_D)$ essentially corresponds to the evader's transitions probability function.
In order to model the fact that the players can independently control their own positions when the game is not over, f is defined as follows: For $s = (q, x_U, x_D) \in \mathcal{S}$, $x' = (x'_U, x'_D) \in \mathcal{X}$, $u_x \in \mathcal{U}_x$, $d \in \mathcal{D}$,

$$f_{x'}(s, u_x, d) = \begin{cases} \phi_{x'_U}(x_U, u_x)\epsilon_{x'_D}(x_D, d), & q = q_S \\ \varphi_{x'_U}(x_U, u_x)\epsilon_{x'_D}(x_D, d), & q = q_C \\ 0, \ \text{otherwise.} \end{cases}$$

where $\varphi : \mathcal{R} \times \mathcal{U}_x \to \mathcal{P}(\mathcal{R})$, and $\phi : \mathcal{R}^{n_S} \times \mathcal{U}_x \to \mathcal{P}(\mathcal{R}^{n_S})$. The maps φ, ϕ, and ϵ can be chosen so as to model different motion capabilities of the capturing/searching agents and the evader.
Figure 1 represents the hybrid system $\mathcal{H}_\mathcal{G}$.

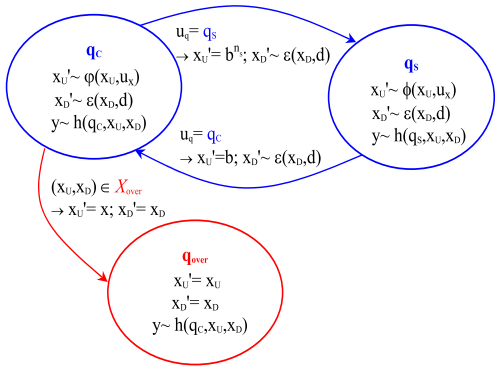


Figure 1: Probabilistic hybrid system model of the game.

We consider the case when the pursuer knows perfectly the operation mode and its agents' positions, but not the evader's location and in fact uses the search team to explore the pursuit region. This can be modeled by setting $\mathcal{Y}_U = \mathcal{S}_U$ and defining h as follows: For $y = (y_U, y_D) \in \mathcal{Y}$, $s = (s_U, x_D) \in \mathcal{S}$,

$$h_y(s) = \begin{cases} \eta_{y_D}(s), & y_U = s_U \\ 0, \ \text{otherwise,} \end{cases}$$

where the map $\eta : \mathcal{S} \to \mathcal{P}(\mathcal{Y}_D)$ and the set $\mathcal{Y}_D$ can be chosen so as to model the different sensing capabilities of the capturing and searching agents.

Closed-loop control policies

Given a hybrid system $\mathcal{H}_\mathcal{G}$, we want to impose a certain behavior to its stochastic executions $(\mathbf{s}, (\mathbf{u}, \mathbf{d}), \mathbf{y})$. In order to achieve this, we can use the input $\mathbf{u}$, which we hence call the *control*, while the input $\mathbf{d}$ acts as a *disturbance*. $\mathbf{u}$ and $\mathbf{d}$ are in fact the inputs of the pursuer and the evader, respectively. Here, we assume that the evader moves according to a known open-loop policy. For simplicity, we take this policy to be stationary, characterized by a distribution $\delta \in \mathcal{P}(\mathcal{D})$. The pursuer, however, can use a feedback controller. We proceed by stating precisely what we mean by this.
Fix a time instant $t \in \mathcal{T}$ and consider the sequence:

$$\mathbf{Y}_t := \{\mathbf{y}(0), \mathbf{u}(0), \mathbf{y}(1), \mathbf{u}(1), \ldots, \mathbf{y}(t-1), \mathbf{u}(t-1), \mathbf{y}(t)\},$$

which is said to be of *length* t. $\mathbf{Y}_t$ represents the information available to the pursuer at time t for deciding which action to take. We denote by $\mathcal{Y}^*$ the set of all possible outcomes Y for $\mathbf{Y}_t$, $t \in \mathcal{T}$, and by $\ell(Y)$ the length of $Y \in \mathcal{Y}^*$. A *feedback controller* is then a map $\mu : \mathcal{Y}^* \to \mathcal{U}$. We denote by Π_U the set of all the feedback controllers.

Definition 3 (closed-loop stochastic execution) Given $\delta \in \mathcal{P}(\mathcal{D})$, a stochastic execution $(\mathbf{s}, (\mathbf{u}, \mathbf{d}), \mathbf{y})$ of $\mathcal{H}_\mathcal{G}$ is said to be *closed-loop* controlled by $\mu \in \Pi_U$ if it satisfies the following conditions, for all $t \in \mathcal{T}$,

- $\mathbf{u}(t)$ is conditionally independent of all other random variables at times smaller or equal to t, given $\mathbf{Y}_t$. Moreover, for all $Y \in \mathcal{Y}^*$, with $\ell(Y) = t$,

$$\mathrm{P}_\mu(\mathbf{u}(t) = u \mid \mathbf{Y}_t = Y) = \begin{cases} 1, & \text{if } \mu(Y) = u \\ 0, & \text{otherwise} \end{cases}, \quad u \in \mathcal{U}.$$

- $\mathbf{d}(t)$ is independent of all other random variables at times smaller or equal to t, and $\mathrm{P}_\delta(\mathbf{d}(t) = d) = \delta_d$, $d \in \mathcal{D}$.

Remark: In general, for each pair (μ, δ) we have a different probability measure $\mathrm{P}_{\mu\delta}$. When an assertion holds true with respect to $\mathrm{P}_{\mu\delta}$ independently of μ, or δ, or both μ and δ, we write P_δ, P_μ, or P, respectively. Similarly for the expectation E. According to this notation, $\mathrm{P}_\mu(\mathbf{u}(t) = u \mid \mathbf{Y}_t = Y)$ and $\mathrm{P}_\delta(\mathbf{d}(t) = d)$ above are respectively independent of δ and μ, whereas equations (1) and (2) and the distribution of $\mathbf{s}(0)$ are independent of both μ and δ.

Given $\delta \in \mathcal{P}(\mathcal{D})$, we aim at designing a controller $\mu \in \Pi_U$ such that the closed loop stochastic execution $(\mathbf{s}, (\mathbf{u}_q, \mathbf{u}_x, \mathbf{d}), \mathbf{y})$ of $\mathcal{H}_\mathcal{G}$ controlled by μ reaches the game-over state in minimum time, with some added cost k_C for each unsuccessful attempt at capturing the evader, and an added reward k_{over} for a successful capture. This translates into selecting $\mu \in \Pi_U$ that minimizes

$$J_{\mu\delta} = \mathrm{E}_{\mu\delta}\big[\sum_{t=0}^{T} c(\mathbf{s}(t), \mathbf{u}_q(t))\big], \tag{3}$$

where, for all $s = (q, x) \in \mathcal{S}$, $u_q \in \mathcal{U}_q$,

$$c(s, u_q) = \begin{cases} 0, & q = q_{\text{over}} \\ -k_{\text{over}}, & s \in \mathcal{S}_{\text{over}} \\ k_C, & u_q = q_C \\ 1, & \text{otherwise.} \end{cases}$$

3 Dynamic programming solution

Consider the hybrid pursuit game modeled by $\mathcal{H}_\mathcal{G}$. Given $\delta \in \mathcal{P}(\mathcal{D})$, we show next that there exists a solution to the game, i.e., a feedback controller

$$\mu^* \in \Pi_U \text{ such that } J_{\mu^*\delta} = \inf_{\mu \in \Pi_U} J_{\mu\delta}, \tag{4}$$

which can be computed using dynamic programming. Due to space limitations, all the results are stated without proof (see [13] for details).

It is straightforward to check that a stochastic execution $(\mathbf{s}, (\mathbf{u}, \mathbf{d}), \mathbf{y})$ of $\mathcal{H}_\mathcal{G}$ is an instance of a partial information Markov game with state $\mathbf{s}$, inputs $\mathbf{u}$ and $\mathbf{d}$, and observations $\mathbf{y}$ ([7]). In the case considered here, player U is pursuing the objective of catching player D, whereas D uses a fixed static policy δ. Hence, we can regard player D as the 'leader' of the game with value (3), which declares its policy δ to player U (the 'follower'). Player U then does its best to counteract the leader's choice by selecting its policy μ so as to minimize (3). In this sense solving the optimal control problem (4) is equivalent to finding a Stackelberg equilibrium for a partial information Markov game with value (3). In the sequel, we shall consider a fixed $\delta \in \mathcal{P}(\mathcal{D})$.

For a given policy $\mu \in \Pi_U$, we define $V^U_{\mu\delta}(Y)$, $Y \in \mathcal{Y}^*$, to be *player U's cost-to-go from* Y associated with the policy μ, after having collected a sequence Y of observations and controls of length $t := \ell(Y) \in \mathcal{T}$, i.e., $\mathrm{E}_{\mu\delta}\big[\sum_{\tau=t}^{T} c(\mathbf{s}(\tau), \mathbf{u}_q(\tau)) \,\big|\, \mathbf{Y}_t = Y\big]$. This expected value is only well defined when $\mathrm{P}_{\mu\delta}(\mathbf{Y}_t = Y) \neq 0$ but it is actually convenient to define cost-to-go for any $Y \in \mathcal{Y}^*$ such that there is some policy $\hat{\mu}_Y$ for which $\mathrm{P}_{\hat{\mu}_Y\delta}(\mathbf{Y}_t = Y) \neq 0$. To do this we formally define

$$V^U_{\mu\delta}(Y) := \mathrm{E}_{\tilde{\mu}_Y\delta}\Big[\sum_{\tau=t}^{T} c(\mathbf{s}(\tau), \mathbf{u}_q(\tau)) \,\Big|\, \mathbf{Y}_t = Y\Big],$$

where the policy $\tilde{\mu}_Y$ is given by $\tilde{\mu}_Y(\bar{Y}) = \mu(\bar{Y})$, if $\ell(\bar{Y}) \geq \ell(Y)$, and $\tilde{\mu}_Y(\bar{Y}) = \hat{\mu}_Y(\bar{Y})$, if $\ell(\bar{Y}) < \ell(Y)$, $\bar{Y} \in \mathcal{Y}^*$ (cf. [13]). The intuition behind this is that once $\mathbf{Y}_t = Y$, it does not really matter what was the value of the policy before time t. So we might as well set the value of the cost-to-go from Y, for a policy $\mu \in \Pi_U$ for which $\mathrm{P}_{\mu\delta}(\mathbf{Y}_t = Y) = 0$, to be identical to that of any policy $\tilde{\mu}_Y$ taking the same actions as μ from t on, but for which $\mathrm{P}_{\tilde{\mu}_Y\delta}(\mathbf{Y}_t = Y) \neq 0$.
The cost $J_{\mu\delta}$ associated with $\mu \in \Pi_U$ can be easily computed from $V^U_{\mu\delta}$. Indeed, $J_{\mu\delta} = \mathrm{E}\big[V^U_{\mu\delta}(\{\mathbf{y}(0)\})\big]$.
We shall see next that it is possible to compute $V^U_{\mu\delta}$ using the operator $T^U_{\mu\delta}$ from the set of functionals $\mathcal{V}^U := \{V : \mathcal{Y}^* \to \mathbb{R}\}$ into itself, defined by

$$T^U_{\mu\delta}V(Y) := \mathrm{E}_{\tilde{\mu}_Y\delta}\big[c\big(\mathbf{s}(t), \mathbf{u}_q(t)\big) + V(\mathbf{Y}_{t+1}) \,\big|\, \mathbf{Y}_t = Y\big],$$

$Y \in \mathcal{Y}^*$, $t := \ell(Y)$, where $V(\mathbf{Y}_{T+1}) := 0$.

Proposition 1 *For each $\mu \in \Pi_U$, $V^U_{\mu\delta}$ is the unique solution to $V^U_{\mu\delta} = T^U_{\mu\delta}V^U_{\mu\delta}$.*

We proceed by showing how to actually compute the function in $\mathcal{V}^U$ that results from applying $T^U_{\mu\delta}$ to some function $V \in \mathcal{V}^U$. For each $u \in \mathcal{U}$, we can define an operator $H^U_{u\delta}$ from $\mathcal{V}^U$ into itself by setting for each $Y \in \mathcal{Y}^*$:

$$H^U_{u\delta}V(Y) := \sum_{s,s',y,d} \big(c(s, u_q) + V\big(\{Y, u, y\}\big)\big)h_y(s') \; p(s \xrightarrow{ud} s')\delta_d I_\delta(s, Y),$$

with $V\big(\{Y,u,y\}\big) = 0$ if $\ell(Y) = T$. The function $I_\delta : \mathcal{S} \times \mathcal{Y}^* \to \mathbb{R}$ is defined recursively by

$$I_\delta(s',\{Y,u,y\}) = \frac{\sum_{s,d} h_y(s')p(s \xrightarrow{ud} s')\ \delta_d\ I_\delta(s,Y)}{\sum_{\bar{s},\bar{d},\bar{s}'} h_y(\bar{s}')p(\bar{s} \xrightarrow{u\bar{d}} \bar{s}')\ \delta_{\bar{d}}\ I_\delta(\bar{s},Y)},$$

$s' \in \mathcal{S}, u \in \mathcal{U}, d \in \mathcal{D}, y \in \mathcal{Y}$, and initialized with $I_\delta(s',\{y\}) = \frac{h_y(s')p^0_{s'}}{\sum_{\bar{s}'} h_y(\bar{s}')p^0_{\bar{s}'}}$.
It can be shown that, for each $\mu \in \Pi_U$ such that $\mathrm{P}_{\mu\delta}(\boldsymbol{Y}_\tau = Y) > 0$, $I_\delta(s,Y) = \mathrm{P}_{\mu\delta}(\boldsymbol{s}(\tau) = s \mid \boldsymbol{Y}_\tau = Y)$, $s \in \mathcal{S}$, i.e., I_δ is the so-called *information state* ([8]). Moreover,

$$T^U_{\mu\delta} V(Y) = H^U_{\mu(Y)\delta} V(Y), \qquad Y \in \mathcal{Y}^*.$$

Proposition 2 *Let $\mu^* \in \Pi_U$ be a policy such that*

$$T^U_{\mu^*\delta} V^U_{\mu^*\delta}(Y) = \min_{u\in\mathcal{U}} H^U_{u\delta} V^U_{\mu^*\delta}(Y), \qquad Y \in \mathcal{Y}^*. \tag{5}$$

Then, $J_{\mu^*\delta} = \inf_{\mu\in\Pi_U} J_{\mu\delta}$.

By Propositions 1 and 2, an optimal feedback controller $\mu^* \in \Pi_U$ with associated cost $J_{\mu^*\delta} = E[V^U_{\mu^*\delta}(\mathbf{y}(0))]$ can be constructed as follows: for every $t \in \mathcal{T}$, and for every $Y \in \mathcal{Y}^*$ with $\ell(Y) = t$,

1. set $\mu^*(Y) = \arg\min_{u\in\mathcal{U}} H^U_{u\delta} V^U_{\mu^*\delta}(Y)$,
2. set $V^U_{\mu^*\delta}(Y) = H^U_{\mu^*(Y)\delta} V^U_{\mu^*\delta}(Y)$,

starting at $t = T$ and going backwards in time.

Remark: In general, the number of elements in $\mathcal{Y}^*$ that needs to be considered in the procedure above is equal to $\sum_{t\in\mathcal{T}} n_y^{t+1} n_u^t = \frac{n_y^{T+2} n_u^{T+1} - n_y}{n_y n_u - 1}$, n_y and n_u being the cardinality of $\mathcal{Y}$ and $\mathcal{U}$, respectively. Though one can reduce this number by exploiting the structure of the problem at hand, (for example, in our case, the optimal controller μ^* can select an arbitrary $u \in \mathcal{U}$ for all those $Y \in \mathcal{Y}^*$ revealing that at some instant the discrete state value is q_{over}), the problem remains computationally very difficult in most realistic situations.

4 Hierarchical greedy solution

In this section, we propose a solution to the described hybrid pursuit game which is based on a two step design process. In particular, we suggest a *greedy control* of the agents' motion within both the search and capture modes, and a *threshold-based logic* for switching between them. This leads to a suboptimal controller $\mu \in \Pi_U$, but it makes the problem computationally very attractive.

The greedy control applied within mode $q \in \{q_S, q_C\}$ consists of directing at each time instant $t \in \mathcal{T}$ the agent(s) to the locations that maximize the probability of finding the evader at $t+1$, conditional to the information $Y \in \mathcal{Y}^*$, $\ell(Y) = t$, collected up to time t. This means that, if $Y = \{\bar{Y}, u, (q, x_U, y_D)\}$, and no condition for switching is satisfied, then $\mu(Y)$ is equal to (u_q, u_x) with $u_q = q$ and

$$u_x = \arg\max_{u_x \in \mathcal{U}_x} \mathrm{P}_{\mu\delta}\left(\exists i : \mathbf{x}_D(t+1) = \mathbf{x}^i_U(t+1) \big| \mathbf{Y}_t = Y\right).$$

Note that

$$\mathrm{P}_{\mu\delta}\left(\exists i : \mathbf{x}_D(t+1) = \mathbf{x}^i_U(t+1) \big| \mathbf{Y}_t = Y\right) = \begin{cases} \sum_{\bar{x}_D} \varphi_{\bar{x}_D}(x_U, u_x) m_\delta(\bar{x}_D, Y), & u_q = q_C \\ \sum_{\bar{x}_D, \bar{x}_U \text{ s.t. } \exists i:\ \bar{x}^i_U = \bar{x}_D} \phi_{\bar{x}_U}(x_U, u_x) m_\delta(\bar{x}_D, Y), & u_q = q_S \end{cases}$$

where $m_\delta(\bar{x}_D, Y) := \mathrm{P}_{\mu\delta}(\mathbf{x}_D(t+1) = \bar{x}_D \,|\, \mathbf{Y}_t = Y)$, $\bar{x}_D \in \mathcal{R}$, can be computed as follows:

$$m_\delta(\bar{x}_D, Y) = \sum_{d, x_D} \epsilon_{\bar{x}_D}(x_D, d)\ \delta_d\ I_\delta\big((q, x_U, x_D), Y\big).$$

As for the switching logic, the pursuer switches at time $t \in \mathcal{T}$ from the search to the capture mode when the maximum of $m_\delta(x_D, \mathbf{Y}_t)$, $x_D \in \mathcal{R}$, (which is the probability of the evader being at its most likely location) exceeds a certain threshold. If the capture operation is successful, then the game is over and the pursuer won it. Otherwise, the pursuer switches back to the search mode so as to collect more observations and eventually switch back to the capture mode. If the time horizon allowed to perform the mission elapses and the evader is not caught, then the game is over and the pursuer lost it. One then should set the threshold so as to optimize the value of the cost (3).

We consider next a specific game to which we apply the proposed approach. In this game the pursuit takes place in a rectangular grid with $n_r \times n_c$ square cells. The cell at row i and column j is identified by the vector (i,j). Hence, $\mathcal{R} := \{(i,j) : i = 1,\dots,n_r,\ j = 1,\dots,n_c\}$, and $\mathcal{R}^{n_S}$ is the set of all ordered n_S-tuple of elements in $\mathcal{R}$. The evader is 'slow', in the sense that, in a single time step, it can only move to a cell in the set $\mathcal{A}(x) \subseteq \mathcal{R} \setminus \{x\}$ of cells adjacent to its present position $x \in \mathcal{R}$. Let $Adj := \{-1,0,1\} \times \{-1,0,1\}$. Then this can be modeled by setting $\mathcal{D} := Adj$ and

$$\epsilon_{x'_D}(x_D, d) = \begin{cases} 1, & x'_D = x_D + d \in \mathcal{A}(x_D) \\ 1, & x'_D = x_D \wedge x_D + d \notin \mathcal{A}(x_D) \\ 0, & \text{otherwise,} \end{cases}$$

$d \in \mathcal{D}$, $x_D, x'_D \in \mathcal{R}$. As for the pursuer, the capturing agent is 'fast' in that it can move from any cell to any other cell in a single time step, whereas, similarly to the evader, the searching agents are slow. This is modeled by setting $\mathcal{U}_x := Adj^{n_S} \cup \mathcal{R}$, and defining $\varphi_{x'_U}(x_U, u_x) = 1$, if $u_x = x'_U \in \mathcal{R}$, and 0 otherwise, $x_U, x'_U \in \mathcal{R}$, while the definition of ϕ is similar to that of ϵ.

We suppose that the information the searching agents report regarding the presence of the evader in the cell they are occupying is accurate, whereas there is a nonzero probability that a searching agent reports the presence of an evader in a cell adjacent to its current position, when there is no evader in that cell and vice-versa. Specifically, the sensor model is a function of the *probability of false positive* $\nu_p \in [0,1]$, i.e., the probability of detecting an evader

in an adjacent cell, given that none is there, and the *probability of false negative* $\nu_n \in [0,1]$, i.e., the probability of not detecting an evader, given that the evader is there. As for the capturing agent, it has no sensors. The sensing capabilities of the pursuer can then be modeled by setting $\mathcal{Y}_D = 2^{\mathcal{R}}$, where $2^{\mathcal{R}}$ denotes the set of all subsets of $\mathcal{R}$, and defining η as follows: for $y_D \in \mathcal{Y}_D$, and $x = (x_U, x_D) \in \mathcal{X}$,

$$\eta_{y_D}((q_C, x)) = \begin{cases} 1, & y_D = \emptyset \\ 0, & \text{otherwise,} \end{cases}$$

$$\eta_{y_D}((q_S, x)) = \begin{cases} 1, & x_D \in \{x_U^i\}_{i=1}^{n_S} \wedge y_D = \{x_D\} \\ g_\nu(x_U), & x_D \notin \{x_U^i\}_{i=1}^{n_S} \wedge y_D \subseteq \delta\mathcal{A}(x_U) \\ 0, & \text{otherwise,} \end{cases}$$

where $\delta\mathcal{A}(x_U)$ denotes the subset of $\mathcal{A}^{n_p}(x_U) = \mathcal{A}(x_U^1) \times \ldots \times \mathcal{A}(x_U^{n_S})$ not occupied by any searching agent and $g_\nu(x_U) := \nu_p^{k_1}(1-\nu_p)^{k_2}\nu_n^{k_3}(1-\nu_n)^{k_4}$, k_1, k_2, k_3 and k_4 being, respectively, the number of false positives, true negatives, false negatives, and true positives.

In the simulations, $\delta_d = \rho$, $d \in \mathcal{D}\setminus\{(0,0)\}$, $\delta_{(0,0)} = 1-8\rho$, with $\rho \in [0, 1/8]$. We adopt the algorithms in [6], which implement at low computational cost the greedy control in both the cases of constrained and unconstrained motions. Figure 2 refers to a pursuit game with $\rho = 1/20$, $n_S = 3$, and $\mathcal{T} = 30$. The left-hand side represents the cost (3) ($k_{\text{over}} = 50$, $k_C = 10$) as a function of the threshold. The cost is estimated by Monte Carlo simulations starting each game with $\mathbf{x}_U(0) = (q_S, b^{n_S})$, and $\mathbf{x}_D(0)$ extracted at random from the uniform distribution on $\mathcal{R}$. The right-hand side shows a simulation with the threshold set equal to the estimated optimal value. The pursuer at time $t = 6$ is in the search mode with the searching agents (light stars) trying to find the evader (dark circle). The background color of each cell $x \in \mathcal{R}$ encodes the probability $m_\delta(x, \mathbf{Y}_t)$: a light color for low probability and a dark color for high probability. The dark color around cell $(14, 3)$ reveals that the occurrence of a false positive ($\nu_p = \nu_n = 1\%$).

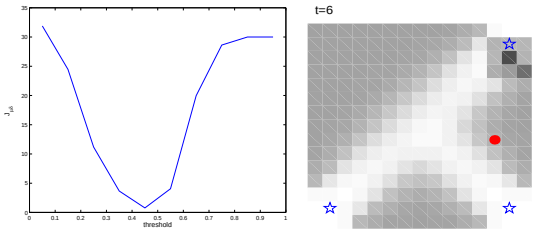


Figure 2: Hybrid pursuit game ($n_r = n_c = 15$, $b = (8,8)$).

5 Conclusions

In this paper we consider a game where a pursuer is trying to catch a moving evader in minimum time and with minimum costs. When the pursuer has available teams of agents capable of performing different operations, but these operations cannot be done simultaneously, the game is naturally formulated as an optimal control problem on a hybrid system. We show how an optimal feedback controller can be computed using dynamic programming in the 2-operations case. Since this approach is computationally very difficult in most realistic situations, we then propose a hierarchical greedy solution.

Acknowledgments

Research supported by Office of Naval Research, Defense Advanced Research Projects Agency, and Ministero dell'Università e della Ricerca Scientifica e Tecnologica.

References

[1] Tamer Başar and Geert Jan Olsder. *Dynamic Noncooperative Game Theory*. Number 23 in Classics in Applied Mathematics. SIAM, Philadelphia, 2nd edition, 1999.

[2] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[3] P. Bernhard, A.-L. Colomb, and G. P. Papavassilopoulos. Rabbit and hunter game: Two discrete stochastic formulations. *Comput. Math. Applic.*, 13(1–3):205–225, 1987.

[4] T. Dean and S. Lin. Decomposition techniques for planning in stochastic domains. In *Proc. of the 14th Int. Joint Conf. on Artificial Intelligence*, 1995.

[5] J.H. Discenza and L.D. Stone. Optimal survivor search with multiple states. *Operations Research*, 29(2):309–323, April 1981.

[6] J.P. Hespanha, H.J. Kim, and S. Sastry. Multiple-agent probabilistic pursuit-evasion games. In *Proc. of the 38th Conf. on Decision and Contr.*, December 1999.

[7] J.P. Hespanha and M. Prandini. Nash equilibria in partial-information games on Markov chains. Submitted to the 40th Conf. on Decision and Contr., March 2001.

[8] P. Kumar and P. Varaiya. *Stochastic Systems: Estimation, Identification and Adaptive Control*. Prentice Hall, Inc. Englewood Cliffs, New Jersey, 1986.

[9] S. M. LaValle and J. Hinrichsen. Visibility-based pursuit-evasion: The case of curved environments. In *Proc. of IEEE Int. Conf. Robot. & Autom.*, 1999.

[10] C. Melolidakis. Stochastic games with lack of information on one side and positive stop probabilities. In Raghavan et al. [14], pages 113–126.

[11] G.J. Olsder and G.P. Papavassilopoulos. About when to use a searchlight. *J. of Mathematical Analysis and Applications*, 136:466–478, 1988.

[12] R. Parr and S. Russell. Reinforcement learning with hierarchies of machines. In *Proc. of the Neural Information Processing Systems Conf.*, 1997.

[13] M. Prandini, J.P. Hespanha, and G.J. Pappas. Greedy control for hybrid pursuit games. Technical report, Dept. of Electronics for Automation, University of Brescia, 2001.

[14] T.E.S. Raghavan, T.S. Ferguson, and T. Parthasarathy, editors. *Stochastic Games and Related Topics: In Honor of Professor L. S. Shapley*, volume 7 of *Theory and Decision Library, Series C, Game Theory, Math. Programming and Operat. Research*. Kluwer Academic Publishers, Dordrecht, 1991.

[15] O. Sharkenia, G.J. Pappas, and S. Sastry. Decidable controller synthesis for a class of linear systems. In N. Lynch and B. H. Krogh, editors, *Hybrid Systems: Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*, pages 407–420. Springer Verlag, 2000.

[16] S. Sorin and S. Zamir. "Big Match" with lack of information on one side (III). In Raghavan et al. [14], pages 101–112.

[17] L.D. Stone. *Theory of Optimal Search*. Academic Press, 1975.

[18] C. Tomlin, J. Lygeros, and S. Sastry. A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE*, 88:949–970, July 2000.