Distributed Hybrid Control for Multiple-Pursuer Multiple-Evader Games *

Michael M. Zavlanos and George J. Pappas

GRASP Laboratory, Department of Electrical and Systems Engineering
University of Pennsylvania
3330 Walnut Street, Philadelphia, PA 19104, USA
Email: {zavlanos,pappasg}@grasp.upenn.edu

Abstract. Multiple-pursuer multiple-evader games raise fundamental and novel problems in control theory and robotics. In this paper, we propose a distributed solution to this problem that simultaneously addresses the discrete assignment of pursuers to evaders as well as the continuous control strategies for capturing individual evaders. The resulting hybrid controller for each pursuer consists of both local coordination protocols guaranteeing that distinct evaders are captured by distinct pursuers, and time-varying potential fields ensuring capturability of single evaders. The overall hybrid system guarantees not only the mutual exclusion property of the assignment for all initial conditions, but also capturing all evaders after exploring at most a polynomial number of discrete assignments. Therefore, our distributed approach dramatically reduces the combinatorial nature of purely discrete assignment problems. Our scalable approach is illustrated with nontrivial computer simulations.

1 Introduction

Multiple-pursuer multiple-evader games consist of tracking and capturing of several evaders by multiple pursuers. Termination of the game occurs when every pursuer has captured an evader, and hence, the capturing condition becomes equivalent to establishing an assignment between pursuers and evaders. If no such a priori assignment information is provided, then the assignment has to be determined on-line. Moreover, if only local information from nearest neighbors is available to the pursuers, the resulting control framework is fully distributed.

For stationary destinations instead of mobile evaders, an on-line approach to the assignment problem is proposed in [1], where the space of permutation invariant multi-robot formations is represented using complex polynomials whose roots correspond to the configurations of the robots in the formation. The proposed approach is open loop and centralized, since it requires global knowledge of the environment. On the other hand, off-line approaches are studied in [2] and [3]. In particular, in [2] a polynomial time algorithm is developed that computes

* This work is partially supported by ARO MURI SWARMS Grant W911NF-05-1-0219 and the NSF ITR Grant 0324977.

a suboptimal assignment between pursuers and destinations based on a "minimum distance to the goal" policy. Any navigation strategy can then be used to drive the pursuers to their destinations. Similarly, in [3] the assignment problem is modeled as a weighted graph matching problem. The authors consider a desired graph as a destination configuration and propose novel dynamical systems that always converge to a suboptimal assignment.

Pursuit-evasion games can be classified in continuous and purely discrete games. Continuous games explicitly model the physical motion and constraints of the players [4], [5], [6] and often assume worst case motion for the evaders [4], [5]. Studying optimality of such strategies involves numerically solving the computationally challenging Hamilton-Jacobi-Isaacs partial differential equations. On the other hand, discrete games are either played in purely discrete environments such as graphs [7], [8], or in continuous environments disregarding though any physical dynamics of the players [9], [10]. Such models may result in discrete strategies which are dynamically infeasible. Recently, however, hybrid control has been used to bridge the gap between the discrete and continuous games and model, more realistic, pursuers with limited sensing and communication capabilities. Approaches involve visibility-based [11] and probabilistic [12], [13] games. Closely related to the topics discussed in this paper are also multiple-target tracking by sensor networks [14], [15] which, however, focus more on the sensing and estimation problem of tracking rather than the actuation and control.

In this paper, we propose a novel distributed approach to the multiple-pursuer multiple-evader game, which is inspired by our previous work on dynamic assignment for stationary targets [16], [17]. Under the assumption that the pursuers have knowledge of the evaders' locations, we simultaneously address the discrete assignment of pursuers to evaders as well as the continuous control strategies for tracking and capturing the individual evaders. The resulting hybrid controller for each pursuer consists of both local coordination protocols [17], [18] guaranteeing that distinct evaders are captured by distinct pursuers, and single-pursuer single-evader time-varying potential fields ensuring convergence of the tracking error to any neighborhood of zero. Composition of the hybrid controllers for all pursuers results in a highly efficient overall system that is illustrated in nontrivial multiple-pursuer multiple-evader games. Similarly to [17], the assignment of evaders to pursuers is determined dynamically by means of information propagation in the underlying network regarding captured evaders and is shown to have the desired mutual exclusion property for all initial conditions. Furthermore, the overall system is shown to have at most polynomial complexity, despite the exponential growth of the number of assignments with respect to the number of pursuers.

The rest of this paper is organized as follows. In Section 2 we define the multiple-pursuer multiple-evader game. In Sections 3 and 4 we define the target tracking potential fields and the local coordination protocols, respectively, that consist the hybrid automata that model the pursuers. Properties of the overall system are discussed in Section 5, while in Section 6, we illustrate our scalable approach with nontrivial computer simulations.

2 Problem Formulation

Consider n identical pursuers in a p-dimensional space $\mathbb{R}^p$ and denote by $x_i(t) \in \mathbb{R}^p$ the coordinates of pursuer i at time t. We assume kinematic models for the pursuers and so,

$$\dot{x}_i(t) = u_i(t) \quad \forall\, i = 1, \ldots, n \tag{1}$$

where $u_i(t)$ is the control vector which in general can be unbounded. Consider, further, $m \geq n$ evaders and denote by $y_k(t) \in \mathbb{R}^p$ the coordinates of evader k at time t. Evader velocities can be time varying but are assumed to be bounded for all time, i.e., $\|\dot{y}_k(t)\|_2 < \infty$ for all time t and every evader k. For any evader k, let,

$$\mathcal{B}_r(y_k(t)) = \{x \in \mathbb{R}^p \mid \|x - y_k(t)\|_2 < r\} \tag{2}$$

denote an open ball of radius $r > 0$ centered at $y_k(t)$ and define the closure of $\mathcal{B}_r(y_k(t))$ by,

$$[\mathcal{B}_r](y_k(t)) = \{x \in \mathbb{R}^p \mid \|x - y_k(t)\|_2 \leq r\} \tag{3}$$

We say that pursuer i can *capture* evader k if for any given constant $\delta > 0$ there exists a time instant $T_i > 0$ such that $x_i(t) \in [\mathcal{B}_\delta](y_k(t))$ for all $t > t_0 + T_i$. Let $T = \max_i\{T_i\}$ denote the time instant that every pursuer has captured a distinct evader. Then, the time instant $t_0 + T$ corresponds to the *termination of the game* and the notion of capturing the evaders becomes equivalent to that of assigning evaders to pursuers.

Since the pursuers are considered "identical" any assignment, among the $\binom{m}{n}n!$ possible assignments between pursuers and evaders, is equally desirable. A popular approach is to decouple the assignment and tracking problems, i.e., determine first an assignment between pursuers and evaders, which can be either random or optimal, based on a "minimum distance to the goal" policy [2], [3], and then design controllers for every pursuer to capture its preassigned evader. Such approaches result in *centralized* and *off-line* control frameworks since, although navigation can be decentralized, an off-line centralized assignment decision needs to be made first. In this paper we propose a *dynamic* and fully *distributed* solution to the aforementioned problem. In particular, we assume that every pursuer has only knowledge of the evaders' locations, while the assignment decision is embedded in the controller and relies on pursuer communication. We therefore, address the following problem.

Problem 1 (Distributed Multiple-Pursuer Multiple-Evader Games). Given n identical pursuers, $m \geq n$ evaders and no a priori assignment information, derive distributed control laws for every pursuer i such that, for any $\delta > 0$ and any initial configuration $x_i(t_0)$, there exists a $T > 0$ such that $x_i(t) \in [\mathcal{B}_\delta](y_k(t))$ for all time $t > t_0 + T$, all pursuers i and distinct evaders k.

Implicit in Problem 1 is the *mutual exclusion* property of the final assignment, i.e., that no two pursuers may capture the same evader. In the following section we propose a control strategy for the *single-pursuer single evader game*, upon which we build a framework for *multiple-pursuer multiple-evader games*.

Compared to the optimal *line of sight* (LOS) policy for *single-pursuer single evader games* [5], the proposed strategy is smoother and less conservative since it does not explicitly depend on the worst case bounds on the evaders' velocities, while as the LOS policy, it also guarantees capturing of the evader.

3 Single-Pursuer Single-Evader Games

As in Section 2, let $x_i(t), y_k(t) \in \mathbb{R}^p$ denote the coordinates of pursuer i and evader k at time t, and define the functions $\gamma(x_i, y_k) = \|x_i - y_k\|_2$ and $r(t) = Re^{-a(t-t_0)}$, with $a \geq 0$ and $R > 0$. Let $\mathcal{B}_{r(t)}(y_k(t)) = \{x \in \mathbb{R}^p \mid \gamma(x, y_k(t)) < r(t)\}$ denote the open ball of radius $r(t)$ centered at $y_k(t)$. We then have the following result.

Theorem 1. *Let,* $\varphi_{R,a}(x_i, y_k, t) = \frac{1}{\beta_{R,a}(x_i, y_k, t)}$ *with* $\beta_{R,a}(x_i, y_k, t) = r^2(t) - \gamma^2(x_i, y_k)$ *and define the gradient flow,*

$$\dot{x}_i = u_{R,a}(x_i, y_k, t) := -K\nabla_{x_i}\varphi_{R,a}(x_i, y_k, t) \tag{4}$$

Assume, further, that $\sup_{t\geq t_0}\{\|\dot{y}_k(t)\|\} = M < \infty$ *and* $x_i(t_0) \in \mathcal{B}_{r(t_0)}(y_k(t_0))$. *Then,* $x_i(t) \in \mathcal{B}_{r(t)}(y_k(t))$ *for all* $t > t_0$. *Moreover, if* $a > 0$ *then, for any* $\delta > 0$ *there exists a* $T > 0$ *such that* $x_i(t) \in \mathcal{B}_\delta(y_k(t))$ *for all* $t > t_0 + T$.

Proof. Let $\partial\mathcal{B}_{r(t)}(y_k(t)) = \{x \in \mathbb{R}^p \mid \gamma(x, y_k(t)) = r(t)\}$ denote the boundary of the open ball $\mathcal{B}_{r(t)}(y_k(t))$. Then, $\varphi_{R,a}(\partial\mathcal{B}_{r(t)}(y_k(t)), y_k(t), t) = \infty$, while $\varphi_{R,a}(\mathcal{B}_{r(t)}(y_k(t)), y_k(t), t) < \infty$ for all $t \geq t_0$, which implies that $\varphi_{R,a}(\mathcal{B}_{r(t)}(y_k(t)), y_k(t), t) < \varphi_{R,a}(\partial\mathcal{B}_{r(t)}(y_k(t)), y_k(t), t)$ for all $t \geq t_0$. Moreover, for any $\varepsilon > 0$, define the *positive* and *negative* neighborhoods of $\partial\mathcal{B}_{r(t)}(y_k(t))$ by $\partial\mathcal{B}^+_{r(t)}(y_k(t)) = \{x \in \mathbb{R}^p \mid 0 < r(t) - \gamma(x, y_k(t)) < \varepsilon\}$ and $\partial\mathcal{B}^-_{r(t)}(y_k(t)) = \{x \in \mathbb{R}^p \mid -\varepsilon < r(t) - \gamma(x, y_k(t)) < 0\}$, respectively. We have that,

$$\begin{aligned}\dot{\varphi}_{R,a}(x_i, y_k, t) &= \frac{\partial\varphi_{R,a}(x_i, y_k, t)}{\partial t} + \frac{\partial\varphi_{R,a}(x_i, y_k, t)}{\partial y_k}\dot{y}_k + \frac{\partial\varphi_{R,a}(x_i, y_k, t)}{\partial x_i}\dot{x}_i \\ &\leq \frac{2aRe^{-a(t-t_0)}}{\beta^2_{R,a}(x_i, y_k, t)} + 2M\frac{\gamma(x_i, y_k)}{\beta^2_{R,a}(x_i, y_k, t)} - 4K\frac{\gamma^2(x_i, y_k)}{\beta^4_{R,a}(x_i, y_k, t)}\end{aligned}$$

Let $\varepsilon \to 0$. Then, $\dot{\varphi}_{R,a}(\partial\mathcal{B}^+_{r(t)}(y_k(t)), y_k(t), t) < 0$ for all $t \geq t_0$, since $2aRe^{-a(t-t_0)} + 2M\gamma(x_i, y_k) < 4K\frac{\gamma^2(x_i, y_k)}{\beta^2_{R,a}(x_i, y_k, t)}$ for all $x_i(t) \in \partial\mathcal{B}^+_{r(t)}(y_k(t))$ and all time $t \geq t_0$. Hence, the open ball $\mathcal{B}_{r(t)}(y_k(t))$ is an invariant set of the system (4), which proves the first claim. The second part of the claim, follows directly from the fact that $\lim_{t\to\infty} r(t) = 0$ and $x_i(t) \in \mathcal{B}_{r(t)}(y_k(t))$ for all $t \geq t_0$.

Note by Theorem 1 that capturing evader k is exponentially fast with rate $a > 0$, which combined with the fact that the equation (4) does not depend on the bounds on the evader's velocity, makes the proposed strategy very appealing. In the rest of this paper we elaborate on *multiple-pursuer multiple-evader games*. We build a distributed coordination scheme which combined with the tracking controllers (4), results in a hybrid control approach to Problem 1.

4 Multiple-Pursuer Multiple-Evader Games

Let $\mathcal{I}_0 = \{1, \ldots, m\}$ denote the index set corresponding to a fixed labeling of the evaders. We assume that every evader $k \in \mathcal{I}_0$ is uniquely associated to a coordinate vector $y_k(t) \in \mathbb{R}^p$ for all time t, through the injective map $evad : \mathcal{I}_0 \times \mathbb{R}_+ \to \mathbb{R}^p$, with respect to the variable k, which is such that,

$$evad(k, t) := y_k(t) \;\; \forall\, j \in \mathcal{I}_0 \tag{5}$$

Let $\mathcal{I}(t)$ and $\mathcal{I}^c(t)$ denote the index sets of available and taken evaders at time $t \geq t_0$, respectively.[1] Clearly, $\mathcal{I}(t_0) = \mathcal{I}_0$, $\mathcal{I}^c(t_0) = \emptyset$ and $\mathcal{I}(t) \cap \mathcal{I}^c(t) = \emptyset$, $\mathcal{I}(t) \cup \mathcal{I}^c(t) = \mathcal{I}_0$ for all $t \geq t_0$. In a distributed control framework, where the pursuers do not have access to the system's global variables, we require that every pursuer i is equipped with its own sets of available and taken evaders denoted by $\mathcal{I}_i^a(t)$ and $\mathcal{I}_i^t(t)$, respectively. The variables $\mathcal{I}_i^a(t)$ and $\mathcal{I}_i^t(t)$ are initialized such that every pursuer has knowledge of all available evaders in $\mathcal{I}_0$, i.e., $\mathcal{I}_i^a(t_0) = \mathcal{I}_0$ and $\mathcal{I}_i^t(t_0) = \emptyset$. Moreover, we require that $\mathcal{I}_i^a(t) = \{k\}$, for any evader $k \in \mathcal{I}_0$, if and only if pursuer i is assigned to that evader. On the other hand, as long as pursuer i is not yet assigned to any evader in $\mathcal{I}_0$, i.e., as long as $|\mathcal{I}_i^a(t)| > 1$, no evader can be considered both available and taken, i.e., $\mathcal{I}_i^a(t) \cap \mathcal{I}_i^t(t) = \emptyset$, while any evader that is not available, has to be taken, i.e., $\mathcal{I}_i^a(t) \cup \mathcal{I}_i^t(t) = \mathcal{I}_0$.

To achieve local coordination among the pursuers, we further define the set of neighbors of pursuer i at time t by $\mathcal{N}_i^\epsilon(t) = \{j \mid x_j(t) \in \mathcal{B}_\epsilon(x_i(t))\}$, where $\mathcal{B}_\epsilon(x_i(t))$ is defined as in equation (2), and call $\epsilon > 0$ the *coordination radius* of pursuer i. We assume that every pursuer can only exchange information with its neighbors in $\mathcal{N}_i^\epsilon(t)$ for all $t \geq t_0$. With the above notation, we now state the assumptions on which the construction of our model is based.

Assumptions 2 *For every pursuer i and all time $t \geq t_0$ we assume that,*

(a) it can capture an available evader $k \in \mathcal{I}(t)$, if and only if $k \in \mathcal{I}_i^a(t)$, $|\mathcal{I}_i^a(t)| > 1$ and $x_i(t) \in \mathcal{B}_\delta(y_k(t))$,[2]

(b) if $|\mathcal{I}_i^a(t)| > 1$ then, for any evader $k \in \mathcal{I}_0$ and any initial configuration $x_i(t_0)$, there exists a controller $u_i(x_i(t), y_k(t), t)$ such that, for any $\delta > 0$, guarantees that eventually $x_i(t) \in \mathcal{B}_\delta(y_k(t))$. Moreover, if $\mathcal{I}_i^a(t) = \{k\}$ for any $k \in \mathcal{I}_0$, then $u_i(x_i(t), y_k(t), t)$ is such that $x_i(t) \in \mathcal{B}_\delta(y_k(t))$ for all t,

(c) $\epsilon, \delta > 0$ are such that $\epsilon > 2\delta$.

Assumption 2(a) implies that, for any pursuer i, the condition $x_i(t) \in \mathcal{B}_\delta(y_k(t))$ is not sufficient for capturing evader k, since evader k must also be available. Assumption 2(b), on the other hand, implies that every pursuer can track any of its available evaders, unless it has already been assigned to one, whence it should always remain in a neighborhood of that evader. Combined with Assumption 2(a), Assumption 2(b) implies that *capturing* an evader is equivalent

[1] We call an evader "taken" if it is assigned to a pursuer and "available" otherwise.

[2] We denote by $|\mathcal{A}|$ the cardinality of the set $\mathcal{A}$.

to being *assigned* to an evader, as noted in previous sections. Note also that the controller proposed in Theorem 1 satisfies this assumption. Finally, Assumption 2(c) combined with Assumption 2(b) implies that any pursuer sufficiently close to an evader can sense whether this evader is taken or not.

To resolve cases where for any given pursuer i, Assumption 2(a) is simultaneously satisfied for multiple evaders, we assume that pursuer i can select a single evader to capture, among the ones satisfying Assumption 2(a), which we denote by $s_i \in \mathcal{I}_0$. Furthermore, to resolve *tie breaking* scenarios, where for any evader k, Assumption 2(a) is simultaneously satisfied for multiple pursuers, we require that every pursuer can identify the candidate pursuers, denoted by $\mathcal{C}_i(t)$, requesting to capture evader $k \in \mathcal{I}_0$ at time t, and can also break the tie if necessary. To achieve this specification, we introduce the function $tb : 2^{\mathbb{N}} \to \mathbb{N} \cup \{0\}$ such that,

$$tb(A) := \begin{cases} i \in A \text{ if } A \neq \emptyset \\ 0 \quad \text{if } A = \emptyset \end{cases} \tag{6}$$

and assume that every pursuer is equipped with such a function.[3] Then, the action $tb(\mathcal{C}_i)$, taken by any of the pursuers in $\mathcal{C}_i$, can break a tie for evader k, while the outcome can be transmitted to the other neighbors. Note that the set $\mathcal{C}_i$ is common for all pursuers $j \in \mathcal{C}_i$, by Assumptions 2(a) and 2(c). The rest of this section is devoted in developing a distributed hybrid coordination framework for the pursuers that is according to Assumptions 2. Then, in Section 5, the integrated system is studied and is shown to satisfy Problem 1.

4.1 Distributed Coordination

To achieve distributed coordination we propose a hybrid model for every pursuer that consists of a *navigation automaton* responsible for tracking and capturing any evader in $\mathcal{I}_i^a(t)$, and a *coordination automaton* designed to identify neighbors in $\mathcal{N}_i^\epsilon(t)$ and exchange information with them. The following notion of a *predicate* enables us to formally define the aforementioned automata.

Definition 1 (Predicate). *Let $X = \{x_1, \ldots, x_n\}$ be a finite set of variables. We define a predicate $\psi(X)$ over X to be a finite conjunction of strict or non-strict inequalities over X. We denote the set of all predicates over X by $Pred(X)$.*

In other words, a predicate is a logical formula. For example, the predicate $\psi(X) = \big(\|x - x_0\| < r\big)$ over the set of variables $X \in \mathbb{R}^N$ returns 1 if x belongs in the open ball $\|x - x_0\| < r$ and 0 otherwise. Hence, the navigation automaton for pursuer i can be defined as follows.[4]

Definition 2 (Navigation Hybrid Automaton). *We define the navigation hybrid automaton for pursuer i to be the tuple $N_i = (X_{N_i}, V_{N_i}, E_{N_i}, \Sigma_{N_i}, sync, inv, init, guard, reset, flow)$, where,*

[3] Note that $i \in A$ can be chosen randomly from any probability distribution.

[4] To simplify notation, we drop the dependence of the state variables on time.

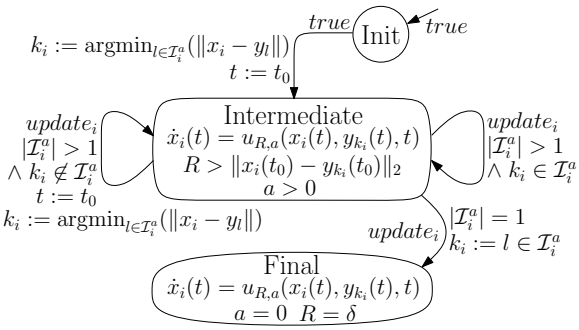


Fig. 1. Navigation Automaton for Pursuer i.

- $X_{N_i} = \{x_i, k_i, t_i\}$ *denotes the set of owned state variables with* $x_i \in \mathbb{R}^p$, $k_i \in \mathcal{I}_0$ *and* $t_i \in \mathbb{R}_+$.
- $V_{N_i} = \{Init, I, F\}$ *denotes the finite set of control modes.*[5]
- $E_{N_i} = \{(Init, I), (I, I), (I, F)\}$ *denotes the set of control switches.*
- $\Sigma_{N_i} = \{update_i\}$ *denotes the set of synchronization labels.*
- $sync : E_{N_i} \to \Sigma_{N_i}$ *with* $sync(e) = update_i$ *for all* $e \in E_{N_i} \backslash \{(Init, I)\}$, *denotes the synchronization map mapping each control switch to a synchronization label.*
- $inv : V_{N_i} \to Pred(X_{N_i})$ *with* $inv(v) = true$ *for all* $v \in V_{N_i}$, *denotes the invariant conditions of the hybrid automaton.*
- $init : V_{N_i} \to Pred(X_{N_i})$ *with* $init(v) = true$ *for* $v = Init$ *denotes the set of initial conditions.*
- $guard : E_{N_i} \to Pred(X_{N_i})$ *with,* $guard\big((Init, I)\big) = true$, $guard\big((I, I)\big) = \big(|\mathcal{I}_i^a| > 1\big)$ *and* $guard\big((I, F)\big) = \big(|\mathcal{I}_i^a| = 1\big)$, *denotes the set of guards of the hybrid automaton.*
- $reset : E_{N_i} \to X_{N_i}$ *with* $[x_i \;\; k_i \;\; t_i] := reset(e)$ *with,* $reset\big((Init, I)\big) = [x_i \; \text{argmin}_{l \in \mathcal{I}_i^a}(\|x_i - y_l\|_2) \; t_0]$, $reset\big((I, I)\big) = [x_i \; \text{argmin}_{l \in \mathcal{I}_i^a}(\|x_i - y_l\|_2) \; t_0]$ *if* $k_i \notin \mathcal{I}_i^a$, $reset\big((I, I)\big) = [x_i \; k_i \; t_i]$ *if* $k_i \in \mathcal{I}_i^a$ *and* $reset\big((I, F)\big) = [x_i \; \mathcal{I}_i^a \; t_0]$, *denotes the set of resets associated with the guards of the hybrid automaton.*
- $flow : V_{N_i} \to \dot{X}_{N_i}$ *with* $[\dot{x}_i \;\; \dot{k}_i \;\; \dot{t}_i] := flow(Init) = [0 \; 0 \; 1]$ *and* $[\dot{x}_i \;\; \dot{k}_i \;\; \dot{t}_i] := flow(v) = [u_{R,a}(x_i, y_{k_i}, t) \; 0 \; 1]$ *with* $u_{R,a}(x_i, y_{k_i}, t)$ *as in equation (4) and* $a > 0$, $R > \|x_i(t_0) - y_{k_i}(t_0)\|_2$ *if* $v = I$ *and* $a = 0$, $R = \delta$ *if* $v = F$, *denotes the flow conditions of the hybrid automaton that constrain the first time derivatives of the system variables in mode* $v \in V_{N_i}$.

Note in Definition 2 that every time the set of available evaders $\mathcal{I}_i^a$ is updated, automaton N_i selects a new evader $k_i \in \mathcal{I}_i^a$ for pursuer i to track. In our framework, this selection is based on a "minimum distance to the evader" policy, however, any other selection policy could also be used. Given any evader k_i, the flow conditions and Theorem 1 guarantee tracking of that evader such that the

[5] The shorthand notation stands for $I := Intermediate$, $F := Final$.

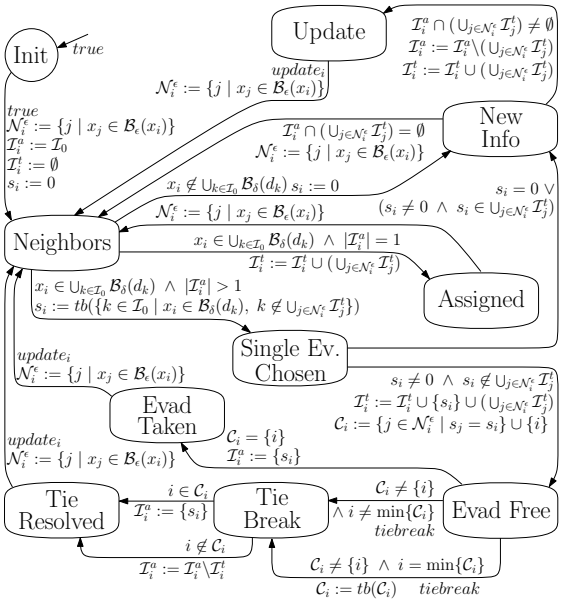


Fig. 2. Coordination Automaton for Pursuer i.

tracking error converges to any neighborhood of the origin. On the other hand, updating $\mathcal{I}_i^a$, which is due to the coordination automata, always decreases $|\mathcal{I}_i^a|$ and so eventually $|\mathcal{I}_i^a| = 1$, in which case $v_{N_i} = F$ and an assignment/capturing for pursuer i has been established. Figure 1 shows the graph representation of hybrid automaton N_i.

In the following definition, the coordination automaton for pursuer i is described. The coordination mechanism is such that every pursuer i always captures an available evader if it is sufficiently close to it (or breaks a tie if necessary) and always removes from $\mathcal{I}_i^a$ any taken evaders as a result of incoming information from its neighbors in $\mathcal{N}_i^\epsilon$.

Definition 3 (Coordination Hybrid Automaton). *We define the coordination hybrid automaton of pursuer i to be the tuple $C_i = (X_{C_i}, V_{C_i}, E_{C_i}, \Sigma_{C_i}, sync, inv, init, guard, reset, flow)$, where,*

- $X_{C_i} = \{\mathcal{I}_i^a, \mathcal{I}_i^t, \mathcal{N}_i^\epsilon, \mathcal{C}_i, s_i\}$ *denotes the set of owned state variables such that* $\mathcal{I}_i^a, \mathcal{I}_i^t \in 2^{\mathcal{I}_0}$, $\mathcal{N}_i^\epsilon, \mathcal{C}_i \in 2^{\{1,\dots,n\}}$ *and* $s_i \in \mathcal{I}_0$.
- $V_{C_i} = \{Init, N, I, U, A, S, F, T, B, R\}$ *denotes the finite set of control modes.*[6]
- $E_{C_i} = \{(Init, N), (N, I), (I, N), (I, U), (U, N), (N, A), (A, N), (N, S), (S, I), (S, F), (F, T), (F, B), (T, N), (B, R), (R, N)\}$, *denotes the set of control switches.*

[6] The shorthand notation stands for $N := Neighbors$, $I := New\ Info$, $U := Update$, $A := Assigned$, $S := Single\ Evad\ Chosen$, $F := Evad\ Free$, $T := Evad\ Taken$, $B := Tie\ Break$ and $R := Tie\ Resolved$.

- $\Sigma_{C_i} = \{update_i, tiebreak\}$ *denotes the set of synchronization labels.*
- $sync : E_{C_i} \rightarrow \Sigma_{C_i}$ *with,* $sync\big((F,B)\big) = tiebreak$ *and* $sync\big(e\big) = update_i$*, for* $e = (U,N),(T,N),(R,N)$*, denotes the synchronization map mapping each control switch to a synchronization label.*
- $inv : V_{C_i} \rightarrow Pred(X_{C_i})$ *with* $inv(v) = true$ *for all* $v \in V_{C_i}$*, denotes the invariant conditions of the hybrid automaton.*
- $init : V_{C_i} \rightarrow Pred(X_{C_i})$ *with* $init(v) = true$ *for* $v = Init$*, denotes the set of initial conditions.*
- $guard : E_{C_i} \rightarrow Pred(X_{C_i})$ *such that,* $guard\big((N,I)\big) = \big(x_i \notin \bigcup_{k\in\mathcal{I}_0} \mathcal{B}_\delta(y_k)\big)$, $guard\big((I,N)\big) = \big(\mathcal{I}_i^a \cap \big(\bigcup_{j\in\mathcal{N}_i^\epsilon} \mathcal{I}_j^t\big) = \emptyset\big)$, $guard\big((I,U)\big) = \big(\mathcal{I}_i^a \cap \big(\bigcup_{j\in\mathcal{N}_i^\epsilon} \mathcal{I}_j^t\big) \neq \emptyset\big)$, $guard\big((N,A)\big) = \big(x_i \in \bigcup_{k\in\mathcal{I}_0} \mathcal{B}_\delta(y_k) \ \wedge \ |\mathcal{I}_i^a| = 1\big)$, $guard\big((N,S)\big) = \big(x_i \in \bigcup_{k\in\mathcal{I}_0} \mathcal{B}_\delta(y_k) \wedge |\mathcal{I}_i^a| > 1\big)$, $guard\big((S,I)\big) = \big(s_i \neq 0 \wedge s_i \in \bigcup_{j\in\mathcal{N}_i^\epsilon} \mathcal{I}_j^t\big) \vee (s_i = 0)$, $guard\big((S,F)\big) = \big(s_i \neq 0 \wedge s_i \notin \bigcup_{j\in\mathcal{N}_i^\epsilon} \mathcal{I}_j^t\big)$, $guard\big((F,T)\big) = \big(\mathcal{C}_i = \{i\}\big)$, $guard\big((F,B)\big) = \big(\mathcal{C}_i \neq \{i\}\big)$ *and* $guard(e) = true$ *otherwise, denotes the set of guards of the hybrid automaton.*
- $reset : E_{C_i} \rightarrow X_{C_i}$ *with,* $[\mathcal{I}_i^a \ \mathcal{I}_i^t \ \mathcal{N}_i^\epsilon \ \mathcal{C}_i \ s_i] := reset(e)$ *such that,* $reset\big((Init, N)\big) = \big[\mathcal{I}_0 \ \emptyset \ \{j \mid x_j \in \mathcal{B}_\epsilon(x_i)\} \ \emptyset \ 0\big]$, $reset\big((N,I)\big) = \big[\mathcal{I}_i^a \ \mathcal{I}_i^t \ \mathcal{N}_i^\epsilon \ \mathcal{C}_i \ 0\big]$, $reset\big((I,N)\big) = \big[\mathcal{I}_i^a \ \mathcal{I}_i^t \ \{j \mid x_j \in \mathcal{B}_\epsilon(x_i)\} \ \mathcal{C}_i \ s_i\big]$, $reset\big((I,U)\big) = \big[\mathcal{I}_i^a \backslash \big(\bigcup_{j\in\mathcal{N}_i^\epsilon} \mathcal{I}_j^t\big) \ \mathcal{I}_i^t \cup \big(\bigcup_{j\in\mathcal{N}_i^\epsilon} \mathcal{I}_j^t\big) \ \mathcal{N}_i^\epsilon \ \mathcal{C}_i \ s_i\big]$, $reset\big((U,N)\big) = \big[\mathcal{I}_i^a \ \mathcal{I}_i^t \ \{j \mid x_j \in \mathcal{B}_\epsilon(x_i)\} \ \mathcal{C}_i \ s_i\big]$, $reset\big((N,A)\big) = \big[\mathcal{I}_i^a \ \mathcal{I}_i^t \cup \big(\bigcup_{j\in\mathcal{N}_i^\epsilon} \mathcal{I}_j^t\big) \ \mathcal{N}_i^\epsilon \ \mathcal{C}_i \ s_i\big]$, $reset\big((A,N)\big) = \big[\mathcal{I}_i^a \ \mathcal{I}_i^t \ \{j \mid x_j \in \mathcal{B}_\epsilon(x_i)\} \ \mathcal{C}_i \ s_i\big]$, $reset\big((N,S)_{C_i}\big) = \big[\mathcal{I}_i^a \ \mathcal{I}_i^t \ \mathcal{N}_i^\epsilon \ \mathcal{C}_i \ tb(\{k \in \mathcal{I}_0 \mid x_i \in \mathcal{B}_\delta(y_k), \ k \notin \bigcup_{j\in\mathcal{N}_i^\epsilon} \mathcal{I}_j^t\})\big]$, $reset\big((S,I)\big) = \big[\mathcal{I}_i^a \ \mathcal{I}_i^t \ \mathcal{N}_i^\epsilon \ \mathcal{C}_i \ s_i\big]$, $reset\big((S,F)\big) = \big[\mathcal{I}_i^a \ \mathcal{I}_i^t \cup \{s_i\} \cup \big(\bigcup_{j\in\mathcal{N}_i^\epsilon} \mathcal{I}_j^t\big) \ \mathcal{N}_i^\epsilon \ \{j \in \mathcal{N}_i^\epsilon \mid s_j = s_i\} \cup \{i\}\big]$, $reset\big((F,T)\big) = \big[\{s_i\} \ \mathcal{I}_i^t \ \mathcal{N}_i^\epsilon \ \mathcal{C}_i \ s_i\big]$, $reset\big((F,B)\big) = \big[\mathcal{I}_i^a \ \mathcal{I}_i^t \ \mathcal{N}_i^\epsilon \ tb(\mathcal{C}_i) \ s_i\big]$ *if* $i = \min\{\mathcal{C}_i\}$, $reset\big((F,B)\big) = \big[\mathcal{I}_i^a \ \mathcal{I}_i^t \ \mathcal{N}_i^\epsilon \ \mathcal{C}_i \ s_i\big]$ *if* $i \neq \min\{\mathcal{C}_i\}$, $reset\big((T, N)\big) = \big[\mathcal{I}_i^a \ \mathcal{I}_i^t \ \{j \mid x_j \in \mathcal{B}_\epsilon(x_i)\} \ \mathcal{C}_i \ s_i\big]$, $reset\big((B,R)\big) = \big[\{s_i\} \ \mathcal{I}_i^t \ \mathcal{N}_i^\epsilon \ \mathcal{C}_i \ s_i\big]$ *if* $i \in \mathcal{C}_i$, $reset\big((B,R)\big) = \big[\mathcal{I}_i^a \backslash \mathcal{I}_i^t \ \mathcal{I}_i^t \ \mathcal{N}_i^\epsilon \ \mathcal{C}_i \ s_i\big]$ *if* $i \notin \mathcal{C}_i$, $reset\big((R,N)\big) = \big[\mathcal{I}_i^a \ \mathcal{I}_i^t \ \{j \mid x_j \in \mathcal{B}_\epsilon(x_i)\} \ \mathcal{C}_i \ s_i\big]$*, denotes the set of resets associated with the guards of the hybrid automaton.*
- $flow : V_{C_i} \rightarrow \dot{X}_{C_i}$ *with* $[\dot{\mathcal{I}}_i^a \ \dot{\mathcal{I}}_i^t \ \dot{\mathcal{N}}_i^\epsilon \ \dot{\mathcal{C}}_i \ \dot{s}_i] := flow(v) = [0 \ 0 \ 0 \ 0 \ 0]$ *for all* $v \in V_{C_i}$*, denotes the flow conditions of the hybrid automaton that constrain the first time derivatives of the system variables in mode* $v \in V_{C_i}$*.*

Note first that for any pursuer i, automaton C_i is designed to select a single evader s_i, among the ones satisfying Assumption 2(a). This evader is then checked for availability and the sets $\mathcal{I}_i^a$ and $\mathcal{I}_i^t$ are appropriately updated. Whenever $\mathcal{I}_i^a$ is updated, a transition is triggered in automaton N_i due to synchronization labels "$update_i$". This synchronization models the communication between automata C_i and N_i. On the other hand, in a case of a *tie break* all the involved coordination automata are also synchronized to break the tie, due to the synchronization labels "*tiebreak*". Figure 2 shows the graph representation of hybrid automaton C_i.

5 Integration of the Overall System

Having defined the navigation and coordination automata of pursuer i, we now proceed with their composition in a product system.

Definition 4 (Product System). *We define the product of the hybrid automata $N_1, \ldots, N_n, C_1, \ldots, C_n$ by the tuple $S = (X_S, V_S, E_S, \Sigma_S, sync, inv, init, guard, reset, flow)$, where,*

- $X_S = X_{N_1} \cup \cdots \cup X_{N_n} \cup X_{C_1} \cup \cdots \cup X_{C_n}$ *denotes the set of state variables.*
- $V_S = V_{N_1} \times \cdots \times V_{N_n} \times V_{C_1} \times \cdots \times V_{C_n}$ *denotes the finite set of control modes.*
- $E_S = \{e_S\}$ *denotes the set of control switches such that, $e_S = e_{N_i} \| e_{C_i} \in E_S$ is defined as the set of control switches of S corresponding to control switches $e_{N_i} \in E_{N_i}$, and $e_{C_i} \in E_{C_i}$, with $sync(e_{N_i}) = sync(e_{C_i}) = update_i$, $e_S = \|_{i \in \mathcal{J}} e_{C_i} \in E_S$ is defined as the set of control switches of S corresponding to control switches $e_{C_i} \in E_{C_i}$, with $sync(e_{C_i}) = tiebreak$, for all $i \in \mathcal{J} \subseteq \{1, \ldots, n\}$, $\mathcal{J} \neq \emptyset$ and $e_S = (v_S, v'_S) \in E_S$ otherwise. Hence, the only variables that change with every transition $v_S \xrightarrow{e_S} v'_S$ are the ones involved in the control switch e_S through the corresponding automata.*
- $\Sigma_S = \Sigma_{N_1} \cup \cdots \cup \Sigma_{N_n} \cup \Sigma_{C_1} \cup \cdots \cup \Sigma_{C_n}$ *denotes the set of synchronization labels.*
- $sunc : E_S \rightarrow \Sigma_S$ *denotes the synchronization map mapping each control switch to a synchronization label.*
- $inv : V_S \rightarrow Pred(X_S)$ *with $inv(v_S) = inv(v_{N_1}) \wedge \cdots \wedge inv(v_{N_n}) \wedge inv(v_{C_1}) \wedge \cdots \wedge inv(v_{C_n})$ for all $v_S \in V_S$, denotes the invariant conditions of the product automaton.*
- $init : V_S \rightarrow Pred(X_S)$ *with $init(v_S) = init(v_{N_1}) \wedge \cdots \wedge init(v_{N_n}) \wedge init(v_{C_1}) \wedge \cdots \wedge init(v_{C_n})$ for all $v_S \in V_S$, denotes the set of initial conditions.*
- $guard : E_S \rightarrow Pred(X_S)$ *such that for any $\mathcal{J} \subseteq \{1, \ldots, n\}$, $\mathcal{J} \neq \emptyset$, $guard(e_S) = guard(e_{N_i}) \wedge quard(e_{C_i})$, if $e_S = e_{N_i} \| e_{C_i} \in E_S$, $guard(e_S) = \bigwedge_{i \in \mathcal{J}} guard(e_{C_i})$, for all $i \in \mathcal{J}$, if $e_S = \|_{i \in \mathcal{J}} e_{C_i} \in E_S$ and $guard(e_S) = guard\big((v_{C_i}, v'_{C_i})\big)$ for all $e_S = (v_S, v'_S) \in E_S$ otherwise, denotes the set of guards (or transitions) of the hybrid automaton.*
- $reset : E_S \rightarrow X_S$ *such that for any $\mathcal{J} \subseteq \{1, \ldots, n\}$, $\mathcal{J} \neq \emptyset$ with, $reset(e_S) = reset(e_{N_i}) \wedge reset(e_{C_i})$, if $e_S = e_{N_i} \| e_{C_i} \in E_S$, $reset(e_S) = \bigwedge_{i \in \mathcal{J}} reset(e_{C_i})$, for all $i \in \mathcal{J}$, if $e_S = \|_{j \in \mathcal{J}} e_{C_i} \in E_S$ and $reset(e_S) = reset\big((v_{C_i}, v'_{C_i})\big)$ for all $e_S = (v_S, v'_S) \in E_S$ otherwise, denotes the set of resets associated with the guards of the hybrid automaton.*
- $flow : V_S \rightarrow \dot{X}_S$ *with $flow(v_S) = flow(v_{N_1}) \cup \cdots \cup flow(v_{N_n}) \cup flow(v_{C_1}) \cup \cdots \cup flow(v_{C_n})$ for all $v_S \in V_S$, denotes the flow conditions of the hybrid automaton that constrain the first time derivatives of the system variables in mode v_S.*

Clearly, the product system S, being the composition of all elementary automata C_i and N_i, models the interconnection between them. Hence, studying

S, we can identify the properties of the whole multiple-pursuer multiple-evader system. The following result characterizes the transition guards in S.[7]

Proposition 1. *For every pursuer i and all t, let $s_i(t) \in \{k \in \mathcal{I}_0 \mid x_i(t) \in \mathcal{B}_\delta(y_k(t)),\ k \notin \bigcup_{j\in\mathcal{N}_i^\varepsilon(t)} \mathcal{I}_j^t(t)\}$. Then, $s_i(t) \neq 0$ if and only if $\{k \in \mathcal{I}_0 \mid x_i(t) \in \mathcal{B}_\delta(y_k(t)),\ k \notin \bigcup_{j\in\mathcal{N}_i^\varepsilon(t)} \mathcal{I}_j^t(t)\} \neq \emptyset$. Moreover, for any i, t such that $s_i(t) \neq 0$, if $|\mathcal{I}_i^a(t)| = 1$, then $s_i(t) = const.$, while if $|\mathcal{I}_i^a(t)| > 1$, then the product system S has the following properties:*

(a) $s_i(t) \notin \bigcup_{j\in\mathcal{N}_i^\varepsilon(t)} \mathcal{I}_j^t(t)$ if and only if evader $s_i(t)$ is available.
(b) $s_i(t) \in \bigcup_{j\in\mathcal{N}_i^\varepsilon(t)} \mathcal{I}_j^t(t)$ if and only if evader $s_i(t)$ is taken.

Proposition 1 shows that for any pursuer i the product system S can always select a unique evader s_i among the ones close to that pursuer, if any, and it can always identify whether s_i is available or taken. Moreover, for every pursuer i that has captured an evader, S guarantees that s_i is constant such that the assignment is final. We now proceed to showing that under the product system S, pursuer i always captures an available evader if it is sufficiently close to it, while it appropriately updates its sets of available and taken evaders, $\mathcal{I}_i^a$ and $\mathcal{I}_i^t$ respectively, otherwise.

Proposition 2. *For any pursuer i and all time t, the product system S has the following properties:*

(a) If $s_i(t) \neq 0$ is available at time t, then $\mathcal{I}_i^a(t) := \{s_i(t)\}$ and $\mathcal{I}_i^t(t) := \mathcal{I}_i^t(t) \cup \{s_i(t)\} \cup \big(\bigcup_{j\in\mathcal{N}_i^\varepsilon(t)} \mathcal{I}_j^t(t)\big)$.
(b) If $s_i(t) \neq 0$ is available at time t and there exist pursuers $j \neq i$ such that $s_i(t) = s_j(t)$, then S is able to break the tie.
(c) $\mathcal{I}_i^a(t) := \mathcal{I}_i^a(t) \backslash \big(\bigcup_{j\in\mathcal{N}_i^\varepsilon(t)} \mathcal{I}_j^t(t)\big)$ and $\mathcal{I}_i^t(t) := \mathcal{I}_i^t(t) \cup \big(\bigcup_{j\in\mathcal{N}_i^\varepsilon(t)} \mathcal{I}_j^t(t)\big)$ otherwise.

Note that Proposition 2 further implies that for all time $t \geq t_0$ such that $|\mathcal{I}_i^a(t)| > 1$, we have that $\mathcal{I}_i^a(t) \cap \mathcal{I}_i^t(t) = \emptyset$ and $\mathcal{I}_i^a(t) \cup \mathcal{I}_i^t(t) = \mathcal{I}_0$. Hence, the construction of our model is consistent with the system requirements in Section 4. The following proposition shows that every pursuer that has not yet captured an evader, has always knowledge of at least all available evaders in $\mathcal{I}(t)$. This property of system S is necessary to show that each pursuer will eventually capture a distinct evader in $\mathcal{I}_0$.

Proposition 3. *The product system S guarantees that $\mathcal{I}(t) \subseteq \mathcal{I}_i^a(t)$ for all time t and all pursuers i with $|\mathcal{I}_i^a(t)| > 1$.*

Our next result concerns the running time of the hybrid system S. In particular, we show that the product system S in the worst case can only take a finite number of transitions $v_S \xrightarrow{e_S} v'_S$ such that $sync(e_S) = update_i$, which is

[7] Proofs for this and all other results in this section can be found in Appendix A.

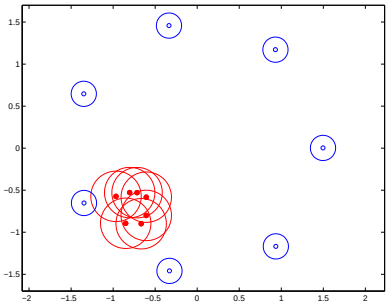


Fig. 3. Initial Configuration.

polynomial with respect to the number of pursuers n. Such transitions are triggered whenever $\mathcal{I}_i^a$ is updated for any i, and each time they result in $v_{N_i} = F$, an evader has been captured, while each time they result in $v_{N_i} = I$, information about taken evaders has been received. Clearly, many other transitions may occur in the meanwhile, but as long as these transitions are polynomial with n, it is guaranteed that the explored assignments are also polynomial with n. This result is important, given that the number of assignments, and hence the space of control modes V_S of S, grows exponentially with the number of pursuers.

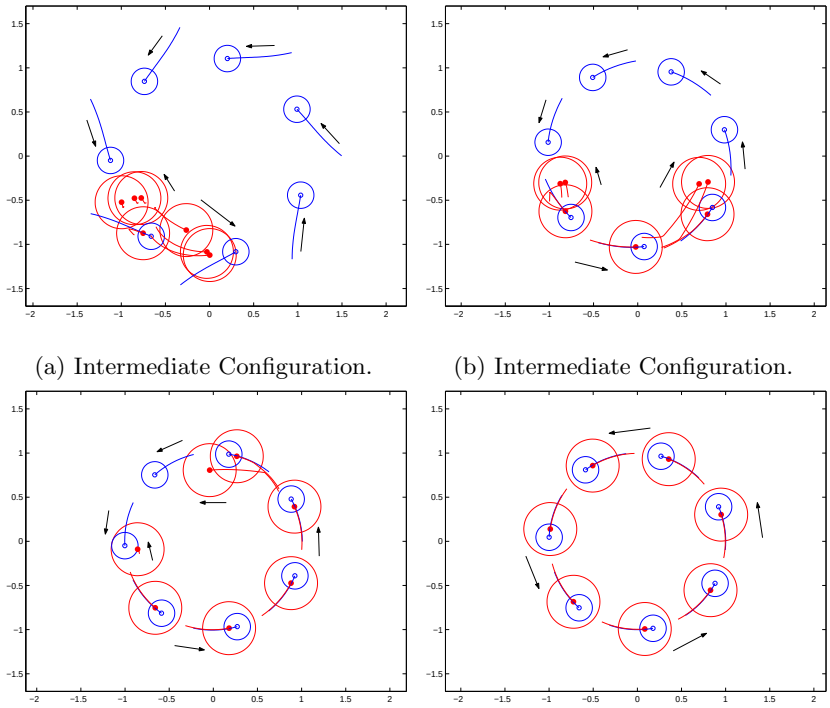
Proposition 4. *Let $v_S^\star = (v_{N_1}^\star, \ldots, v_{N_n}^\star, v_{C_1}, \ldots, v_{C_n})$ be such that $v_{N_i}^\star = F$ and $\mathcal{I}_i^a \cap \mathcal{I}_j^a = \emptyset$ for all $j \neq i$. Then, initialized at v_S^0, the product system S can reach $v_S^\star$ in at most $\frac{n(n+1)}{2}$ transitions $v_S \xrightarrow{e_S} v_S'$ with $sync(e_S) = update_i$.*

Having showed that the integrated system S satisfies the problem specifications and has also reasonable complexity, we now show that it also has the desired *liveness* and *safety* properties, i.e., that every pursuer will eventually capture a distinct evader. We hence, have the following theorem.

Theorem 3. *For all initial conditions $x_i(t_0)$, there exists a constant $T > 0$ such that for all time $t > t_0 + T$, the product system S is in mode $v_S^\star = (v_{N_1}^\star, \ldots, v_{N_n}^\star, v_{C_1}, \ldots, v_{C_n})$ with $v_{N_i}^\star = F$ and $\mathcal{I}_i^a(t) \cap \mathcal{I}_j^a(t) = \emptyset$ for all $j \neq i$. We call $v_S^\star$ the equilibrium mode of the system.*

6 Simulation Results

We consider a pursuit-evasion task where $n = 7$ pursuers, starting from randomly chosen initial configurations, are to capture $m = 7$ evaders, initialized on the perimeter of a circle (Figure 3). Figures 4 show the evolution of the system at 5 different time instants. The evaders are denoted with blue small circles and the δ-neighborhoods (with $\delta = .15$) around each evader, with big blue circles. The pursuers, on the other hand, are denoted with red color and the ϵ-neighborhoods (with $\epsilon = .3$) of each pursuer, with red circles. We observe that under the hybrid system S every pursuer eventually captures a distinct evader. Moreover, in every



(a) Intermediate Configuration. (b) Intermediate Configuration.

(c) Intermediate Configuration. (d) Final Configuration.

Fig. 4. Simulation for $n = 7$ pursuers and $m = 7$ evaders.

figure we plot the trace of the pursuers and evaders so that we can track their motion. Note how the pursuers change direction of motion when they receive information about taken evaders from their neighbors, without visiting the taken evaders themselves.

7 Conclusions

In this paper, we considered a distributed hybrid approach to the multiple-pursuer multiple-evader game, that simultaneously addresses the discrete assignment of pursuers to evaders as well as the continuous control strategies for capturing individual evaders. The assignment was determined dynamically through distributed coordination protocols, while tracking and capturing of the evaders was guaranteed by single-pursuer single-evader time-varying potential fields. The overall hybrid system was shown to always guarantee the mutual exclusion property of the final assignment and have at most polynomial complexity, despite the exponential growth of the number of assignments with respect to the number of pursuers. Our scalable approach was verified through non-trivial computer simulations.

References

1. S. Kloder and S. Hutchinson. *Path Planning for Permutation-Invariant Multirobot Formations*, IEEE Transactions on Robotics, vol. 22(4), pp. 650 - 665, Aug. 2006.
2. M. Ji, S. Azuma, and M. Egerstedt. *Role-Assignment in Multi-Agent Coordination*, International Journal of Assistive Robotics and Mechatronics, vol. 7(1), pp. 32-40, March 2006.
3. Michael M. Zavlanos and George J. Pappas. *A Dynamical Systems Approach to Weighted Graph Matching*, 45th IEEE Conference on Decision and Control, San Diego, December 2006, To appear.
4. T. Basar and G. Olsder. *Dynamic Noncooperative Game Theory*, Number 23. SIAM, 2nd edition, 1999.
5. R. Isaacs. *Differential Games*, Dover Publications, Mineola, New York, 1965.
6. A. Pedro Aguiar, Joao Hespanha. *Trajectory-Tracking and Path-Following of Underactuated Autonomous Vehicles with Parametric Modeling Uncertainty*, IEEE Transactions on Automatic Control, 2006. To appear.
7. M. Adler, H. Racke, N. Sivadasan, C. Sohler and B. Vocking. *Randomized Pursuit-Evasion in Graphs*, Combinatorics, Probability and Computing, vol. 12(3), pp. 225 - 244, 2003.
8. N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson and C. H. Papadimitriou. *The complexity of searching a graph*, Journal of the Association for Computing Machinery, vol. 35(1), pp. 18 44, 1988.
9. L. J. Guibas, J. C. Latombe, S. M. LaValle, D. Lin and R. Motwani. *A Visibility-Based Pursuit-Evasion Problem*, International Journal of Computational Geometry and Applications, vol. 9(4/5), pp. 471, 1999.
10. S. M. LaValle and J. E. Hinrichsen. *Visibility-Based Pursuit-Evasion: The Case of Curved Environments*, IEEE Transactions Robotics and Automation, vol. 17(2), pp. 196 - 202, April 2001.
11. V. Isler, C. Belta, K. Daniilidis and G. J. Pappas. *Hybrid Control for Visibility-Based Pursuit Evasion Games*, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, September 2004.
12. R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, S. Sastry. *Probabilistic Pursuit-Evasion Games: Theory, Implementation and Experimental Evaluation*, IEEE Transactions on Robotics and Automation, vol. 18(5), pp. 662 - 669, Oct. 2002.
13. M. Prandini, J. Hespanha, G. J. Pappas. *Greedy Control for Hybrid Pursuit Games*, Proceedings of the 2001 European Control Conference, Sep. 2001.
14. I. Hwang, H. Balakrishnan, K. Roy, and C. Tomlin. *Multiple-Target Tracking and Identity Management in Clutter, with Application to Aircraft Tracking*, Proceedings of the 2004 American Control Conference, Boston, MA, June 2005.
15. S. Oh, I. Hwang, K. Roy, and S. Sastry. *A Fully Automated Distributed Multiple-Target Tracking and Identity Management Algorithm*, Proceedings of the AIAA Guidance, Navigation, and Control Conference, San Francisco, CA, August 2005.
16. Michael M. Zavlanos and George J. Pappas. *Sensor-Based Dynamic Assignment in Distributed Motion Planning*, IEEE International Conference on Robotics and Automation, Rome, Italy, April 2007. Submitted.
17. Michael M. Zavlanos and George J. Pappas. *Dynamic Assignment in Distributed Motion Planning with Local Coordination*, American Control Conference, New York, NY, July 2007. Submitted.
18. L. Pallottino, V.G. Scordio, E. Frazzoli, and A. Bicchi. *Decentralized Cooperative Policy for Conflict Resolution in Multi-Vehicle Systems*, IEEE Transactions on Robotics, 2006. Submitted.

A Appendix

A.1 Proof of Proposition 1

Suppose that for $t = t_k$ automaton S is in mode v_S^k with $v_{C_i}^k = N$. The first part of this result follows directly from Assumption 2(b), as well as the resets $s_i(t_{k+1}) := reset\big((N,S)_{C_i}\big) = tb(\{l \in \mathcal{I}_0 \mid x_i(t_k) \in \mathcal{B}_\delta(y_l(t_k)),\ l \notin \bigcup_{j\in\mathcal{N}_i^\epsilon(t_k)} \mathcal{I}_j^t(t_k)\})$ and $s_i(t_{k+1}) := reset\big((N,I)_{C_i}\big) = 0$ and the definition of the function $tb(\cdot)$ in equation (6). On the other hand, if $s_i(t_k) \neq 0$ and $|\mathcal{I}_i^a(t_k)| = 1$, then $s_i(t_{k+1}) := reset\big((N,A)_{C_i}\big) = s_i(t_k)$ guarantees that $s_i(t) = const.$

To show property (a) when $|\mathcal{I}_i^a(t)| > 1$, note that if evader $s_i(t) \neq 0$ is available, then $s_i(t) \notin \mathcal{I}_j^t(t)$ for all j, which shows the "only if" part of the claim. On the other hand, for any pursuer i such that $s_i(t) \neq 0$ and $|\mathcal{I}_i^a(t)| > 1$, Assumptions 2(b) and 2(c) guarantee that if there exists a pursuer $j \neq i$ such that $\mathcal{I}_j^a(t) = \{s_i(t)\}$, then $j \in \mathcal{N}_i^\epsilon(t)$. Thus, if $s_i(t) \notin \bigcup_{j\in\mathcal{N}_i^\epsilon(t)} \mathcal{I}_j^t(t)$, then evader $s_i(t)$ has to be available, which shows the "if" part of the claim.

To show property (b), assume first that, for any pursuer i, evader $s_i(t) \neq 0$ is taken. Then, there exists a pursuer $j \neq i$ such that $\mathcal{I}_j^a(t) = \{s_i(t)\}$, and Assumptions 2(b) and 2(c) guarantee that $j \in \mathcal{N}_i^\epsilon(t)$. Thus, $s_i(t) \in \bigcup_{j\in\mathcal{N}_i^\epsilon(t)} \mathcal{I}_j^t(t)$, which shows the "only if" part of the claim. On the other hand, for any pursuer i, if $s_i(t) \in \bigcup_{j\in\mathcal{N}_i^\epsilon(t)} \mathcal{I}_j^t(t)$ then, evader $s_i(t)$ is clearly taken, which shows the "if" part of the claim.

A.2 Proof of Proposition 2

Suppose that for $t = t_k$ automaton S is in mode v_S^k, for any pursuer i, and consider the following cases.

Case I: Assume that $v_{C_i}^k = S$, $s_i(t_k) \neq 0$, $|\mathcal{I}_i^a(t_k)| > 1$ and that $s_i(t_k)$ is available, i.e., $s_i(t_k) \notin \big(\bigcup_{j\in\mathcal{N}_i^\epsilon(t_k)} \mathcal{I}_j^t(t_k)\big)$. Then, at $t = t_{k+1}$, S transitions to mode v_S^{k+1} with $v_{C_i}^{k+1} = F$ and $\mathcal{I}_i^t(t_{k+1}) := reset(e_S^k) = \mathcal{I}_i^t(t_k) \cup \{s_i(t_k)\} \cup \big(\bigcup_{j\in\mathcal{N}_i^\epsilon(t_k)} \mathcal{I}_j^t(t_k)\big)$. Moreover, the reset $\mathcal{C}_i(t_{k+1}) := reset(e_S^k) = \{j \in \mathcal{N}_i^\epsilon(t_k) \mid s_j(t_k) = s_i(t_k)\} \cup \{i\}$ identifies other pursuers that can simultaneously capture evader $s_i(t_k)$. If i is the only pursuer that can capture evader $s_i(t_k)$, i.e., if $\mathcal{C}_i(t_{k+1}) = \{i\}$, then at $t = t_{k+2}$, S transitions to mode v_S^{k+2} with $v_{C_i}^{k+2} = T$ and $\mathcal{I}_i^a(t_{k+2}) := reset(e_S^{k+1}) = \{s_i(t_k)\}$. Thus, property (a) is satisfied. Otherwise, if $\mathcal{C}_i(t_{k+1}) \neq \{i\}$, S transitions to mode v_S^{k+2} with $v_{C_j}^{k+2} = B$ for all $j \in \mathcal{C}_i(t_{k+1})$ instantaneously, due to the control switch $e_S^{k+1} = \|_{j\in\mathcal{C}_i(t_{k+1})} e_{C_j}^{k+1}$ which is such that $sync(e_{C_j}^{k+1}) = tiebreak$ for all $j \in \mathcal{C}_i(t_{k+1})$.[8] If i is the "leading" of the candidate pursuers in $\mathcal{C}_i(t_{k+1})$, i.e., if $i = \min\{\mathcal{C}_i(t_{k+1})\}$, then pursuer i "tosses a coin" to break the tie, i.e., $\mathcal{C}_i(t_{k+2}) := reset(e_S^{k+1}) = tb(\mathcal{C}_i(t_{k+1}))$, where the tie breaking function $tb(\cdot)$ is defined in equation (6). At time $t = t_{k+3}$,

[8] Note that the set $\mathcal{C}_i(t_{k+1})$ is common for all pursuers $j \in \mathcal{C}_i(t_{k+1})$ since condition $\epsilon > 2\delta$ guarantees that they are all neighbors of each other.

automaton S transitions to mode v_S^{k+3} with $v_{C_j}^{k+3} = R$ for all $j \in \mathcal{C}_i(t_{k+1})$, such that $\mathcal{I}_i^a(t_{k+3}) := reset(e_S^{k+2}) = \{s_i(t_{k+1})\}$ if $i \in \mathcal{C}_i(t_{k+2})$, where $s_i(t_k) = s_i(t_{k+1})$, and $\mathcal{I}_i^a(t_{k+3}) := reset(e_S^{k+2}) = \mathcal{I}_i^a(t_{k+2}) \backslash \mathcal{I}_i^t(t_{k+2})$ if $i \notin \mathcal{C}_i(t_{k+2})$, where $\mathcal{I}_i^t(t_{k+1}) = \mathcal{I}_i^t(t_{k+2})$. Hence, the tie is resolved and S also satisfies property (b).

Observe that, if at $t = t_{k+1}$ (or at $t = t_{k+2}$) there exists a pursuer $j \notin \mathcal{C}_i(t_{k+1})$ such that $s_j(t_{k+1}) = s_i(t_{k+1})$, then $\mathcal{C}_i(t_{k+1}) \subseteq \mathcal{N}_j^\epsilon(t_{k+1})$ and so $s_j(t_{k+1}) \in \bigcup_{i \in \mathcal{N}_j^\epsilon(t_{k+1})} \mathcal{I}_i^t(t_{k+1})$, i.e., evader $s_j(t_{k+1})$ is considered taken for pursuer j. In other words, tie breaking occurs only among the pursuers in $\mathcal{C}_i(t_{k+1})$.

Case II: Assume that $v_{C_i}^k = S$, and either $s_i(t_k) = 0$ or $s_i(t_k) \neq 0$, $|\mathcal{I}_i^a(t_k)| > 1$ and $s_i(t_k)$ is taken, i.e., $s_i(t_k) \in \big(\bigcup_{j \in \mathcal{N}_i^\epsilon(t_k)} \mathcal{I}_j^t(t_k)\big)$. Then, at $t = t_{k+1}$, S transitions to mode v_S^{k+1} with $v_{C_i}^{k+1} = I$. If pursuer i has already knowledge of all taken evaders provided by its neighbors, i.e., if $\mathcal{I}_i^a(t_{k+1}) \cap \big(\bigcup_{j \in \mathcal{N}_i^\epsilon(t_{k+1})} \mathcal{I}_j^t(t_{k+1})\big) = \emptyset$, then at $t = t_{k+2}$, S transitions to mode v_S^{k+2} with $v_{C_i}^{k+2} = N$ and no update is done. Otherwise, if $\mathcal{I}_i^a(t_{k+1}) \cap \big(\bigcup_{j \in \mathcal{N}_i^\epsilon(t_{k+1})} \mathcal{I}_j^t(t_{k+1})\big) \neq \emptyset$, then S transitions to mode v_S^{k+2} with $v_{C_i}^{k+2} = U$ and $\mathcal{I}_i^a(t_{k+2}) := reset(e_S^{k+1}) = \mathcal{I}_i^a(t_{k+1}) \backslash \big(\bigcup_{j \in \mathcal{N}_i^\epsilon(t_{k+1})} \mathcal{I}_j^t(t_{k+1})\big)$ and $\mathcal{I}_i^t(t_{k+2}) := reset(e_S^{k+1}) = \mathcal{I}_i^t(t_{k+1}) \cup \big(\bigcup_{j \in \mathcal{N}_i^\epsilon(t_{k+1})} \mathcal{I}_j^t(t_{k+1})\big)$. So S satisfies property (c).

Case III: Assume that $v_{C_i}^k = N$ and that pursuer i is already assigned to evader $s_i(t_k) \neq 0$, i.e., that $\mathcal{I}_i^a(t_k) = \{s_i(t_k)\}$. Then, by Assumption 2(b), $x_i(t_k) \in \mathcal{B}_\delta(d_{s_i(t_k)})$, and at $t = t_{k+1}$, S transitions to mode v_S^{k+1} with $v_{C_i}^{k+1} = A$. Clearly, $\mathcal{I}_i^t(t_{k+1}) := reset(e_S^k) = \mathcal{I}_i^t(t_k) \cup \big(\bigcup_{j \in \mathcal{N}_i^\epsilon(t_k)} \mathcal{I}_j^t(t_k)\big)$ and so S satisfies property (c).

Case IV: Assume that $v_{C_i}^k = N$ and that pursuer i is far from any evader, i.e., that $x_i(t_k) \notin \bigcup_{l \in \mathcal{I}_0} \mathcal{B}_\delta(y_l(t_k))$. Then, at $t = t_{k+1}$, S transitions to mode v_S^{k+1} with $v_{C_i}^{k+1} = I$ and $s_i(t_{k+1}) := reset\big(e_S^k\big) = 0$. If pursuer i has already knowledge of all taken evaders provided by its neighbors, i.e., if $\mathcal{I}_i^a(t_{k+1}) \cap \big(\bigcup_{j \in \mathcal{N}_i^\epsilon(t_{k+1})} \mathcal{I}_j^t(t_{k+1})\big) = \emptyset$, then at $t = t_{k+2}$, S transitions to mode v_S^{k+2} with $v_{C_i}^{k+2} = N$ and no update is done. Otherwise, if $\mathcal{I}_i^a(t_{k+1}) \cap \big(\bigcup_{j \in \mathcal{N}_i^\epsilon(t_{k+1})} \mathcal{I}_j^t(t_{k+1})\big) \neq \emptyset$, then S transitions to mode v_S^{k+2} with $v_{C_i}^{k+2} = U$ and $\mathcal{I}_i^a(t_{k+2}) := reset(e_S^{k+1}) = \mathcal{I}_i^a(t_{k+1}) \backslash \big(\bigcup_{j \in \mathcal{N}_i^\epsilon(t_{k+1})} \mathcal{I}_j^t(t_{k+1})\big)$ and $\mathcal{I}_i^t(t_{k+2}) := reset(e_S^{k+1}) = \mathcal{I}_i^t(t_{k+1}) \cup \big(\bigcup_{j \in \mathcal{N}_i^\epsilon(t_{k+1})} \mathcal{I}_j^t(t_{k+1})\big)$. So S satisfies property (c).

A.3 Proof of Proposition 3

Because of the assumption $|\mathcal{I}_i^a(t)| > 1$ on the system variables, we restrict our study to those pursuers i that are in a mode $v_{N_i} = I$. Let t_k denote the time instant that the the product system S takes its k-th transition. Clearly, between transitions the variables $\mathcal{I}_i^a(t)$ are constant and so it is sufficient to show that $\mathcal{I}(t) \subseteq \mathcal{I}_i^a(t)$ for all i with $v_{N_i} = I$, at the transition time instants t_k. To do so we use induction on k. Clearly, for $k = 0$ we have that $\mathcal{I}_i^a(t_0) = \mathcal{I}(t_0) = \mathcal{I}_0$ for all i, by initialization of the problem, and so $\mathcal{I}(t_0) \subseteq \mathcal{I}_i^a(t_0)$ for all i. Assume that $\mathcal{I}(t_k) \subseteq \mathcal{I}_i^a(t_k)$ for any $k > 0$ and all i with $v_{N_i}^k = I$, and consider the

transition $v_S^k \to v_S^{k+1}$ with corresponding control switch $e_S^k = (v_S^k, v_S^{k+1})$. Then, for $t = t_{k+1}$, we have the following cases:

Case I: For all pursuers i such that $v_{C_i}^k = I$ and $v_{C_i}^{k+1} = U$, the reset becomes $\mathcal{I}_i^a(t_{k+1}) := reset(e_S^k) = \mathcal{I}_i^a(t_k) \backslash \big(\bigcup_{j \in \mathcal{N}_i^\epsilon(t_k)} \mathcal{I}_j^t(t_k) \big)$. Since $v_{N_i}^k = I$ we have that, $\mathcal{I}_i^a(t_k) \cup \mathcal{I}_i^t(t_k) = \mathcal{I}_0$ and $\mathcal{I}_i^a(t_k) \cap \mathcal{I}_i^t(t_k) = \emptyset$. Hence, by Lemma 1(a) in Appendix B, the induction hypothesis $\mathcal{I}(t_k) \subseteq \mathcal{I}_i^a(t_k)$ implies that $\mathcal{I}_i^t(t_k) \subseteq \mathcal{I}_0 \backslash \mathcal{I}(t_k)$ and so, $\bigcup_{j \in \mathcal{N}_i^\epsilon(t_k)} \mathcal{I}_j^t(t_k) \subseteq \mathcal{I}_0 \backslash \mathcal{I}(t_k) = \mathcal{I}_0 \backslash \mathcal{I}(t_{k+1})$, since $\mathcal{I}(t_k) = \mathcal{I}(t_{k+1})$ if $v_{C_i}^k = I$ and $v_{C_i}^{k+1} = U$. By Lemma 1(b), the induction hypothesis and the fact that $\mathcal{I}(t_k) = \mathcal{I}(t_{k+1})$ we conclude that $\mathcal{I}_i^a(t_k) \backslash \big(\bigcup_{j \in \mathcal{N}_i^\epsilon(t_k)} \mathcal{I}_j^t(t_k) \big) \supseteq \mathcal{I}(t_{k+1})$ and so, $\mathcal{I}_i^a(t_{k+1}) \supseteq \mathcal{I}(t_{k+1})$.

Case II: For all pursuers i with $v_{C_i}^k = F$ and $v_{C_i}^{k+1} = T$, the reset $\mathcal{I}_i^a(t_{k+1}) := reset(e_S^k) = \{s_i(t_t)\}$ gives $|\mathcal{I}_i^a(t_{k+1})| = 1$.

Case III: For any pursuer i with $v_{C_i}^k = B$, $v_{C_i}^{k+1} = R$ and $i \in \mathcal{C}_i(t_k)$, the reset $\mathcal{I}_i^a(t_{k+1}) := reset(e_S^k) = \{s_i(t_k)\}$ gives $|\mathcal{I}_i^a(t_{k+1})| = 1$.

Case IV: For any pursuer i with $v_{C_i}^k = B$, $v_{C_i}^{k+1} = R$ and $i \notin \mathcal{C}_i(t_k)$, the reset becomes $\mathcal{I}_i^a(t_{k+1}) := reset(e_S^k) = \mathcal{I}_i^a(t_k) \backslash \big(\{s_i(t_k)\} \cup \big(\bigcup_{j \in \mathcal{N}_i^\epsilon(t_k)} \mathcal{I}_j^t(t_k) \big) \big)$. Applying Lemma 1(a) in Appendix B, the induction hypothesis implies that $\mathcal{I}_i^t(t_k) \subseteq \mathcal{I}_0 \backslash \mathcal{I}(t_k)$ and so,

$$
\begin{aligned}
\{s_i(t_k)\} \cup \Bigg(\bigcup_{j \in \mathcal{N}_i^\epsilon(t_k)} \mathcal{I}_j^t(t_k) \Bigg) &\subseteq \big(\mathcal{I}_0 \backslash \mathcal{I}(t_k) \big) \cup \{s_i(t_k)\} = \big(\mathcal{I}_0 \cap \mathcal{I}^c(t_k) \big) \cup \{s_i(t_k)\} \\
&= \big(\mathcal{I}_0 \cup \{s_i(t_k)\} \big) \cap \big(\mathcal{I}^c(t_k) \cup \{s_i(t_k)\} \big) \\
&= \mathcal{I}_0 \cap \big(\mathcal{I}(t_k) \cap \{s_i(t_k)\}^c \big)^c = \mathcal{I}_0 \cap \big(\mathcal{I}(t_k) \backslash \{s_i(t_k)\} \big)^c \\
&= \mathcal{I}_0 \backslash \big(\mathcal{I}(t_k) \backslash \{s_i(t_k)\} \big) = \mathcal{I}_0 \backslash \mathcal{I}(t_{k+1})
\end{aligned}
$$

since $\mathcal{I}(t_k) \backslash \{s_i(t_k)\} = \mathcal{I}(t_{k+1})$ if $v_{C_i}^k = B$ and $v_{C_i}^{k+1} = R$. By Lemma 1(b) in Appendix B, the induction hypothesis and the fact that $\mathcal{I}(t_k) \backslash \{s_i(t_k)\} = \mathcal{I}(t_{k+1})$ we conclude that $\mathcal{I}_i^a(t_k) \backslash \big(\{s_i(t_k)\} \cup \big(\bigcup_{j \in \mathcal{N}_i^\epsilon(t_k)} \mathcal{I}_j^t(t_k) \big) \big) \supseteq \mathcal{I}(t_{k+1})$ and so, $\mathcal{I}_i^a(t_{k+1}) \supseteq \mathcal{I}(t_{k+1})$.

A.4 Proof of Proposition 4

Let, $v_S^\star = (v_{N_1}^\star, \ldots, v_{N_n}^\star, v_{C_1}, \ldots, v_{C_n})$ be such that $v_{N_i}^\star = F$ for all i. Then, the condition that $\mathcal{I}_i^a \cap \mathcal{I}_j^a = \emptyset$ for all $j \neq i$ is guaranteed by Propositions 1, 2 and 3. In particular, Propositions 1 and 2 imply that only available evaders can be captured by any pursuer with $v_{N_i} = I$, and Proposition 3 that there always exists an available evader in $\mathcal{I}_i^a$ if $v_{N_i} = I$. Hence, a 1-1 correspondence is established between pursuers and evaders, which implies that mode $v_S^\star$ is a reachable mode of the system S. On the other hand, every transition $v_S \xrightarrow{e_S} v_S'$ such that $sync(e_S) = update_i$ decreases the value of $|\mathcal{I}_i^a|$ and thus, results in progress towards reaching mode $v_S^\star$. Since $\mathcal{I}_i^a$ are finite sets, the number of these transitions can only be finite. To derive an upper bound on these transitions,

we construct a worst case scenario and count the number of transitions that S takes in that scenario.[9]

Observe first, that to maximize the total number of transitions, we require that $\mathcal{I}_i^a$ is always updated such that $|\mathcal{I}_i^a(t_{k+1})| := |\mathcal{I}_i^a(t_k)| - 1$, since larger updates result in faster progress towards mode $v_S^\star$. Without loss of generality, we also assume that the order of transitions is such that the last transition of v_{N_i} indicates the first transition of $v_{N_{i+1}}$. Reordering the transitions, or relabeling the automata N_i, we can get any desired transition scheme. With these observations, we construct a worst case scenario as follows.

Initially, v_S^0 is such that $v_{N_i}^0 = I$ for all pursuers i. Let, $(I,F)_{N_1} \| (T,N)_{C_1}$ be the first control switch to be enabled. Then, transition $v_S^0 \xrightarrow{e_S} v_S^1$ is such that $v_{N_1}^1 = F$. Let $(I,I)_{N_2} \| (U,N)_{C_2}$ be the second control switch to be enabled. Then, transition $v_S^1 \xrightarrow{e_S} v_S^2$ is such that $v_{N_2}^2 = I$. The third control switch to be enabled is $(I,F)_{N_2} \| (F,N)_{C_2}$ and transition $v_S^2 \xrightarrow{e_S} v_S^3$ is such that $v_{N_2}^3 = F$. In the same way the fourth, fifth and sixth control switches to be enabled are $(I,I)_{N_3} \| (U,N)_{C_3}$, $(I,I)_{N_3} \| (U,N)_{C_3}$ and $(I,F)_{N_3} \| (T,N)_{C_3}$ respectively, and after the sixth transition $v_{N_3}^6 = F$. Hence, S takes 1 transition until $v_{N_1} = F$, 2 transitions until $v_{N_2} = F$, 3 transitions until $v_{N_3} = F$, up to n transitions until $v_{N_n} = F$, in which case S has reached the terminal mode $v_S^\star$. Adding up these transitions we get that S transitions $\frac{n(n+1)}{2}$ times in total, which completes the proof.

A.5 Proof of Theorem 3

Observe that we only need to show that there exists a constant $T > 0$ such that for all time $t > t_0 + T$, the product system S is in mode $v_S^\star = (v_{N_1}^\star, \ldots, v_{N_n}^\star, v_{C_1}, \ldots, v_{C_n})$ with $v_{N_i}^\star = F$ since then, Propositions 1, 2 and 3 guarantee that $v_S^\star$ is such that $\mathcal{I}_i^a(t) \cap \mathcal{I}_j^a(t) = \emptyset$ for all $j \neq i$. Thus, we only need to show that there does not exist a pursuer i such that $v_{N_i} = I$ for ever. In other words, we need to show that transitions $v_S \xrightarrow{e_S} v_S'$ such that $sync(e_S) = update_i$ will eventually occur until $v_S' = v_S^\star$. Then, since by Proposition 4, the system S can only take a finite number of such transitions, we can get $T > 0$ simply by adding the time intervals between these transitions.

But, the flow conditions for any mode v_S are given by the system of differential equations,

$$\dot{x}_i = u_{R,a}(x_i, y_{k_i}, t), \quad \forall\, i = 1, \ldots, n \ \text{ and } \ k_i \in \mathcal{I}_0 \tag{7}$$

such that $a > 0$ for all i with $v_{N_i} = I$ and $a = 0$ for all i with $v_{N_i} = F$. The system (7) is decoupled and hence, by Theorem 1, for all pursuers i with $v_{N_i} = I$, all initial conditions $x_i(t_0)$ and any $\delta > 0$, there exist $T_i = T_i(\delta, t_0) > 0$ such

[9] Without loss of generality, we do not include in the worst case scenario transitions due to *tie breaking*, since they correspond to one single successful assignment and multiple failed assignments occurring simultaneously, and could hence be treated separately without affecting the total number of transitions.

that $x_i(t) \in \mathcal{B}_\delta(y_{k_i}(t))$ for all $t > t_0 + T_i$. Let $T = \min_i\{T_i\}$. Then $t = t_0 + T$ denotes the time of the transition $v_S \to v'_S$, where v'_S is such that $v'_{N_j} = F$ for $j = \text{argmin}_i\{T_i\}$. Applying the same argument inductively until $v'_S = v^\star_S$ completes the proof.[10]

B Appendix

Lemma 1. *Let A, B, C and D be any sets. Then,*

(a) if $A \cup B = D$ and $A \cap B = \emptyset$, the inclusion $C \subseteq A$ implies that $B \subseteq D\backslash C$.
(b) if $C \subseteq D$, the inclusions $B \subseteq D\backslash C$ and $C \subseteq A$ imply that $C \subseteq A\backslash B$.

Proof. To prove (a) observe that if $x \in B$ then $x \in A \cup B$ and $x \notin A$. Hence, $x \in D$ and $x \notin C$ which implies that $x \in D\backslash C$.

To prove (b) observe that if $x \in C$ then $x \in A$ and $x \notin D\backslash C$. Hence, $x \in A$ and $x \notin B$ which implies that $x \in A\backslash B$.

[10] Observe that, $x_i(t_0) \in \mathcal{B}_{r(t_0)}(y_{k_i}(t_0))$ since $R > \|x_i(t_0) - y_{k_i}(t_0)\|_2$ for $v_{N_i} = I$ and so the conditions of Theorem 1 are satisfied.