On Localization and Communication Issues in Pursuit-Evasion Games*

Alberto Speranzon†, Karl Henrik Johansson

Royal Institute of Technology
Department of Signals, Sensors and Systems
Osquldasväg 10, 100-44 Stockholm, Sweden
{*albspe,kallej*}*@s3.kth.se*

Abstract

A probabilistic pursuit–evasion game from the literature is used as an example of a multi-robot system. The model is extended with particular focus on localization errors and communication limitations. These constraints are important in real multi-robot systems, but has often been neglected in previous studies. It is shown in the paper how multi-robot systems outperform single robots, also in the case of imperfect localization and bandwidth limitations in the communication channel. Simulation results give trade-offs between communication constraints, number of pursuers, and control performance.

1 Introduction

Multi-robot systems have many advantages compared to single-robot solutions, such as improved performance, sensing, distribution, and reliability. Examples of recent approaches for the study of multi-robot systems include behavior-based robotics [2, 15, 18], fuzzy control [17], dynamic game theory [8], and nonlinear control theory [4, 14]. Similarly to a single-robot system, one need in the design of multi-robot systems to consider challenges like partial knowledge of the environment, sensors noise, dynamic changes of the environment, and self-localization. For a multi-robot system, however, also the communication is of outmost importance. Every communication channel has a bandwidth limitation, due to that transmitted data packets must less than a maximum size, and (possibly) due to that the channel is shared so that communication only can take place during certain time intervals. From a heuristic view point, it seems obvious that communication must improve the performance of the overall system. Remarkably few research results exist, however, that support and quantify this belief. An exception is the simulation study and experiments by Balch and Arkin [1], which compared performances of a multi-robot system with and without perfect communication. Another exception is a result by Dudek et al. [6], which states that if the individual robots are modelled as finite automata and they can perfectly communicate their state to each other, then the multi-robot system is as powerful as a Turing machine. The current paper contributes to a paradigm for integrated control and communication design for multi-robot systems [13]. We let a classical pursuit–evasion game [10, 3] with several pursuers serve as a prototype system. The focus is on imperfect localization and communication issues. These are strongly coupled since position information is present in much of the relevant information to be communicated in a multi-robot system. The framework of pursuit–evasion games can be used to model several multi-robot situations, such as search-and-rescue operations, localization of lost objects, search-and-capture missions, forage-and-consume tasks, etc. In the paper we assume a Markov model for the motion of the evader, so the multi-robot system could hence model a search-and-rescue operations. The paper is organized as follows. In Section 2 we describe the model for the pursuit–evasion game. Section 3 extends the probabilistic framework developed by Hespanha et al. in [8] to the situation of imperfect localization of the pursuers. In Section 4 the classical approach of Multi Hypothesis Tracking is described in the case of pursuit–evasion games to keep low the number of hypothesis. We discuss pursuit–evasion games with limited communication bandwidth in Section 5. Simulation results show the interaction between communication constraints, number of pursuers, and control performance (rescue time). In Section 6 we suggest a simple information measure that can be used to

*This work is supported by the Swedish Research Council and Swedish and by the Swedish Foundation for Strategic Research through its Center for Autonomous Systems at the Royal Institute of Technology.

†Corresponding author.

decide when communicating. Finally, Section 7 contains some concluding remarks and directions for future research.

Notation Throughout the paper, we denote by $(\Omega, \mathcal{B}, P)$ the probability space where all probabilities are defined: Ω is the event set that contains all the possible events related to the pursuit–evasion games, $\mathcal{B}$ is a family of subsets of Ω forming a σ-algebra, and $P : \mathcal{B} \to [0, 1]$ is a probability measure on $\mathcal{B}$. The assumption on $\mathcal{B}$ is that it is rich enough so that all the probabilities considered below are well defined. We use boldface characters to denote random variables. For simplicity of notation we also denote probabilities like $P(\mathbf{x}_e(t) = x_e|\mathbf{Y}_t = Y_t)$ as $P(\mathbf{x}_e(t)|\mathbf{Y}_t)$, when values at which the function is calculated are arbitrary values.

2 Pursuit–Evasion Games

Consider a pursuit-evasion game with several pursuers and one randomly moving evader. The following notation is borrowed mainly from [8]. The main extension here is the introduction of pursuer localization errors and communication limitations. See [8, 9, 12] for other considerations. We suppose the space and the time are quantized. In particular, the space is divided in a finite number of cells $\mathcal{X} = \{1, 2, \dots, n_c\}$ and each event is associated a time instant t which belongs to the discrete set of times $\mathcal{T} = \{1, 2, \dots\}$. Each pursuer can sense the surroundings and collect measurements. Since the sensor measurements are corrupted by noise we use random variables to model the sensor data. We denote with $\mathbf{y}(t)$ the random vector of the measurements taken by the pursuers at time instant t, defined in a measurement space $\mathcal{Y}$. At each time instant $t \in \mathcal{T}$ the pursuers can execute a *control action* $\mathbf{u}(t)$ that, in general, corresponds to a movement to another cell of the state space. We denote with $\mathcal{U}$ the set of all possible control actions. The control action $\mathbf{u}(t)$ is a function of the measurements collected by the pursuers and is thus regarded as a random variable. For each time t we denote by $\mathbf{Y}_t \in \mathcal{Y}^*$ the set of all measurements[1] taken up to time t, which means that $\mathbf{Y}_t = \{\mathbf{y}(1), \dots, \mathbf{y}(t)\}$. We define the *control law* $g : \mathcal{Y}^* \to \mathcal{U} : \mathbf{Y}_{\bar{t}} \mapsto g(\mathbf{Y}_{\bar{t}})$ as a function that maps measurements taken up to some time $\bar{t}$ to a control action executed at the time instant $\bar{t} + 1$:

$$\mathbf{u}(\bar{t} + 1) = g(\mathbf{Y}_{\bar{t}}), \ \bar{t} \in \mathcal{T} \tag{1}$$

The control law g determines in general the next control action in order to minimize the time of finding the evader. In the pursuit–evasion game, $n_p \geq 1$

[1] We denote with $\mathcal{Y}^*$ the set of all finite sequences of the set $\mathcal{Y}$.

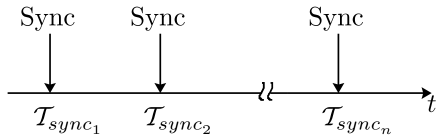


Figure 1: *Synchronization. At times $\mathcal{T}_{sync}$ the pursuers can use the communication channel to share information.*

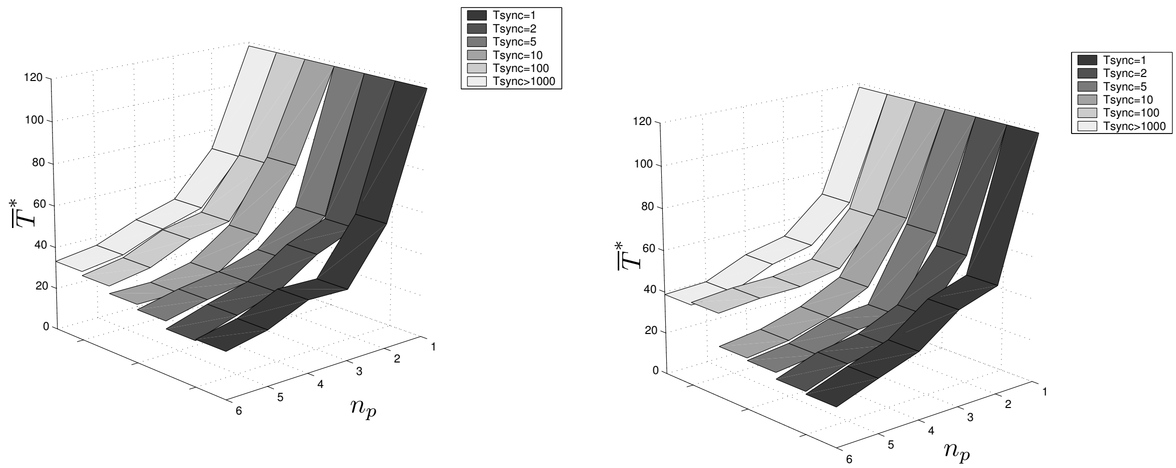
pursuers try to find a single moving evader. At each time $t \in \mathcal{T}$, denote the position of the evader as $\mathbf{x}_e(t) \in \mathcal{X}$ and the position of the pursuers as $\mathbf{x}(t) = \{\mathbf{x}_1(t), \mathbf{x}_2(t), \dots, \mathbf{x}_{n_p}(t)\} \in \mathcal{X}$. The positions $\mathbf{x}_e(t)$ and $\mathbf{x}_i(t)$ are random variables due to the uncertainty of the locations of the evader and the pursuers, respectively. Some cells are occupied by fixed obstacles, so that neither the pursuers nor the evader can move to these cells. We suppose that the position of the obstacles is unknown by the pursuers, but some initial map estimate is given. The obstacle map is defined by a function $\mathbf{m} : \mathcal{X} \to \{0, 1\}$, which takes the value one in those cells that contain an obstacle and zero otherwise. Since the positions of the obstacles is not known in advance, we consider $\mathbf{m}(x)$ to be a random variable. We assume that $\mathbf{m}(x)$, $x \in \mathcal{X}$, are independent random variables and that the game starts with an *a priori* obstacle map. To specify the structure of the control law g, we do not need at this point a precise probabilistic model of the pursuers and the evaders.It is enough to assume that, for each $x, x_e \in \mathcal{X}$ and for each $Y \in \mathcal{Y}^*$, it is possible to compute the conditional probability $p_{e,p}(x_e, x, Y) = P(\mathbf{x}_e(t+1) = x_e, \mathbf{x}(t+1) = x|\mathbf{Y}_t = Y_t)$ of the evader being in position x_e and the pursuers in position x given the measurements $\mathbf{Y}_t = Y$. We can then define the control law implicitly as

$$g(\mathbf{Y}_t) = \arg \max_{\{x_1, x_2, \dots, x_{n_p}\} \in \mathcal{U}} \sum_{k=1}^{{n_c}^{n_p}} p_{e,p}(x_k, x_k, Y_t) \tag{2}$$

which moves the pursuers to the cells that maximize the posterior probability of the evader and the pursuers being in the same cell at time $t + 1$. Note that this control law is a deterministic policy.

3 Probabilistic Map with Localization Errors

In this section we define the probabilistic map approach [21, 8, 7] for the pursuit–evasion game with one evader and several pursuers. When the information to build a map is not perfect, it is possible to assign the probability densities of the location of the objects. Such a probability density function is called a probabilistic map. Since the map is built using



(a) Each pursuer rely only on its own measurements until the fusion step at each $t = T_{sync}$.

(b) Each pursuer estimates the position of the teammates if $t \in (T_{sync_k}, T_{sync_{k+1}})$ and performs a fusion step at synchronization time.

Figure 2: *The average rescue time* $\overline{T}^*$ *as function of the number of pursuers* n_p *and the period of synchronization* $\mathbb{T}_{sync}$

sensor data, the probabilistic map is the probability density of object positions conditioned on the sensor measurements. For the problem of pursuit–evasion, we consider a Bayesian filtering [5] assuming that the environment is Markov (i.e., the past and the future data are (conditionally) independent given knowledge of the current state). The basic idea is thus to estimate the posterior probability density over the state space conditioned on the data. In robotics this posterior probability is also called *belief*.

3.1 Probabilistic Maps

In Section 2 we defined a model for the case of many pursuers and one evader. As in [8], we consider as sensor measurement for the single pursuer i the vector

$$\mathbf{y}_i(t) = (\mathbf{v}_i(t), \mathbf{o}_i(t), \mathbf{e}_i(t)) \tag{3}$$

where $\mathbf{v}_i(t) \in \mathcal{U}$ denotes the measured position of the pursuer (given, for example, from odometry), $\mathbf{o}_i(t) \subset \mathcal{X}$ the set of all the cells where an obstacle was detected, and $\mathbf{e}_i(t) \subset \mathcal{X}$ the set of cells where an evader was detected (using, for example, a sonar). A sensor measurement belongs to the set $\mathcal{Y} \triangleq \mathcal{U} \times 2^{\mathcal{X}} \times 2^{\mathcal{X}}$, where $2^{\mathcal{X}}$ denotes the power set of the set $\mathcal{X}$ (i.e., the set of all subsets of $\mathcal{X}$). The detection of obstacles is assumed to be perfect. The detection of the evader is however subjected to errors. In particular, we consider false positives (which correspond to detecting an evader even if the evader is not in the corresponding cell) and false negatives (which correspond to not detecting an evader even if the evader is in the corresponding cell). We denote the probability of false positives p and the probability of false negatives q, which hence are assumed to be constant. We also suppose that the detection of the objects (both evader and obstacles) can happen only if these objects are in a cell next to the cell occupied by the pursuer. Formally we define the set of adjacent cells to a cell x as $\mathcal{A}(x)$, which is thus the collection of cells that share a side or a corner with x. We assume that the motion of the evader and pursuers is of one cell at each step and the cells where they can move is in the neighborhood of the current position with exception of the evader that can also remain in the same cell. Thus in (2), $\mathcal{U}$ is defined to be in the set of all the cells neighborhood of cells where the probability density of the pursuer to be in those cells is not zero (or larger than a fixed threshold). We describe next how to compute the conditional posterior probability, or belief, $p_{e,p}(x_e, x, Y_t) = P(\mathbf{x}_e(t+1) = x_e, \mathbf{x}(t+1) = x | \mathbf{Y}_t = Y_t)$ of the evader being in cell x_e and the pursuer in the cell x at time $t+1$ given the measurements $\mathbf{Y}_t = Y_t$ taken up to time $t = |Y_t|$. We compute $p_{e,p}(x_e, x, Y_t)$ recursively in two steps:

1. *Measurement Step*: the probability $P(\mathbf{x}_e(t) = x_e, \mathbf{x}(t) = x | \mathbf{Y}_t = Y_t)$ of the evader being in cell x_e and the pursuers in x at time t given the measurements Y_t, is computed based on $p_{e,p}(x_e, x, Y_{t-1})$ and on the model of the drift $P(\mathbf{v}(t) = v | \mathbf{x}(t) = x)$ (localization error).

2. *Prediction Step*: the probability $p_{e,p}(x_e, x, Y_t) = P(\mathbf{x}_e(t+1) = x_e, \mathbf{x}(t+1) = x | \mathbf{Y}_t = Y_t)$ of the evader being in the cell x_e and the pursuer in the cell x at time $t+1$ given the measurements Y_t, computed from $P(\mathbf{x}_e(t) = x_e, \mathbf{x}(t) = x | \mathbf{Y}_t = Y_t)$.

The two steps of this recursive algorithm are further described below. The algorithm is initialized with some *a priori* probability $p_{e,p}(x_e, x, \emptyset)$, $x_e, x \in \mathcal{X}$, for the position of the evader and the pursuers[2].

3.2 Measurement Step

The probability map $p_{e,p}(x_e, x, Y_t)$ can be written, using Bayes' rule, as

$$p_{e,p}(x_e, x, Y_t) = \zeta_2 \, p_{e,p}(x_e, x, Y_{t-1}) \cdot P(\mathbf{y}(t) | \mathbf{Y}_{t-1}, \mathbf{x}_e(t), \mathbf{x}(t)) \quad (4)$$

where $\zeta_2 = 1/P(\mathbf{y}(t) = y | \mathbf{Y}_{t-1} = Y_{t-1})$ is a positive normalizing constant, independent of x_e and x. The last term of (4) is equal to

$$\begin{aligned} P(\mathbf{y}(t) | \mathbf{x}_e(t), \mathbf{x}(t)) &= P(\mathbf{v}(t), \mathbf{o}(t), \mathbf{e}(t) | \mathbf{x}_e(t), \mathbf{x}(t)) = \\ & P(\mathbf{v}(t) | \mathbf{x}_e(t), \mathbf{x}(t)) \cdot P(\mathbf{e}(t) | \mathbf{x}_e(t), \mathbf{x}(t), \mathbf{v}(t)) \cdot \\ & P(\mathbf{o}(t) | \mathbf{x}_e(t), \mathbf{x}(t), \mathbf{v}(t), \mathbf{e}(t)) \end{aligned} \quad (5)$$

The first term of (5) can be written as

$$P(\mathbf{v}(t) = v | \mathbf{x}_e(t) = x_e, \mathbf{x}(t) = x) = P(\mathbf{v}(t) = v | \mathbf{x}(t) = x)$$

since the odometry measurements are independent of the position of the evader. A model of the drift error $\mathbf{w}$ of the pursuers is now needed. We capture them by introducing the probability density function $p_w(x)$, $x \in \mathcal{X}$, for the position of the pursuers, which we assume to be known[3]. If we assume that the measurements can be written as

$$\mathbf{v}(t) = \mathbf{x}(t) + \mathbf{w}(t) \quad (6)$$

where $\mathbf{w}$ is some noise, then we have

$$P(\mathbf{v}(t) = v | \mathbf{x}(t) = x) = p_w(\mathbf{v}(t) - \mathbf{x}(t)) \quad (7)$$

For the second term of (5) we have

$$P(\mathbf{e}(t) | \mathbf{x}_e(t), \mathbf{x}(t), \mathbf{v}(t)) = \begin{cases} 0 & \text{if } x_e \notin \mathcal{A}(x) \\ p^{k_1} (1-p)^{k_2} q^{k_3} (1-q)^{k_4} & \text{otherwise} \end{cases} \quad (8)$$

where k_1 are the number of pursuers that give false positive, k_2 the number of pursuers that give true negative, k_3 the number of pursuers that give false negatives and k_4 the number of pursuers that give true positives. Due to the low obstacle density, the third term of (5) is approximately independent of the position of the evader $\mathbf{x}_e(t)$ and the position of the pursuer $\mathbf{x}(t)$. Thus it can be rewritten as

$$P(\mathbf{o}(t) | \mathbf{x}_e(t), \mathbf{x}(t), \mathbf{v}(t), \mathbf{e}(t)) = P(\mathbf{o}(t) | \mathbf{v}(t), \mathbf{e}(t)) \quad (9)$$

which is not a function of the state variables and thus can be regarded as a constant ζ_3, say.

3.3 Prediction Step

For the prediction step we can write the posterior conditional probability as follows

$$\begin{aligned} p_{e,p}(x_e, x, Y_t) &= P(\mathbf{x}_e(t+1) = x_e, \mathbf{x}(t+1) = x | \mathbf{Y}_t = Y_t) \\ &= \sum_{\overline{x} \in \mathcal{A}(x)} P(\mathbf{x}_e(t+1) = x_e, \mathbf{x}(t+1) = x, \\ & \quad \mathbf{x}(t) = \overline{x}, \mathbf{m}(x) = 0 | \mathbf{Y}_t = Y_t) \\ &= \sum_{\overline{x} \in \mathcal{A}(x)} \sum_{\overline{x}_e \in \{x_e\} \cup \mathcal{A}(x_e)} P(\mathbf{x}_e(t+1) = x_e, \mathbf{x}_e(t) = \overline{x}_e, \\ & \quad m(x_e) = 0, \mathbf{x}(t+1) = x, \mathbf{x}(t) = \overline{x}, \mathbf{m}(x) = 0 | \mathbf{Y}_t = Y_t) \end{aligned} \quad (10)$$

where we used the model of the motion of the evader and the pursuers. We can expand (10) using Bayes' rule:

$$\begin{aligned} & \sum_{\overline{x} \in \mathcal{A}(x)} \sum_{\overline{x}_e \in \{x_e\} \cup \mathcal{A}(x_e)} P(\mathbf{x}_e(t+1) = x_e, \mathbf{x}(t+1) = x | \mathbf{x}(t) = \overline{x}, \\ & \quad \mathbf{x}_e(t) = \overline{x}_e, \mathbf{m}(x_e) = 0, \mathbf{m}(x) = 0, \mathbf{Y}_t = Y_t) \cdot \\ & \quad P(\mathbf{m}(x_e) = 0, \mathbf{m}(x) = 0 | \mathbf{x}_e(t) = x_e, \mathbf{x}(t) = x, \mathbf{Y}_t = Y_t) \cdot \\ & \quad P(\mathbf{x}_e(t) = \overline{x}_e, \mathbf{x}(t) = \overline{x} | \mathbf{Y}_t) \end{aligned} \quad (11)$$

where the last factor of (11) is known from the measurement step. Since the obstacle density is low, $x_e(t)$ and $x(t)$ are approximately independent of $m(x_e)$ and $m(x)$. This means that we have $P(\mathbf{m}(x_e) = 0, \mathbf{m}(x) = 0 | \mathbf{x}_e(t) = x_e, \mathbf{x}(t) = x, \mathbf{Y}_t = Y_t) \approx P(\mathbf{m}(x_e) = 0 | \mathbf{Y}_t = Y_t) \, P(\mathbf{m}(x) = 0 | \mathbf{Y}_t = Y_t)$. Since obstacles do not move, we can rewrite these factors as

$$P(\mathbf{m}(x_e) = 0 | Y_t = Y_t) = \begin{cases} 1 & \text{if } x_e \notin o(t) \text{ and } x_e \in \mathcal{A}(x) \cup \{x\} \\ 0 & \text{if } x_e \in o(t) \text{ and } x_e \in \mathcal{A}(x) \cup \{x\} \\ P(\mathbf{m}(x_e) = 0 | \mathbf{Y}_{t-1} = Y_{t-1}) & \text{otherwise} \end{cases}$$

[2] Here $\emptyset \in \mathcal{Y}^*$ denotes the empty sequence of measurements.

[3] The assumptions here are that the pursuers use the same control law and they can localize themselves after some time, so that the drift is bounded. This is plausible since it is in many cases possible to find landmarks (for example, doors) that resets the localization error to zero [11].

The first factor of (11) can be expanded as

$$P(\mathbf{x}_e(t+1) = x_e | \mathbf{x}(t+1) = x, \mathbf{x}(t) = \overline{x}, \\ \mathbf{x}_e(t) = \overline{x}_e, \mathbf{Y}_t = Y_t) \cdot P(\mathbf{x}(t+1) = x | \mathbf{x}(t) = \overline{x}, \\ \mathbf{x}_e(t) = \overline{x}_e, \mathbf{Y}_t = Y_t) \quad (12)$$

In the first factor of (12), we note that the position of the evader at time step $t+1$ is independent of the position of the pursuers (Markovian motion of the evader). The probability of the evader to move from its position to an adjacent cell or to stay in the same cell is uniform and equal to 1/9. The second factor of (12) is equal to 1 since the control law is deterministic.

4 Multiple Hypothesis

The probability map $p_{e,p}(x_e, x, Y_t)$ for a single pursuer, i.e. when the random variable x has only one component which models its position in the environment, for all possible pairs (x_e, x) gives the probability of the evader being in position x_e and the pursuer begin in position x. This means that the dimension of the probability map $p(x_e, x, Y_t)$ in the simple case of one evader and one pursuer is $\mathcal{X}^2$. Roughly speaking this is equivalent to consider n_c copies of the original map $\mathcal{X}$ where each copy represents the probability of the evader being in all the possible positions assuming that the pursuer is fixed in a particular position with a given probability. We can write that each probability map, M_k, is defined as:

$$M_k \triangleq p(x_e(t) = x_e, x(t) = i | Y_t = Y_t) = \\ P(\mathbf{x}_e(t) = x_e, \mathbf{x}(t) = k | \mathbf{Y}_t = Y_t) \quad (13)$$

for $\forall x_e \in \mathcal{X}$ and a given position $k \in \mathcal{X}$ of the pursuer. A particular map M_k is null, i.e. all values are zeros if the probability of the pursuer of being in k is zero as expressed in (5). Depending on the model of the drift of the robot we generally have a high number of not null maps M_k and this means that we have to consider all of them to compute the control law (2). Each probability map M_k represents a *hypothesis* [20, 16]. When we consider a multi-robot system the hypothesis grow exponentially with the number of pursuers and the computation of a control action is very heavy to perform and the performance decreases very fast. To bound the total number of hypothesis we rely on:

- natural landmarks (like walls and obstacles)
- ad-hoc landmarks (landmarks put in the environment with the purpose of letting the pursuer to localize itself)

The natural landmarks allow to prune some hypothesis that do not match the measurements. As example we can suppose to have two hypothesis expressed by the two probability maps M_{k_1} and M_{k_2}, one modelling the fact that the pursuer is in position k_1 with some probability and in position k_2, which is close to a wall, with some other probability. If during the measurement step is not detected a wall the hypothesis that model the possibility of the pursuer to be close to a wall can be considered false and delete. In this simple way it is possible to limit the grew of the hypothesis. In the case of large environments, which means that the pursuer will be very often far from the walls some ad-hoc landmarks spread in the environment can be assumed to be used. Anyway considering a more structured environment the number of these ad-hoc landmarks can be kept quite low.

5 Communication Limitations

In the previous sections we proposed an algorithm that maximizes the posterior probability map of the evader and the pursuer to be in the same cell at step $t+1$. Implicitly we assumed that the pursuers can communicate over a channel that is always available. That system can be considered *centralized*, since each pursuer has access to the measurements of all the other pursuers at each time instance. In this section we consider a different set-up in which the observations are available to all the pursuers only at particular time instances. At these instances, the probability maps of the pursuers can be synchronized by fusing their individual information. In order to simplify the considerations, we assume that each pursuers know with probability one its position in the map $\mathcal{X}$ (i.e., no drift $\mathbf{w} \equiv 0$). The pursuers can communicate only at the time instances:

$$\mathcal{T}_{sync} = \{T_{sync_1}, T_{sync_2}, \dots\} \subset \mathcal{T} \quad (14)$$

as depicted in Figure 1. At times T_{sync} the pursuers can communicate to each other and integrates the measurements of the other pursuers (*sensor fusion*)[4]. Denote by $\mathbf{Y}_t^i$ the sequence of measurements taken by pursuer i up to time t. Then we can recursively define,

$$\mathbf{Y}^i_{T_{sync_k}} = \mathbf{Y}^i_{T_{sync_k}} \cup \bigcup_{j=1, j\neq i}^{n_p} \mathbf{Y}^j_{T_{sync_k}} \quad (15)$$

Between two synchronization instances, each pursuer moves by maximizing the probability of finding the evader at the next time instance given its own measurements. Figure 2(a) shows the results of 100 simulations for different numbers of pursuers and different

[4]In the following we call with T_{sync} a general synchronization time.

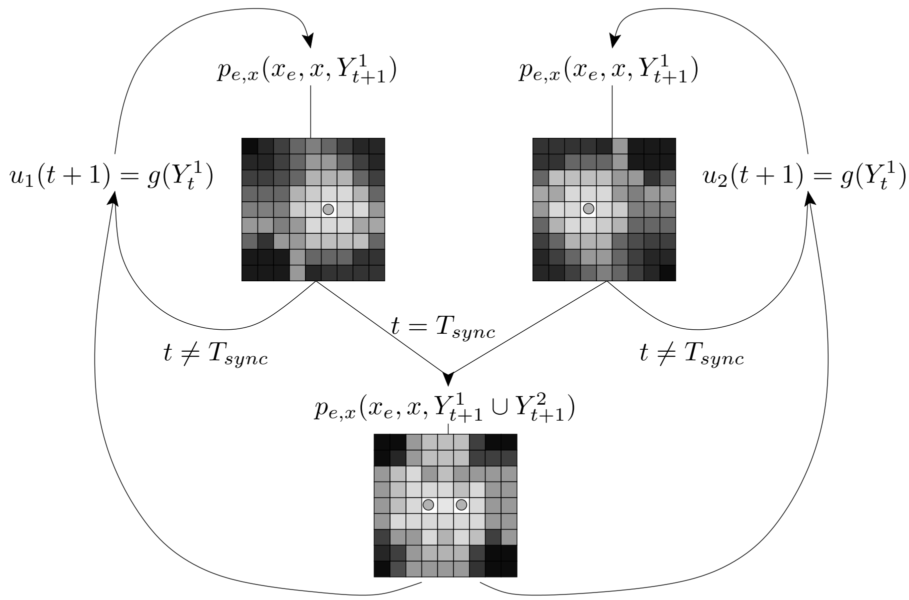


Figure 3: *An example of a two pursuers–one evader game with periodic communication. The darker the color of the cell, the higher is the probability of finding an evader. When there is no synchronization, the pursuers rely on their measurements. At $t = T_{sync}$ the pursuers communicate the measurements and a new probability map is computed (sensor fusion).*

period of $\mathbb{T}_{sync}$. We can see that for a fixed $\mathbb{T}_{sync}$, the average rescue time $\overline{T}^*$ is decreasing as the number of pursuers increases, meaning that a multi-robot system performs better than a single robot with respect to the rescue time. If the period of synchronization increases, the rescue time increases. Figure 3 illustrates the probability maps of two pursuers between and at the synchronization times. The main difficulty in solving the pursuit–evasion problem when having a communication channel that is available only a certain instances, as given by $\mathbb{T}_{sync}$, is that between these instances each pursuer relies only on its own measurements. A simple and intuitive idea to improve the performance of the overall system (i.e., to decrease the average rescue time $\overline{T}^*$) is to let each pursuer estimate the measurements of all the other pursuers between synchronization times. Note that the only measurement that is more reasonable to estimate is the position of the teammates. In order to estimate the positions of the other pursuer, each pursuer must have a model of how the teammates move (i.e., the control laws and possibly a model of the drift). Figure 2(b) shows the results of simulations for different $\mathbb{T}_{sync}$, when each pursuer is estimating the position of the teammates. If either the number of pursuers increases or the synchronization interval decreases, the average rescue time $\overline{T}^*$ decreases. Note that if the synchronization interval is not too large, then the system where pursuers estimate the position of the teammates outperform the other one. A more detailed plot is shown in Figure 5. We can see that the system where pursuers use estimation perform better in terms of average rescue time, but only if the period of synchronization is not too large. The cause of this is due to false positives and false negatives. The estimated position of the pursuer coincides with the real one only if the measurements are perfect. Since we have false positives and false negatives, the error between estimated position and real position is in general not zero. Moreover, if the period of synchronization is very large then the error between estimated and real position becomes large, which result in a degradation of the performance of the overall system.

6 Entropy Based Synchronization

In the previous section we discussed the communication issue between pursuers when the synchronization happens periodically. Intuitively not all the probability maps are very "important" for the teammates, for example a map that is flat (i.e. the probability map is close to the uniform distribution) can be considered quite useless since does not give any idea of where the pursuer can be. The idea described here is very simple and intuitive, but we need to formalize the concept of "important" probability maps. We can relate this concept to the information content of the probability map and we can use the entropy function [19] of the probability map as measure. The

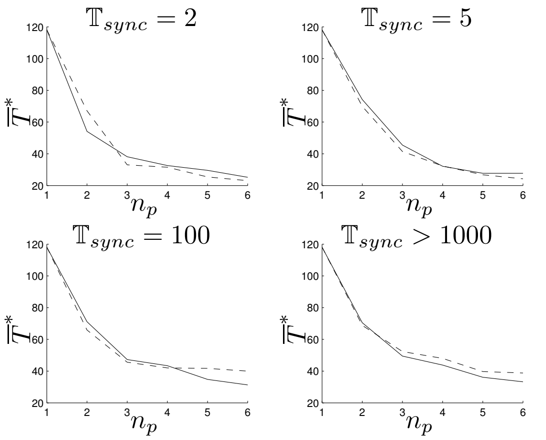


Figure 4: *The average rescue time* $\overline{T}^*$ *as function of the number of pursuers* n_p *for four different synchronization periods. The solid lines refer to the case of no estimation of the position of teammates, while the dashed lines refer to the case of estimation.*

entropy is defined as:

$$H(\Psi) \triangleq -\Psi\, log_2\,(\Psi)) \qquad (16)$$

where $\Psi = p_{e,p}(x_e, x, Y_t)$ is a probability map. A threshold on the entropy can give a simple method to decide when to communicate a particular probability map. For example, if the entropy is very small means that the probability map contains useful information and should be sent. In Figure 5 there are some results obtained using the entropy-based synchronization in the case of two pursuers and one evader game. In the upper plot of Figure 5 curve 1 shows an average of the synchronization period for each simulation. Since the synchronization is controlled by the entropy of the probability map and the synchronization instant becomes dynamic, the plot shows the average $\overline{\mathbb{T}}_{sync}$ for each simulation. Curve 2 shows the number of synchronization instances that happened in one simulation. The curve $E[\overline{\mathbb{T}}_{sync}]$ is the average of all the $\overline{\mathbb{T}}_{sync}$ which gives for 100 simulations and idea of how often the robots synchronize. The average is around 11.7 seconds. In the bottom of the Figure 5 curve 1 in dashed line shows the capture time (T^*) for each simulation and in solid line is shown the average of rescue times ($E[T^*] = \overline{T^*}$). If we now compare the plot in Figure 2(a) corresponding to $\mathbb{T}_{sync} = 10$, that is close to the average of the entropy-based case (11.7 seconds), we have that for the first case, with two pursuers, the average rescue time is around 74.2 seconds while in this case we have an average rescue time of 50.5 seconds. Using an entropy-based in average the rescue time is lower since only high content probability maps are shared.

7 Conclusions and Future Work

In this paper we discussed two important issues of a multi-robot system: localization errors of single robots and limitations in the communication between robots. We used the pursuit–evasion game to model a quite general class of multi-robot systems and we proposed an extension of a probabilistic approach for this kind of systems when there are localization errors. The probabilistic approach was natural to deal with the assumptions of imperfect localization, measurement noise, and inaccurate a priori obstacle map. Simulations showed that a multi-robot systems performs better than a single robot system, when it relies on perfect communication. We discussed a possible extension in the case of communication constraints, such as periodic communication. Some preliminary results using an entropy-based approach was also discussed. Our current research involves three extensions of the pursuit–evasion game for the multi-robot system described in the paper. The first task is to let the the communication channel be available only at *random* time instances, which would model a real situation, for example, when a WLAN is shared by many (unpredictable) users. The second is to find a good measure to compare probability maps calculated in different instants. The third task is to integrate localization errors in the communication model.

8 Acknowledgments

The authors gratefully acknowledge Joao P. Hespanha for the fruitful discussions about probabilistic pursuit–evasion problems and to provide us his simulation environment. We want also to thank Francesco F. Baldelli for the results of the simulations and the plots.

References

[1] T. Balch and R. C. Arkin. Communication in reactive multiagent robotic systems. *Autonomous Robots*, pages 27–52, 1994.

[2] T. Balch and R. C. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Trans. on Robotics and Automation*, 14:926–939, 1998.

[3] T. Basar and J. G. Olsder. *Dynamic noncooperative game theory.* Academic Press, 2nd edition, 1995.

[4] R. W. Beard, J. Lawton, and F. Y. Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Trans. on Control Systems Technology*, 9:777–790, 2001.

[5] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proc. of the 13th National Conf. on Artificial Intelligence, AAAI.* AAAI Press/MIT Press, 1996.

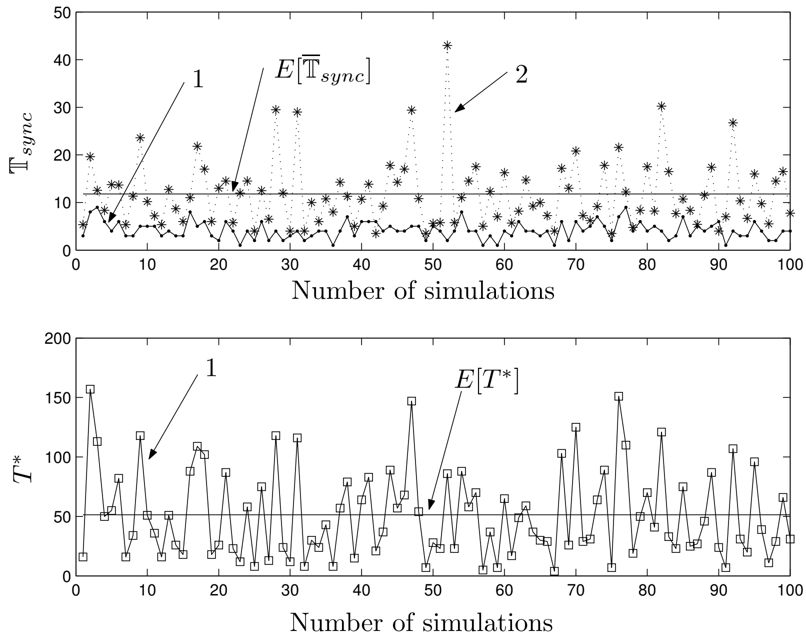


Figure 5: *Results of* 100 *simulations in the case of entropy-based synchronization. In the first plot curve 1 shows the average synchronization time per simulation, curve 2 shows the total number of synchronization per simulations and curve* $E[\overline{\mathbb{T}}_{sync}]$ *shows the average of* $\overline{\mathbb{T}}_{sync}$*. In the second plot is shown the rescue time (curve 1) per each simulation and the average rescue time* ($\overline{T}^* = E[T^*]$)*. The threshold for the (normalized) entropy was* 0.98 *bits.*

[6] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, 3:375–397, 1996.

[7] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Special issue of Autonomous Robots on Hetergeneous Multi-Robot Systems*, 3, 2000.

[8] J. Hespanha, H. J. Kim, and S. Sastry. Multiple-agent probabilistic pursuit-evasion games. In *In 38th Conf. on Decision and Control*, volume 3, pages 2432–2437, 1999.

[9] J. Hespanha, M. Prandini, and S. Sastry. Probabilistic pursuit-evasion games: A one-step nash approach. In *In 39th Conf. on Decision and Control*, pages 2272–2277, 2000.

[10] R. Isaac. *Differential Games*. Robert E. Kriger Publisher Company, 2 edition, 1967.

[11] P. Jensfelt. *Approaches to mobile robot localization in indoor enviroment*. PhD thesis, Royal Institute of Technology, 2001.

[12] J. Kim, R. Vidal, H. Shim, O. Shakernia, and S. Sastry. A hierarchical approach to probabilistic pursuit evasion games with unmanned ground and aerial vehicles. In *In 40th IEEE Conf. on Decision and Control*, pages 634–639, 2001.

[13] S. K. Mitter. Control with limited information: the role of systems theory and information theory. *IEEE Information Theory Society Newsletter*, 4(50):122–131, 2000.

[14] P. Ögren, M. Egerstedt, and H. Xiaoming. A control lyapunov function approach to multi-agent coordination. In *IEEE Conf. on Decision and Control*, 2001.

[15] E. Pagello, A. D'Angelo, F. Montesello, and C. Ferrari. Implicit coordination in a multi-agent system using a behaviour-based approach. In *In Distributed Autonomous Robot Systems 4 (DARS98)*, 1998.

[16] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Transaction on Automatic Control*, 24(6):843–854, 1979.

[17] A. Saffiotti and E. H. Ruspini. Global team coordination by local computation. In *Proc. of the European Control Conference (ECC)*, 2001.

[18] A. Saffiotti, N. B. Zumel, and E. H. Ruspini. Multirobot team coordination using desirabilities. In *Proc. of the 6th Intl. Conf. on Intelligent Autonomous Systems (IAS)*, pages 107–114, 2000.

[19] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423,623–656, 1948.

[20] L. D. Stone, C. A. Barlow, and T. L. Corwin. *Bayesian Multiple Target Traking*. Artech House, 1999.

[21] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots(joint issue)*, 31:29–53, 1998.