Proceedings of the 18th World Congress
The International Federation of Automatic Control
Milano (Italy) August 28 - September 2, 2011

Probabilistically Safe Vehicle Control in a Hostile Environment [★]

Igor Cizelj [*] **Xu Chu (Dennis) Ding** [**] **Morteza Lahijanian** [**]
Alessandro Pinto [***] **Calin Belta** [**]

[*] *Division of Systems Engineering, Boston University, Boston, MA 02215, USA. (e-mail: icizelj@bu.edu)*
[**] *Department of Mechanical Engineering, Boston University, Boston, MA 02215, USA. (e-mail: {xcding; morteza; cbelta}@bu.edu)*
[***] *Embedded Systems and Networks Group, United Technologies Research Center Inc., Berkeley, CA (e-mail: alessandro.pinto@utrc.utc.com)*

Abstract: In this paper we present an approach to control a vehicle in a hostile environment with static obstacles and moving adversaries. The vehicle is required to satisfy a mission objective expressed as a temporal logic specification over a set of properties satisfied at regions of a partitioned environment. We model the movements of adversaries in between regions of the environment as Poisson processes. Furthermore, we assume that the time it takes for the vehicle to traverse in between two facets of a region is exponentially distributed, and we obtain the rate of this exponential distribution from a simulator of the environment. We capture the motion of the vehicle and the vehicle updates of adversaries distributions as a Markov Decision Process. Using tools in Probabilistic Computational Tree Logic, we find a control strategy for the vehicle that maximizes the probability of accomplishing the mission objective. We demonstrate our approach with illustrative case studies.

Keywords: Autonomous vehicles, Markov Decision Processes, Probabilistic risk assessment

1. INTRODUCTION

Robot motion planning and control has been widely studied in the last twenty years. Recently, temporal logics, such as Linear Temporal Logic (LTL) and Computational Tree Logic (CTL) have become increasingly popular for specifying robotic tasks (see, for example, [Conner et al., 2007, Karaman and Frazzoli, 2008, Kloetzer and Belta, 2008b, Loizou and Kyriakopoulos, 2004]). It has been shown that temporal logics can serve as rich languages capable of specifying complex mission tasks such as "go to region A and avoid region B unless regions C or D are visited".

Many of the above-mentioned works that use a temporal logic as a specification language rely on the assumption that the motion of the robot in the environment can be abstracted to a finite transition system by partitioning the environment. The transition system must be finite in order to allow the use of existing model-checking tools for temporal logics (see [Baier et al., 2008]). Furthermore, it is assumed that the resultant transition system obtained from the abstraction process is deterministic (*i.e.*, an available control action deterministically triggers a unique transition from one region of the environment to anther region), and the environment is static. To address environments with dynamic obstacles, [Kress-Gazit et al., 2007, Topcu et al., 2009] find control strategies that guarantee satisfactions of specifications by playing temporal logic games with the environment.

In practice, due to actuator and sensor noise, a deterministic transition system may not adequately represent the motion of the robot. [Kloetzer and Belta, 2008a] proposed a control strategy for a purely non-deterministic transition system (*i.e.*, a control action enables multiple possible transitions to several regions of the environment). [Lahijanian et al., 2010] pushed this approach a step further by modeling the motion of the robot as a Markov Decision Process (MDP) (*i.e.*, a control action triggers a transition from one region to anther with some fixed and known probability). The transition probabilities of this MDP can be obtained from empirical measurements or an accurate simulator of the environment. A control strategy was then derived to satisfy a mission task specified in Probabilistic Computational Tree Logic (PCTL) with the maximum probability.

In this paper, we extend this approach to control a vehicle in a dynamic and threat-rich environment with static obstacles and moving adversaries. We assume that the environment is partitioned into polygonal regions, and a high level mission objective is given over some properties assigned to these regions. The main contribution of this paper is an approach to design a reactive control strategy that provides probabilistic guarantees of accomplishing the mission in a threat-rich environment. This control strategy is reactive in the sense that the control of the vehicle

[★] This work was supported in part by grants ONR-MURI N00014-09-1051, ARO W911NF-09-1-0088, AFOSR YIP FA9550-09-1-020, NSF CNS-0834260, and the United Technologies Research Center.

978-3-902661-93-7/11/$20.00 © 2011 IFAC 10.3182/20110828-6-IT-1002.03248

is updated whenever the vehicle reaches a new region in the environment or it observes movements of adversaries. In order to solve this problem, we capture the motion of the vehicle, as well as vehicle estimates of the adversary distributions as a MDP. This way, we map the vehicle control problem to the problem of finding a control policy for an MDP such that the probability of satisfying a PCTL formula is maximized. For latter, we use our previous approach presented in [Lahijanian et al., 2010].

Due to space limitations preliminaries are not included in this paper. We refer readers to [Baier et al., 2008, Ross, 2006] for information about MDPs and to [Baier et al., 2008, Lahijanian et al., 2010] for detailed description of PCTL. Further insight into our approach can be found in the technical report [Cizelj et al., 2011]. Furthermore, [Cizelj et al., 2011] analyzes computational complexity of our approach.

2. PROBLEM FORMULATION AND APPROACH

We consider a city environment that is partitioned into a set of polytopic regions R. We assume the partition[1] is such that adjacent regions in the environment share exactly one facet. We denote F as the set of facets of all polytopes in R. We assume that one region $r_p \in R$ is labeled as the "pick-up" region, and another region $r_d \in R$ is labeled as the "drop-off" region. Fig. 1 shows an example of a partitioned city environment. We assume that there is a vehicle moving in the environment. We require this vehicle to carry out the following mission objective:

Mission Objective: Starting from an initial facet $f_{init} \in F$ in a region $r_{init} \in R$, the vehicle is required to reach the pick-up region r_p to pick up a load. Then, the vehicle is required to reach the drop-off region r_d to drop-off the load.

We consider a threat-rich environment with dynamic adversaries and static obstacles in some regions. The probability of safely crossing a region depends on the number of adversaries and the obstacles in that region. We say that the vehicle is lost in a region if it fails to safely cross the region (and thus fails the mission objective).

Let integers M_r and N_r be the minimum and maximum number of adversaries in region $r \in R$, respectively. We define

$$p_r^{init} : \{M_r, \ldots, N_r\} \to [0,1] \tag{1}$$

as a given (initial) probability mass function for adversaries in region $r \in R$, *i.e.* $p_r^{init}(n)$ is the probability of having n adversaries in region r and $\sum_{n=M_r}^{N_r} p_r^{init}(n) = 1$. However, adversaries may move in between regions. We model the movements of adversary in a region by arrivals of customers in a queue. Thus we consider the movements of adversary as Poisson processes and we assume that the time it takes for an adversary to leave and enter region r is exponentially distributed with rate $\mu_l(r)$ and $\mu_e(r)$, respectively. We further assume that adversaries move independent of each other, and at region r, the distributions of adversaries in adjacent regions of r depend only on the adversaries in r and the movements of adversaries between r and its adjacent regions.

In addition, each region has an attribute that characterizes the presence of obstacles, which we call obstacle density. We define

$$p_r^o : \{0, 1, \ldots, N_r^o\} \to [0,1], \tag{2}$$

as the probability mass function of the obstacle density in region $r \in R$, *i.e.*, $p_r^o(o)$ is the probability of having obstacle density o in region r and $\sum_{o=0}^{N_r^o} p_r^o(o) = 1$. Unlike adversaries, we assume that obstacles can not move in between regions.

We assume that the vehicle has a map of the environment and can detect its current region. When the vehicle enters a region, it observes the number of adversaries and the obstacle density in this region. When the vehicle is traversing inside a region, it detects movements of adversaries between the current region and its adjacent regions.

The motion capability of the vehicle in the environment is limited by a (not necessarily symmetric) relation $\Delta \subseteq F \times F$, with the following meaning: If the vehicle is at a facet $f \in F$ and $(f, f') \in \Delta$, then it can use a motion primitive to move from f towards f' (without passing through any other facet), *i.e.,* Δ represents a set of motion primitives for the vehicle. The control of the vehicle is represented by $(f, f') \in \Delta$, with the meaning that at facet f, f' is the next facet the vehicle should move towards. Fig. 1 shows possible motions of the vehicle in this environment. We assume that the time it takes for the vehicle to move from facet f to facet f' is exponentially distributed with rate $\lambda(\delta)$, where $\delta = (f, f') \in \Delta$. This assumption is based on results from a simulator of the environment (see Sec. 5).

During the time when the vehicle is executing a mission primitive (f, f') (*i.e.,* moving between facet f and f'), we denote the probability of losing the vehicle as:

$$p_\delta^{lost} : \{M_r, \ldots, N_r\} \times \{0, \ldots, N_r^o\} \to [0,1], \tag{3}$$

where $\delta = (f, f') \in \Delta$, and r is the region bounded by f and f'. We obtain $p_\delta^{lost}(n, o)$ and $\lambda(\delta)$ from the simulator of the environment (see Sec. 5 for more details).

In this paper we aim to find a reactive control strategy for the vehicle. A vehicle control strategy at a region r depends on the facet f through which the vehicle entered r. It returns the facet f' the vehicle should move towards, such that $(f, f') \in \Delta$. The control strategy is reactive in the sense that it also depends on the number of adversaries and the obstacle density observed when entering the current region, as well as the movements of adversaries in the current region. We are now ready to formulate the main problem we consider in this paper:

Problem: Consider the partitioned environment defined by R and F; initial facet and region f_{init} and r_{init}; the motion capability Δ of a vehicle; initial adversary and obstacle density distributions for each region p_r^{init} and p_r^o; the probability of losing the vehicle p_δ^{lost}; rate of adversaries $\mu_l(r)$ and $\mu_e(r)$; and rate of the vehicle $\lambda(\delta)$; Find the vehicle control strategy that maximizes the probability of satisfying the Mission Objective.

The key idea of our approach is to model the motion of the vehicle in the environment, as well as vehicle estimates of adversary distributions in the environment as an MDP. By capturing estimates of adversary distributions in this MDP, the vehicle updates the adversary distributions of its

[1] Throughout the paper, we relax the notion of a partition by allowing regions to share facets

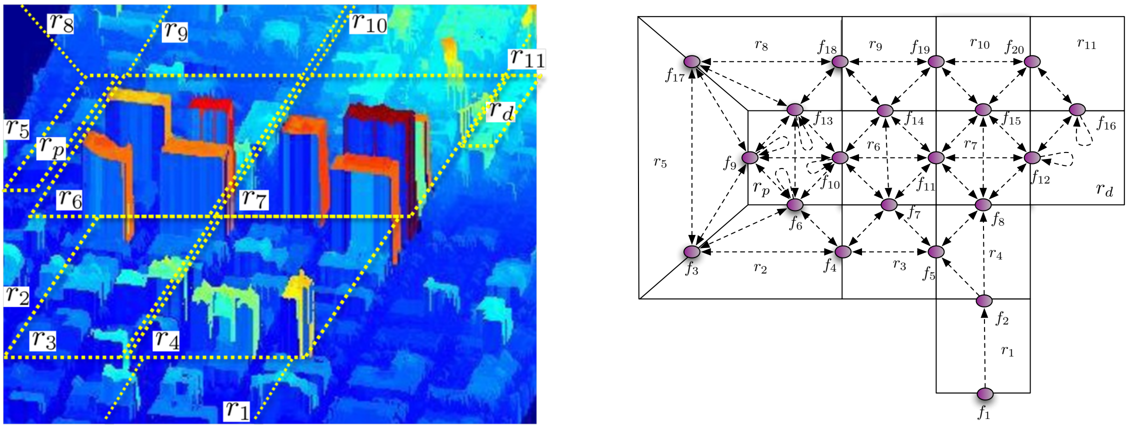

Fig. 1. Example of partitioned city environment. Left: A realistic scenario representing a city environment partitioned into regions. Right: Possible motion of the vehicle in the environment.

adjacent regions as it detects the movements of adversaries in the current region, and the control strategy produces an updated control if necessary. As a result, a policy for the MDP is equivalent to a reactive control strategy for the vehicle in the environment. We then translate the mission objective to a PCTL formula and find the optimal policy satisfying this formula with the maximum probability.

3. CONSTRUCTION OF AN MDP MODEL

3.1 Update of the adversary distributions

As adversaries enter and leave the current region, it is necessary to update the distributions of adversaries in adjacent regions. Because the vehicle can only observe the movements of adversaries in its current region, and due to the assumption that distributions of adversaries in adjacent regions depend only on the current region and its adjacent regions, it is only necessary to update the adversary distributions for adjacent regions, and not for all regions in the environment. Our MDP model captures all possible adversary distributions of adjacent regions at each region.

Let us denote the distribution for region r as p_r. The initial adversary distribution of region r is given in Eq. 1. Thus, the adversary distribution of region r is a probability mass function $p_r : \{M, \dots, N\} \to [0, 1]$, where $M_r \leq M \leq N \leq N_r$. Note that if $M = N = N_r$, then $p_r(N_r) = 1$ and no adversary may enter region r, or else the assumption that N_r is the maximum number of adversaries in region r would be violated. Similarly, if $M = N = M_r$, then no adversary may leave region r.

Given the current adversary distribution p_r, assuming that an adversary has entered region r ($p_r(N_r) \neq 1$), then we define the updated distribution as p_r^+ in the following way:

$$p_r^+ : \begin{cases} \{M+1,\dots,N\} \to [0,1] & \text{if } N = N_r \\ \{M+1,\dots,N+1\} \to [0,1] & \text{if } N < N_r, \end{cases} \tag{4}$$

such that:

$$p_r^+(n) = \begin{cases} p_r(n-1) + \dfrac{p_r(N)}{N-M} & \text{if } N = N_r \\ p_r(n-1) & \text{if } N < N_r. \end{cases} \tag{5}$$

If $N = N_r$, given that an adversary entered region r, we can conclude that the previous number of adversaries cannot be N_r, thus we evenly redistribute the probability associated with N_r before an adversary entered the region. If $N < N_r$, the probability distribution simply shifts by 1.

Similarly, assuming that an adversary has left region r, then $p_r(M_r) \neq 1$ and we define the updated distribution as p_r^- in the following way:

$$p_r^- : \begin{cases} \{M,\dots,N-1\} \to [0,1] & \text{if } M = M_r \\ \{M-1,\dots,N-1\} \to [0,1] & \text{if } M > M_r, \end{cases} \tag{6}$$

such that:

$$p_r^-(n) = \begin{cases} p_r(n+1) + \dfrac{p_r(M)}{N-M} & \text{if } M = M_r \\ p_r(n+1) & \text{if } M > M_r. \end{cases} \tag{7}$$

Given p_r, it is easy to verify that $p_r^+ : \{M, \dots, N\} \to [0, 1]$ is a valid probability mass functions, *i.e.* $\sum_{n=M}^{N} p_r^+(n) = 1$ (similarly for p_r^-). Starting with the initial distribution p_r^{init}, we can use Eq. (4)-(7) to determine D_r, the set of all possible distributions for region r.

3.2 MDP construction

Let us denote $B \subseteq F \times R$ as the boundary relation where $(f, r) \in B$ if and only if f is a facet of region r. We denote the set of regions adjacent to region r as $A_r = \{r_1, \dots, r_m\} \subset R$.

We define a labeled MDP $\mathcal{M}$ as a tuple $(S, s_0, Act, A, P, \Pi, h)$ where: S is the finite set of states such that $S = \bigcup_{r \in R} \{\{(f, z) \in B | z = r\} \times \{M_r, \dots, N_r\} \times \{0, 1, \dots, N_r^o\} \times \{\text{lost,alive}\} \times \prod_{r' \in A_r} D_{r'}\}$. The meaning of the state is as follows: $((f, r), n, o, \text{alive}, p_{r_1}, \dots, p_{r_m})$ means that the vehicle is at facet f, heading towards region r, and in region r there are n adversaries, o obstacles, the vehicle is currently not lost, and the adversary distribution for the adjacent region $r_i \in A_r = \{r_1, \dots, r_m\}$ is p_{r_i}. $((f, r), n, o, \text{lost}, p_{r_1}, \dots, p_{r_m})$ means that the vehicle did not make it to facet f because it was lost in the previous region while heading towards f; $s_0 = ((f_{init}, r_{init}), 0, 0, \text{alive}, p_{r'_1}^{init}, \dots, p_{r'_k}^{init})$ is the initial state, where $A_{r_{init}} = \{r'_1, \dots, r'_k\}$; $Act = \Delta \cup \tau$ is the set of actions, where τ is a dummy action when the vehicle is lost; $A : S \to 2^{Act}$ is a function specifying the enabled actions at a state s and is defined as follows: If the vehicle is alive, then $A(s) = \{(f, f') \in \Delta\}$, otherwise $A(s) = \tau$; $P : S \times Act \times S \to [0, 1]$ is a transition probability function such that for all states $s \in S$ and actions $a \in A(s)$: $\sum_{s' \in S} P(s, a, s') = 1$, and for all actions $a \notin A(s)$ and $s' \in S$, $P(s, a, s') = 0$. We describe how we generate the transition probability function P in Sec. 3.3; $\Pi = \{r_p, r_d, \text{alive}\}$ is the set of properties; $h : S \to 2^{\Pi}$ is a function that assigns some properties in Π to each state of $s \in S$. We define h as follows: If $s = ((f, r), n, o, b, p_1, \ldots, p_m)$, then $\{\text{alive}\} \in h(s)$ if and only if $b = \text{alive}$, $\{r_p\} \in h(s)$ if and only if $r = r_p$, and $\{r_d\} \in h(s)$ if and only if $r = r_d$.

As the vehicle moves in the environment, it updates its corresponding state on $\mathcal{M}$ when: 1) it reaches a facet f and enters a region r, and observes the number of adversary n and obstacle density o in region r, then it updates its state to $((f, r), n, o, \text{alive}, p^{init}_{r_1}, \ldots, p^{init}_{r_m})$; 2) an adversary leaves the current region r and moves into region r', given the current adversary distribution of region r' as $p_{r'}$, the vehicle updates this distribution to $p^{+}_{r'}$; 3) an adversary enters the current region r from region r', given the current adversary distribution of region r' as $p_{r'}$, the vehicle updates this distribution to $p^{-}_{r'}$;

Since actions of $\mathcal{M}$ consists of Δ, $\mathcal{M}$ is designed so that its control policy (for details see [Baier et al., 2008], but roughly control policy is a function that specifies for every finite sequence of states of $\mathcal{M}$, the next action to be applied) can be directly translated to a reactive control strategy for the vehicle. When the vehicle updates its state in $\mathcal{M}$, then the action $\delta \in \Delta$ at its current state determines the next facet the vehicle should move towards.

3.3 Generating the transition probability function P

First, we define a random variable e for the time in between a vehicle entering the current region r at facet f, heading towards facet f' and an event occurring, which can be: 1) an adversary entering the current region; 2) an adversary leaving the current region; or 3) the vehicle reaching facet f'. Note that if $X_1, \ldots, X_n$ are independent exponentially distributed random variables with rate parameters $\lambda_1, \ldots, \lambda_n$, then $min\{X_1, \ldots, X_n\}$ is exponentially distributed with parameter $\lambda = \sum_{i=1}^{n} \lambda_i$. The probability that X_k is the minimum is $Pr(X_k = min\{X_1, \ldots, X_n\}) = \frac{\lambda_k}{\lambda}$. By assumption, movements of adversaries are independent of each other. Since the arrival and departure of adversaries in the current region are modeled as two Poisson processes with inter-arrival and inter-departure time exponentially distributed with rate $\mu_e(r)$ and $\mu_l(r)$, respectively, and the time required for the vehicle to reach facet f' is exponentially distributed with rate $\lambda(\delta)$, where $\delta = (f, f')$, the random variable e is also exponentially distributed. We assume e is exponentially distributed with rate ν.

Since the vehicle can not detect the exact number of adversaries in adjacent regions, only an estimated value ν_e of ν can be obtained from the expected number of adversaries in adjacent regions. Let us denote E_r as the expected value for the distribution p_r. Assume the current state as $((f, r), n, o, \text{alive}, p_{r_1}, \ldots, p_{r_m})$. If an adversary can leave current region r (*i.e.* $n > M_r$ and $C_r \neq \emptyset$, where $C_r \subseteq A_r$ is the set of adjacent regions to which an adversary can enter) then the time it takes for an adversary to leave region r is exponentially distributed with rate $\mu_l(r)n$ because there are n adversaries in the region and any of them can leave region r. Similarly, if an adversary can enter the current region r (*i.e.* $n < N_r$), and there exists an adversary that can leave an adjacent region (*i.e.* $B_r \neq \emptyset$, where $B_r \subseteq A_r$ is the set of adjacent regions from which an adversary can leave), then the time it takes for an adversary to enter region r is exponentially distributed with the estimated rate $\mu_e(r) \sum_{r' \in B_r} E_{r'}$, where $\sum_{r' \in B_r} E_{r'}$ gives the total expected number of adversaries that can enter region r. The time it takes for the vehicle to reach facet f' is exponentially distributed with rate $\lambda(\delta)$. Therefore, the estimated rate ν_e can be obtained as:

$$\nu_e = \lambda(\delta) + \mu_l(r) n \mathbb{I}_l(A_r, n) + \mu_e(r) \sum_{r' \in B_r} E_{r'} \mathbb{I}_e(n) \qquad (8)$$

where n is the number of adversaries in the current region; $\mathbb{I}_l(A_r, n) = 0$ when $n = M_r$ or $C_r = \emptyset$, and $\mathbb{I}_l(A_r, n) = 1$ otherwise; and $\mathbb{I}_e(n) = 0$ if $n = N_r$, and $\mathbb{I}_e(n) = 1$ otherwise. The rate ν_e will be used to generate the probability transition function P.

We define the probability transition function $P : S \times Act \times S \to [0, 1]$ as follows: Let $s = ((f, r), n, o, \text{alive}, p_{r_1}, \ldots, p_{r_m})$ with $\{r_1, \ldots, r_m\} \in A_r$.

- If $s' = ((f', r'), n', o', b', p^{init}_{r'_1}, \ldots, p^{init}_{r'_k})$, with $\{r'_1, \ldots, r'_k\} \in A_{r'}$, $\delta = (f, f') \in \Delta$ and $r' \in A_r$, then: $P(s, \delta, s') =$

$$\begin{cases} \dfrac{\lambda(\delta)}{\nu_e} p_{r'}(n') p^{o}_{r'}(o')(1 - p^{lost}_{\delta}(n, o)), & \text{if } b' = \text{alive} \\ \dfrac{\lambda(\delta)}{\nu_e} p_{r'}(n') p^{o}_{r'}(o') p^{lost}_{\delta}(n, o), & \text{if } b' = \text{lost.} \end{cases}$$

Under the action (f, f'), the transition from state s to s' indicates that either the vehicle reaches facet f' (s' is an “alive” state) or the vehicle is lost while traversing the region r (s' is a “lost” state). Let us first consider the former case. $\frac{\lambda(\delta)}{\nu_e}$ corresponds to the probability that the vehicle reaches facet f' before any adversary entering or leaving region r. $p_{r'}(n')$ corresponds to the probability of observing n' adversaries in region r' when entering region r'. $p^{o}_{r'}(o')$ corresponds to the probability of observing obstacle density o' for region r'. $(1 - p^{lost}_{\delta}(n, o))$ corresponds to the probability of safely crossing the current region with n adversaries and obstacle density o. Since each of these events are independent with each other, the probability of transition is the multiplication of the above probabilities. The same reasoning applies to the latter case, where $(1 - p^{lost}_{\delta}(n, o))$ is replaced by $p^{lost}_{\delta}(n, o)$ as the probability of losing the vehicle while crossing region r.

- If $s' = ((f, r), n + 1, o, \text{alive}, p'_{r_1}, \ldots, p'_{r_m})$, with $\delta = (f, f') \in \Delta$ for some f', $p_{r_i} = p'_{r_i}$ for all $i = \{1, \ldots, m\} \backslash \{j\}$ and $p'_{r_j} = p^{-}_{r_j}$ for some j, then: $P(s, \delta, s') = \frac{\mu_e(r) E_{r_j}}{\nu_e}$.

The transition from state s to s' indicates that an adversary from region r_j enters the current region before the vehicle reaches facet f' or an adversary moves in between the current region and another adjacent region. Thus, the adversary distribution of region r_j is updated to $p'_{r_j} = p^{-}_{r_j}$.

- If $s' = ((f, r), n - 1, o, \text{alive}, p'_{r_1}, \ldots, p'_{r_m})$, with $\delta = (f, f') \in \Delta$ for some f', $p_{r_i} = p'_{r_i}$ for all $i = \{1, \ldots, m\} \backslash \{j\}$ and $p'_{r_j} = p^{+}_{r_j}$ for some j, then: $P(s, \delta, s') = \frac{\mu_l(r) n}{\nu_e |C_r|}$.

The transition from the state s to s' indicates that an adversary leaves the current region and enters region r_j before the vehicle reaches facet f' or an adversary enters the current region. Thus, the adversary distribution of region r_j is updated to $p'_{r_j} = p^+_{r_j}$.

- If $s = ((f, r), n, o, \text{lost}, p_{r_1}, \ldots, p_{r_m})$, then $P(s, \tau, s) = 1$. s corresponds to the case where the vehicle is lost, thus it self-loops with probability 1.
- Otherwise, $P(s, \delta, s') = 0$.

In the technical report [Cizelj et al., 2011] we prove that P is a valid transition probability function.

4. GENERATING THE OPTIMAL CONTROL POLICY AND A VEHICLE CONTROL STRATEGY

After obtaining the MDP model, we solve our proposed problem by using the PCTL control synthesis approach presented in [Lahijanian et al., 2010] by translating the problem to a PCTL formula. Formulas of PCTL are interpreted over states of an MDP and are constructed by connecting properties from a set Π using standard Boolean operators, the temporal operator $\mathcal{U}$ denoting "until", and the probabilistic operator $\mathcal{P}$. The Mission Objective is equivalent to the temporal logic statement "eventually reach r_p and then r_d while always staying alive", which can be translated to the following formula ϕ:

$$\mathcal{P}_{max=?}[\text{alive}\, \mathcal{U}\, (\text{alive} \wedge r_p \wedge \mathcal{P}_{>0}[\text{alive}\, \mathcal{U}\, (\text{alive} \wedge r_d)])]. \quad (9)$$

The PCTL control synthesis tool takes an MDP and a PCTL formula ϕ and returns the control policy that maximizes the probability of satisfying ϕ as well as the corresponding probability value. The tool is based on the off-the-shelf PCTL model-checking tool PRISM (see Kwiatkowska et al. [2004]). We use Matlab to construct the MDP $\mathcal{M}$, which together with ϕ is passed to the PCTL control synthesis tool. The output of the control synthesis tool is the optimal control policy that maximizes the probability of satisfying ϕ. This policy can be directly translated to the desired vehicle control strategy.

5. SIMULATOR OF THE ENVIRONMENT

We constructed a realistic test environment in order to obtain the probability p_δ^{lost} (Eq. 3) from existing data of the distribution of obstacles in each region, and values for rate of the vehicle, $\lambda(\delta), \delta \in \Delta$. This test environment consists of several components, which are shown in Fig. 2. In order to obtain p_δ^{lost}, we first generated the marginal probability $p_\delta^{lost}(o)$, $\delta = (f, f')$ as the probability of losing the vehicle while traversing region r from facet f to f' with obstacle density o. This probability depends on the motion planning algorithm for the vehicle traversing the region, and the ability of the vehicle to detect obstacles.

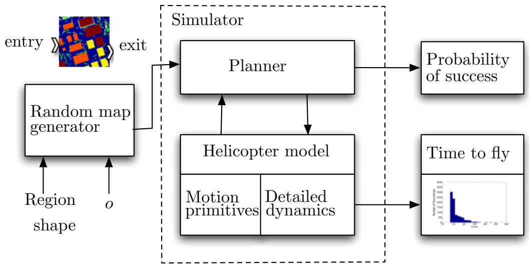

Fig. 2. Test environment used to compute the probability $p_\delta^{lost}(n, o)$ and the rate $\lambda(\delta)$.

We assumed that the obstacle data in the environment was accurate and that there was no need for real-time obstacle detection. We used a probabilistic road-map planner [LaValle, 2006, Frewen et al., 2011] to solve the following problem: given a starting point on a facet f and an ending point on the facet f', find a shortest collision free path between them. The planner uses a randomized algorithm that consists of building a random graph over the free space in the environment, and finding the shortest feasible collision-free path. Because of the randomized nature of the algorithm, there is a non-zero probability that a path can not be found by the planner even if one exists. This is the probability $p_\delta^{lost}(o)$ because it is the probability that the vehicle can not safely traverse from facet f to f'.

We computed $p_\delta^{lost}(o)$ using sampling (Fig. 2). Given the obstacle density o, we generated a random map by instantiating obstacles with random positions and sizes so that the density was o. The map was provided to the planner that generated a path. To implement the planner at the top of Fig. 2, we used the vehicle motion primitives defined in [Frazzoli et al., 2005]. The successes and failures for each path were recorded. When a feasible path was found, a standard model of the dynamics of a helicopter [Bullo and Lewis, 2004] was used to simulate a trajectory following the path and compute $\lambda(\delta)$.

We computed the joint probability $p_\delta^{lost}(n, o)$ as a combination of the marginal probabilities $p_\delta^{lost}(n)$ and $p_\delta^{lost}(o)$. The main reason for this approach was that while an accurate model is available to compute the probability of failing to traverse a region due to obstacles, the effect of adversaries is difficult to model and it is part of our future work. For the purposes of the case study in Sec. 6, we assumed the probability of losing the vehicle due to adversaries to be $p_\delta^{lost}(n) = 0.01(n)^2$ for $n \in [0, 10]$. After the marginal probabilities were obtained, we constructed the joint probability $p_\delta^{lost}(n, o)$ using the following formula (see [Nelsen, 2006]): $p_\delta^{lost}(n, o) = e^{-\sqrt{-log(p_\delta^{lost}(n)) - log(p_\delta^{lost}(o))}}$.

6. RESULTS

We considered the scenario together with the partitioned environment and the possible motion of the vehicle Δ shown on Fig. 1. The initial probability mass function for adversaries in region $r \in R$, p_r^{init}, and the probability mass function of the obstacle density in region $r \in R$, p_r^o, are given in Table 1. In addition, we assumed that there is no adversary or obstacle in region r_p and r_d. The probability $p_\delta^{lost}(n, o)$ and the rates of the vehicle $\lambda(\delta)$ for all $\delta \in A$ were obtained from the simulator. We used the following numerical values: $\lambda((f, f')) = 0.128$ when f and f' are facets of r_1 and r_5, $\lambda((f, f')) = 0.125$ when f and f' are facets of r_2, r_4, r_8, r_9, r_{10}, and r_{11}, and $\lambda((f, f')) = 0.091$ when f and f' are facets of r_3, r_6, and r_7 with $\mu_e(r) = \mu_l(r) = 0.05$ for all $r \in R$.

We obtained the vehicle control strategy through the method described in Sec. 4. Two vehicle runs are shown in Fig. 3, corresponding to case A and case B (Table 1). We found that the maximum probability of satisfying the specification ϕ (Eq. 9) for cases A and B to be 0.141 and 0.805, respectively. The substantial difference between these two maximum probabilities is due to the difference

Table 1. Obstacle density and adversary distribution

Region	Obstacle density	Adversary distribution	
		case A	case B
r_1	1%	$p^{init}_{r_1}(0) = 1$	$p^{init}_{r_1}(0) = 1$
r_2	3%	$p^{init}_{r_2}(x) = 1/3, x \in [7, 9]$	$p^{init}_{r_2}(x) = 1/3, x \in [2, 4]$
r_3	6%	$p^{init}_{r_3}(x) = 1/3, x \in [7, 9]$	$p^{init}_{r_3}(x) = 1/3, x \in [2, 4]$
r_4	5%	$p^{init}_{r_4}(x) = 1/3, x \in [1, 3]$	$p^{init}_{r_4}(x) = 1/3, x \in [2, 4]$
r_5	1%	$p^{init}_{r_5}(x) = 1/3, x \in [7, 9]$	$p^{init}_{r_5}(x) = 1/3, x \in [2, 4]$
r_6	9%	$p^{init}_{r_6}(x) = 1/3, x \in [7, 9]$	$p^{init}_{r_6}(x) = 1/3, x \in [2, 4]$
r_7	9%	$p^{init}_{r_7}(x) = 1/3, x \in [1, 3]$	$p^{init}_{r_7}(x) = 1/3, x \in [2, 4]$
r_8	3%	$p^{init}_{r_8}(x) = 1/3, x \in [1, 3]$	$p^{init}_{r_8}(x) = 1/3, x \in [2, 4]$
r_9	4%	$p^{init}_{r_9}(x) = 1/3, x \in [1, 3]$	$p^{init}_{r_9}(x) = 1/3, x \in [4, 6]$
r_{10}	4%	$p^{init}_{r_{10}}(x) = 1/3, x \in [1, 3]$	$p^{init}_{r_{10}}(x) = 1/3, x \in [4, 6]$
r_{11}	3%	$p^{init}_{r_{11}}(x) = 1/3, x \in [7, 9]$	$p^{init}_{r_{11}}(x) = 1/3, x \in [2, 4]$

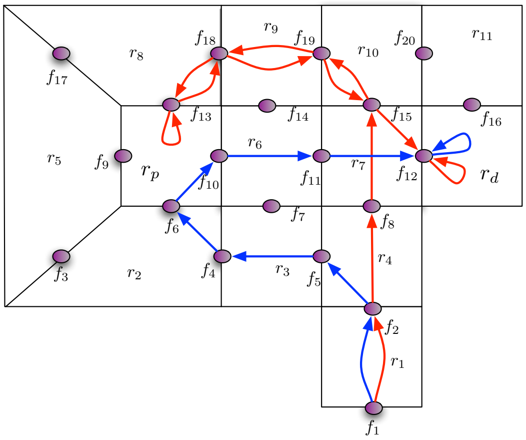

Fig. 3. Runs of the vehicle in the partitioned environment for the given mission scenario and the data. Two different adversary distributions are given in Table 1. The arrows represent movement of the vehicle in between facets. Red and blue arrows correspond to case A and case B, respectively.

in adversary distributions. A close analysis of the vehicle runs together with the adversary distributions shows that in case A the number of adversaries in regions r_2, r_3 and r_6 is high, which results in the vehicle control strategy that ensures that the vehicle avoids this regions.

For this particular case study, the MDP $\mathcal{M}$ had 1079 states. The Matlab code used to construct $\mathcal{M}$ ran for approximately 14 minutes on a MacBook Pro computer with a 2.5 GHz dual core processor. Furthermore, the time it took the control synthesis tool to generate optimal policy is 4 minutes.

7. CONCLUSIONS AND FINAL REMARKS

In this paper we provided an approach to obtain a reactive control strategy that provides probabilistic guarantees for achieving a mission objective in a threat-rich environment. We modeled the motion of the vehicle, as well as vehicle estimates of the adversary distributions as an MDP. We then found the optimal control strategy for the vehicle maximizing the probability of satisfying a given mission task specified as a PCTL formula.

Future work include extensions of this approach to a richer specification language such as probabilistic Linear Temporal Logic (PLTL) and a more general model of the vehicle in the environment such as a Partially Observed Markov Decision Process (POMDP).

REFERENCES

C. Baier, J. P. Katoen, and K. M. Larsen. *Principles of Model Checking*. MIT Press, 2008.

F. Bullo and A. D. Lewis. *Geometric Control of Mechanical Systems*, volume 49 of *Texts in Applied Mathematics*. Springer Verlag, New York-Heidelberg-Berlin, 2004.

I. Cizelj, Xu Chu Ding, M. Lahijanian, A. Pinto, and C. Belta. Probabilistically safe vehicle control in a hostile environment. Technical report, March 2011. URL `http://arxiv.org/abs/1103.4065`.

D. C. Conner, H. Kress-Gazit, H. Choset, A. Rizzi, and G. J. Pappas. Valet parking without a valet. In *Proceedings of 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 572–577, San Diego, CA, 2007.

E. Frazzoli, M. A. Dahleh, and E. Feron. Maneuver-based motion planning for nonlinear systems with symmetries. *IEEE Trans. on Robotics*, 2005.

T. A. Frewen, H. Sane, M. Kobilarov, S. Bajekal, and K. R. Chevva. Adaptive path planning in a dynamic environment using a receding horizon probabilistic roadmap. In *AHS International Specialists' Meeting*, 2011.

S. Karaman and E. Frazzoli. Vehicle routing problem with metric temporal logic specifications. In *IEEE Conf. on Decision and Control*, 2008.

M. Kloetzer and C. Belta. Dealing with nondeterminism in symbolic control. In *Hybrid Systems: Computation and Control: 11th International Workshop*, 2008a.

M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 2008b.

H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas. Where's waldo? sensor-based temporal logic motion planning. In *In IEEE International Conference on Robotics and Automation*, pages 3116–3121, 2007.

M. Kwiatkowska, G. Norman, and D. Parker. Probabilistic symbolic model checking with PRISM: A hybrid approach. *International Journal on Software Tools for Technology Transfer*, 6(2):128–142, 2004.

M. Lahijanian, J. Wasniewski, S. B. Andersson, and C. Belta. Motion planning and control from temporal logic specifications with probabilistic satisfaction guarantees. In *IEEE International Conference on Robotics and Automation*, pages 3227–3232, 2010.

S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006.

S. G. Loizou and K. J. Kyriakopoulos. Automatic synthesis of multi-agent motion tasks based on ltl specifications. In *43rd IEEE Conference on Decision and Control*, pages 153–158, 2004.

R. Nelsen. An introduction to copulas. Springer-Verlag New York, Inc., 2006.

S. Ross. *Introduction to Probability Models*. Academic Press, Inc., 2006.

U. Topcu, T. Wongpiromsarn, and R. M. Murray. Receding horizon temporal logic planning for dynamical systems. In *Proceedings of the 48th IEEE Conference on Decision and Control*, 2009.