

Probabilistically Safe Control of Noisy Dubins Vehicles

Igor Cizelj and Calin Belta

Abstract—We address the problem of controlling a stochastic version of a Dubins vehicle such that the probability of satisfying a temporal logic specification over a set of properties at the regions in a partitioned environment is maximized. We assume that the vehicle can determine its precise initial position in a known map of the environment. However, inspired by practical limitations, we assume that the vehicle is equipped with noisy actuators and, during its motion in the environment, it can only measure its angular velocity using a limited accuracy gyroscope. Through quantization and discretization, we construct a finite approximation for the motion of the vehicle in the form of a Markov Decision Process (MDP). We allow for task specifications given as temporal logic statements over the environmental properties, and use tools in Probabilistic Computation Tree Logic (PCTL) to generate an MDP control policy that maximizes the probability of satisfaction. We translate this policy to a vehicle feedback control strategy and show that the probability that the vehicle satisfies the specification in the original environment is bounded from below by the maximum probability of satisfying the specification on the MDP.

I. INTRODUCTION

In “classical” motion planning problems [LaV06], the specifications are usually restricted to simple primitives of the type “go from A to B and avoid obstacles”, where A and B are two regions of interest in some environment. Often this is not rich enough to describe a task of interest in practical applications. Recently, it has been shown that temporal logics, such as Linear Temporal Logic (LTL) and Computational Tree Logic (CTL), can serve as rich languages capable of specifying complex motion missions such as “go to region A and avoid region B unless regions C or D are visited” (see, for example, [KF08], [KB08b], [KGFP07]).

In order to use tools from formal verification and automata games [BK08] for motion planning and control, most of the works using temporal logics as specification languages assume that the motion of the vehicle in the environment can be modeled as a finite system. This usually takes the form of a finite transition system [CGP99] that is either deterministic (applying an available action triggers a unique transition [KB08b]) or nondeterministic (applying an available action can enable multiple transitions, with no information on their likelihoods [KB08a]). More recent results show that, if sensor and actuator noise models can be obtained through experimental trials, then the robot motion can be modeled as a Markov Decision Process (MDP), and probabilistic temporal logics, such as Probabilistic CTL (PCTL) and

Probabilistic LTL (PLTL), can be used for motion planning and control (see [LWAB10]).

However, robot dynamics are normally described by control systems with state and control variables evaluated over infinite domains. A widely used approach for temporal logic verification and control of such a system is through the construction of a finite abstraction ([TP06], [Gir07]). Even though recent works discuss the construction of abstractions for stochastic systems [JP09], [ADBS08], the existing methods are either not applicable to robot dynamics or are computationally infeasible given the size of the problem in most robotic applications.

In this paper, we provide a conservative solution to the problem of controlling a stochastic Dubins vehicle such that the probability of satisfying a temporal logic specification over a set of properties at the regions in a partitioned environment is maximized. Inspired by a realistic scenario of an indoor vehicle leaving its charging station, we assume that the vehicle can determine its precise initial position in a known map of the environment. The actuator noise is modeled as a random variable with an arbitrary continuous probability distribution supported on a bounded interval. Also, we assume that the vehicle is equipped with a limited accuracy gyroscope, which measures its angular velocity, as the only means of measurement available.

By discretization and quantization, we capture the motion of the vehicle as well as the position uncertainty as a finite state MDP. In this setup, the vehicle control problem is converted to the problem of finding a control policy for an MDP such that the probability of satisfying a PCTL formula is maximized. For the latter, we use the approach from [LWAB10]. By establishing a mapping between the states of the MDP and the sequences of measurements obtained from the gyroscope, we show that a policy for the MDP becomes equivalent to a feedback control strategy for the vehicle in the environment. Finally, we show that the probability that the vehicle satisfies the specification in the original environment is bounded from below by the maximum probability of satisfying the specification on the MDP.

The main contribution of this work lies in the application, since the result holds for a realistic vehicle with noisy actuators and a limited accuracy gyroscope. The method that we propose here is closely related to “classical” Dynamic Programming (DP) - based approaches [ASG07]. In these problems, the set of allowed specifications is restricted to reaching a given destination state, whereas our PCTL control framework allows for richer, temporal logic specifications and multiple destinations. In [SWT09], the authors solve the problem of reaching a given destination while avoiding

This work was partially supported by the ONR MURI under grant N00014-10-10952 and by the NSF under grants CNS-0834260 and CNS-1035588.

The authors are with the Division of Systems Engineering at Boston University, Boston, MA 02215, USA. Email: {icizelj, cbelta}@bu.edu.

obstacles using the Rapidly-exploring Random Tree (RRT) algorithm that takes into account local reachability, as defined by differential constraints. In our approach, in order to obtain an MDP, a tree is also constructed (see Sec. III-B), but we take into account reachability under uncertainty. Moreover, our approach produces a feedback control strategy and a lower bound on the probability of satisfaction, whereas the method presented in [SWT09] only returns a collision-free trajectory. In addition, these methods differ from our work since they require the precise state of the vehicle, at all times, whereas in our case, it is always uncertain.

Due to space limitations, preliminaries are not included in this paper. We refer readers to [BK08] for information about MDPs and to [BK08], [LWAB10] for detailed description of PCTL. Furthermore, results in this paper are stated without proof. Proofs of all results as well as additional remarks can be found in the technical report [CB12].

II. PROBLEM FORMULATION AND APPROACH

A Dubins vehicle ([Dub57]) is a unicycle with constant forward speed and bounded turning radius moving in a plane. In this paper, we consider a stochastic version of a Dubins vehicle, which captures actuator noise:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ u + \varepsilon \end{bmatrix}, \quad u \in U, \quad (1)$$

where $(x, y) \in \mathbb{R}^2$ and $\theta \in [0, 2\pi)$ are the position and orientation of the vehicle in a world frame, u is the control input, U is the control constraint set, and ε is a random variable modeling the actuator noise. We assume that ε has an arbitrary continuous probability distribution supported on the bounded interval $[-\varepsilon_{max}, \varepsilon_{max}]$. The forward speed is normalized to 1 and ρ is the minimum turn radius. We denote the state of the system by $q = [x, y, \theta]^T \in SE(2)$.

As it will become clear later, the control strategy proposed in this paper works for any finite set of controls U . However, motivated by the fact that the optimal Dubins paths use only three inputs ([Dub57]), we assume $U = \{-1/\rho, 0, 1/\rho\}$, but we make no assumptions on the optimality. We define $W = \{u + \varepsilon | u \in U, \varepsilon \in [-\varepsilon_{max}, \varepsilon_{max}]\}$ as the set of applied control inputs, i.e., the set of angular velocities that are applied to the system in the presence of noise. We assume that time is uniformly discretized (partitioned) into stages (intervals) of length Δt , where stage k is from $(k-1)\Delta t$ to $k\Delta t$. The duration of the motion is finite and it is denoted by $K\Delta t$.¹ We denote the control input and the applied control input at stage k as $u_k \in U$ and $w_k \in W$, respectively.

We assume that the noise ε is piece-wise constant, i.e., it can only change at the beginning of a stage. This assumption is motivated by practical applications, in which a servo motor is used as an actuator for the turning angle (see e.g., [Maz04]). This implies that the applied control is also piece-wise constant, i.e., $w : [(k-1)\Delta t, k\Delta t] \rightarrow W$, $k = 1, \dots, K$,

¹Since PCTL has infinite time semantics, after $K\Delta t$ the system remains in the state achieved at $K\Delta t$. In [CB12] we explain how to determine K .

is constant over each stage. We assume that the vehicle is equipped with only one sensor, which is a limited accuracy gyroscope.

Gyroscope model: At stage k , gyroscope returns the measured interval $[\underline{w}_k, \bar{w}_k] \subset [u_k - \varepsilon_{max}, u_k + \varepsilon_{max}]$ containing the applied control input. Motivated by practical applications, we assume that the measurement resolution of the gyroscope, i.e., the length of $[\underline{w}_k, \bar{w}_k]$, is constant, and we denote it by $\Delta\varepsilon$. For simplicity of presentation, we also assume that $n\Delta\varepsilon = 2\varepsilon_{max}$, for some $n \in \mathbb{Z}^+$. Let $\underline{\varepsilon}_i = -\varepsilon_{max} + (i-1)\Delta\varepsilon$ and $\bar{\varepsilon}_i = -\varepsilon_{max} + i\Delta\varepsilon$, $i = 1, \dots, n$. Then, $[-\varepsilon_{max}, \varepsilon_{max}]$ can be partitioned² into n intervals: $[\underline{\varepsilon}_i, \bar{\varepsilon}_i]$, $i = 1, \dots, n$. We denote the set of all noise intervals $\mathcal{E} = \{[\underline{\varepsilon}_1, \bar{\varepsilon}_1], \dots, [\underline{\varepsilon}_n, \bar{\varepsilon}_n]\}$. At stage k , if the applied control input is $u_k + \varepsilon$, the gyroscope will return measured interval $[\underline{w}_k, \bar{w}_k] = [u_k - \underline{\varepsilon}, u_k + \bar{\varepsilon}]$, where $\varepsilon \in [\underline{\varepsilon}, \bar{\varepsilon}] \in \mathcal{E}$.

The vehicle moves in a planar environment in which a set of regions of interest, denoted R , is present. Let $R_u \subset R$ denote a set of unsafe regions, and $R_s = R \setminus R_u$ define the set of safe regions. One set of regions $R_p \subset R_s$ is labeled with “pick-up”, and another set $R_d \subset R_s$ is labeled with “drop-off”. In this work, we assume that the motion specification is as follows:

Specification 1: “Starting from an initial state q_{init} , the vehicle is required to reach a pick-up region to pick up a load. Then, the vehicle should go to a drop-off region to drop off the load. At all times, the vehicle should avoid the unsafe regions.”

We assume that the vehicle can precisely determine its initial state $q_{init} = [x_{init}, y_{init}, \theta_{init}]^T$ in a known map of the environment. While the vehicle moves, gyroscope measurements $[\underline{w}_k, \bar{w}_k]$ are available at each stage k . We define a *vehicle control strategy* as a map that takes as input a sequence of measured intervals $[\underline{w}_1, \bar{w}_1][\underline{w}_2, \bar{w}_2] \dots [\underline{w}_{k-1}, \bar{w}_{k-1}]$ and returns the control input $u_k \in U$ at stage k . We are ready to formulate the main problem that we consider in this paper:

Problem 1: Given a partitioned environment R , a vehicle model described by Eqn. (1) with initial state q_{init} , a motion task in the form of Specification 1, find a vehicle control strategy that maximizes the probability of satisfying the specification.

In this paper, we develop an approximate solution to Problem 1. By discretizing the noise interval, we define a finite subset of the set of possible applied control inputs. We use this to define a Quantized System (QS) that approximates the original system given by Eqn. (1). Next, we capture the uncertainty in the position of the vehicle and map the QS to an MDP. Finally, we translate the specification to a PCTL formula and find an MDP control policy that satisfies the formula with the maximum probability. This policy is mapped to a vehicle control strategy and we show that the probability that the original system under the obtained control strategy satisfies ϕ is bounded from below by the obtained maximum probability.

²Throughout the paper, we relax the notion of a partition by allowing the endpoints of the intervals to overlap.

III. APPROXIMATION

A. Quantized System

Let $q : [0, K\Delta t] \rightarrow SE(2)$ be the state trajectory. We use $q_k(t)$ and w_k , $t \in [(k-1)\Delta t, k\Delta t]$, $k = 1, \dots, K$ to denote the state trajectory and the constant applied control at stage k , respectively. With a slight abuse of notation, we use q_k to denote the end of state trajectory $q_k(t)$, i.e., $q_k = q_k(k\Delta t)$. Given a state q_{k-1} , the state trajectory $q_k(t)$ can be derived by integrating the system given by Eqn. (1) from the initial state q_{k-1} , and taking into account that the applied control is constant and equal to w_k . Throughout the paper, we will also denote this trajectory by $q_k(q_{k-1}, w_k, t)$, when we want to explicitly capture the initial state q_{k-1} and the constant applied control w_k .

In this paper, we assume ε has uniform continuous probability distribution supported on the bounded interval $[-\varepsilon_{max}, \varepsilon_{max}]$, but our approach is general in the sense that it holds for any continuous distribution supported on a bounded interval. For the uniform distribution the following holds:

$$\Pr(\varepsilon \in [\underline{\varepsilon}_i, \bar{\varepsilon}_i]) = \frac{|\bar{\varepsilon}_i - \underline{\varepsilon}_i|}{2\varepsilon_{max}} = \frac{\Delta\varepsilon}{2\varepsilon_{max}} = \frac{1}{n}, \quad (2)$$

$[\underline{\varepsilon}_i, \bar{\varepsilon}_i] \in \mathcal{E}$, $i = 1, \dots, n$.

For each interval in \mathcal{E} we define a representative value $\varepsilon_i = \frac{\underline{\varepsilon}_i + \bar{\varepsilon}_i}{2}$, $i = 1, \dots, n$. i.e., ε_i is the midpoint of interval $[\underline{\varepsilon}_i, \bar{\varepsilon}_i]$. We denote the set of all representative values as $E = \{\varepsilon_1, \dots, \varepsilon_n\}$. We define $W_d = \{u + \varepsilon \mid u \in U, \varepsilon \in E\} \subset W$ as a finite set of applied control inputs. Also, let $\omega : U \rightarrow W_d$ be a random variable, where $\omega(u) = u + \varepsilon$ with the probability mass function $p_\omega(\omega(u) = u + \varepsilon) = \frac{1}{n}$, $\varepsilon \in E$. This probability follows from Eqn. (2) since $\varepsilon \in E$ is the representative value of interval $[\underline{\varepsilon}_i, \bar{\varepsilon}_i] \in \mathcal{E}$. Finally, we define a Quantized System (QS) that approximates the original system as follows: The set of applied control inputs in QS is W_d ; for a state q_{k-1} and a control input $u_k \in U$, QS returns

$$q_k(q_{k-1}, \omega(u_k), t) = q_k(q_{k-1}, u_k + \varepsilon, t) \quad (3)$$

with probability $\frac{1}{n}$, where $\varepsilon \in E$.

B. Reachability graph

We denote $u_1 u_2 \dots u_K$, in which each $u_k \in U$ gives a control input at stage k , as a finite sequence of control inputs of length K . We use Σ_K to denote the set of all such sequences. For the initial state q_{init} and Σ_K , we define the reachability graph $G_K(q_{init})$ (see [LaV06] for a related definition), which encodes the set of all state trajectories originating from q_{init} that can be obtained, with a positive probability, by applying sequences of control inputs from Σ_K according to QS given by Eqn. (3). In Fig. 1 we give an example of a reachability graph.

IV. POSITION UNCERTAINTY

Since the specification is a statement about the regions in the environment, in order to answer whether some state trajectory satisfies the specification, it is sufficient to know its projection in \mathbb{R}^2 . Therefore, we focus only on the position uncertainty.

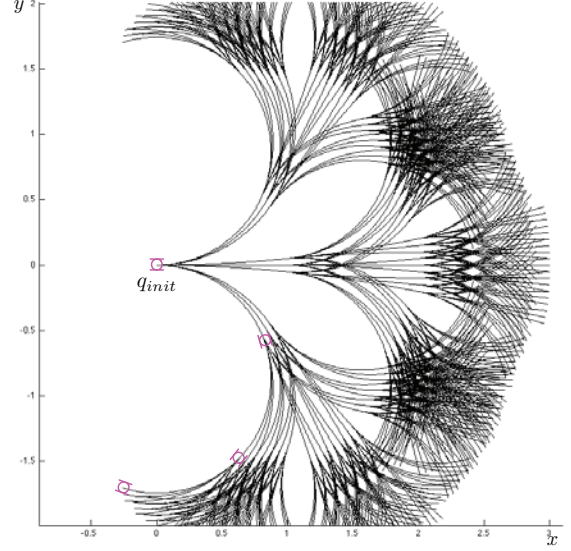


Fig. 1. The projection of reachability graph $G_3(q_{init})$ in \mathbb{R}^2 when $U = \{-\frac{\pi}{3}, 0, \frac{\pi}{3}\}$ and $E = \{-0.1, 0, 0.1\}$ with $\Delta t = 1.2$. Magenta objects represent the states of the vehicle.

The position uncertainty of the vehicle when its nominal position is $(x, y) \in \mathbb{R}^2$ is modeled as a disc centered at (x, y) with radius $\xi \in \mathbb{R}$, where ξ denotes the uncertainty:

$$D((x, y), \xi) = \{(x', y') \in \mathbb{R}^2 \mid \|(x, y), (x', y')\| \leq \xi\}, \quad (4)$$

where $\|\cdot\|$ denotes the Euclidian distance. Next, we explain how to obtain ξ .

Any state trajectory $q(t) \in G_K(q_{init})$, $t \in [0, K\Delta t]$, can be partitioned into K state trajectories: $q_k(t) = q(t')$, $t' \in [(k-1)\Delta t, k\Delta t]$, $k = 1, \dots, K$ (see Fig. 2). We denote the uncertainty at state q_k as ξ_k . Let $u_k + \varepsilon_k \in W_d$ be the applied control input at stage k such that $q_k(t) = q_k(q_{k-1}, u_k + \varepsilon_k, t)$, $k = 1, \dots, K$, with $q_0 = q_{init}$. Then, we set the uncertainty at state $q_k = [x_k, y_k, \theta_k]^T$ equal to:

$$\xi_k = \max_{[x', y', \theta']^T \in \{\bar{q}_k, q_k\}} \{ \|(x_k, y_k), (x', y')\| \} \text{ where} \\ \underline{q}_k(t) = \underline{q}_k(\underline{q}_{k-1}, u_k + \underline{\varepsilon}_k, t) \text{ and } \bar{q}_k(t) = \bar{q}_k(\bar{q}_{k-1}, u_k + \bar{\varepsilon}_k, t), \quad (5)$$

for $k = 1, \dots, K$, where $\underline{q}_0 = \bar{q}_0 = q_{init}$. Eqn. (5) is obtained using a worst case scenario assumption. If $u_k + \varepsilon_k \in W_d$ is the applied control input for QS, the corresponding applied control input at stage k for the original system is in $[u_k - \underline{\varepsilon}_k, u_k + \bar{\varepsilon}_k]$, where $\varepsilon_k \in [\underline{\varepsilon}_k, \bar{\varepsilon}_k] \in \mathcal{E}$. The position of the end state of the original system at stage k would be the farthest (in the Euclidean sense) from q_k , if the applied control input was either $u_i + \underline{\varepsilon}_i$, $i = 1, \dots, k$, or $u_i + \bar{\varepsilon}_i$, $i = 1, \dots, k$ (see [FMAG98] for more details). An example is given in Fig. 2.

From Eqn. (5) it follows that, given a state trajectory $q(t) \in G_K(q_{init})$, $t \in [0, K\Delta t]$, the uncertainty is increasing as a function of time. The way the uncertainty changes along $q(t)$ makes it difficult to characterize the exact shape of the position uncertainty region. Instead, we use a conservative approximation of the region. We define $\xi : [0, K\Delta t] \rightarrow \mathbb{R}$ as

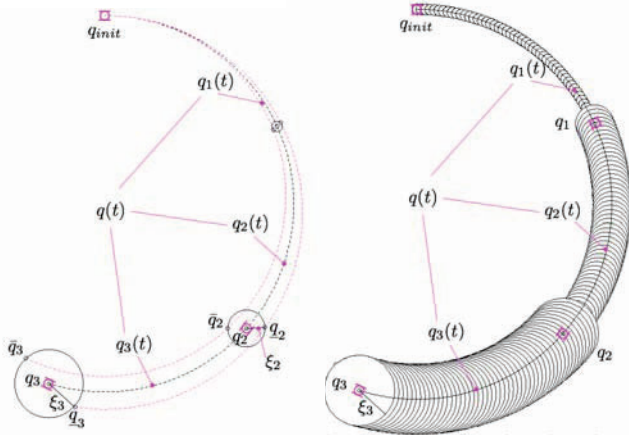


Fig. 2. Left: Evolution of the position uncertainty along the state trajectory $q(t)$, where $q(t)$ is partitioned into 3 state trajectories, $q_k(t)$, $k = 1, 2, 3$. Right: The conservative approximation of region $D((x(t), y(t)), \xi(t))$ along the state trajectory $q(t) = [x(t), y(t), \xi(t)]^T$, when the uncertainty trajectory is $\xi(t') = \xi_k(t)$, $t' \in [(k-1)\Delta t, k\Delta t]$, where $\xi_k(t) = \xi_k$, $k = 1, 2, 3$.

an approximated uncertainty trajectory and we set $\xi(t) = \xi_k$, $t \in [(k-1)\Delta t, k\Delta t]$, $k = 1, \dots, K$, i.e., we set the uncertainty along the state trajectory $q_k(t)$ equal to the maximum value of the uncertainty along $q_k(t)$, which is at state q_k . An example illustrating this idea is given in Fig. 2.

V. CONSTRUCTION OF AN MDP MODEL

Let $\Pi = \{\pi_p, \pi_d, \pi_u\}$ be a set of propositions, where π_p, π_d , and π_u label the pick-up, drop-off, and unsafe regions, respectively. We define $[\pi_p] = \{(x, y) \in \mathbb{R}^2 | (x, y) \in \cup_{r \in R_p} r\}$, $[\pi_d] = \{(x, y) \in \mathbb{R}^2 | (x, y) \in \cup_{r \in R_d} r\}$ and $[\pi_u] = \{(x, y) \in \mathbb{R}^2 | (x, y) \in \cup_{r \in R_u} r\}$ as the set of all positions that satisfy propositions π_p , π_d , and π_u , respectively.

A. MDP construction

A labeled MDP M that models the motion of the vehicle in the environment and the evolution of position uncertainty is defined as a tuple $(S, s_0, Act, A, P, \Pi, h)$ where:

- S is the finite set of states. For every state trajectory $q_k(t) \in G_K(q_{init})$, $t \in [(k-1)\Delta t, k\Delta t]$, $k = 1, \dots, K$, a state of the MDP is created. The meaning of the state is as follows: $(q(t), q, \bar{q}, \varepsilon, \bar{\varepsilon}, \Theta) \in S$ means that along the state trajectory $q(t)$, the uncertainty trajectory is

$$\xi(t) = \max_{[x', y', \theta']^T \in \{q, \bar{q}\}} \|(x, y), (x', y')\|,$$

where $[x, y, \theta]^T$ is the end state of $q(t)$; The noise interval is $[\underline{\varepsilon}, \bar{\varepsilon}] \in \mathcal{E}$; For $\Theta \in 2^\Pi$: (i) $\pi_p \in \Theta$, (ii) $\pi_d \in \Theta$, and (iii) $\pi_u \in \Theta$, mean that (i) it can be guaranteed that a pick-up region is entered, (ii) it can be guaranteed that the end state is inside of a drop-off region, and (iii) it is possible to enter R_u , along the state trajectory $q(t)$ when the uncertainty trajectory is $\xi(t)$ (see Fig. 3 for an example).

- $s_0 = (q_{init}, q_{init}, q_{init}, \emptyset, \emptyset, \Theta_{init}) \in S$ is the initial state, where $\Theta_{init} \in 2^\Pi$ is the set of propositions satisfied at q_{init} .
- $Act = U \cup \varphi$ is the set of actions, where φ is a dummy

action when the termination time is reached;

- $A : S \rightarrow 2^{Act}$ gives the enabled actions at state s : at termination time, $A(s) = \varphi$, otherwise $A(s) = U$;
- $P : S \times Act \times S \rightarrow [0, 1]$ is a transition probability function (its construction is described below);
- $\Pi = \{\pi_p, \pi_d, \pi_u\}$ is the set of propositions;
- $h : S \rightarrow 2^\Pi$ assigns propositions from Π to states $s \in S$, where h is as follows: if $s = (q(t), q, \bar{q}, \varepsilon, \bar{\varepsilon}, \Theta)$, then $\pi_p \in h(s)$ iff $\pi_p \in \Theta$, $\pi_d \in h(s)$ iff $\pi_d \in \Theta$, and $\pi_u \in h(s)$ iff $\pi_u \in \Theta$.

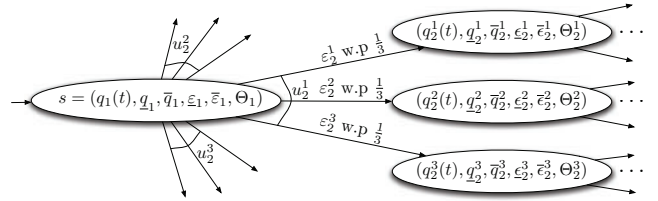
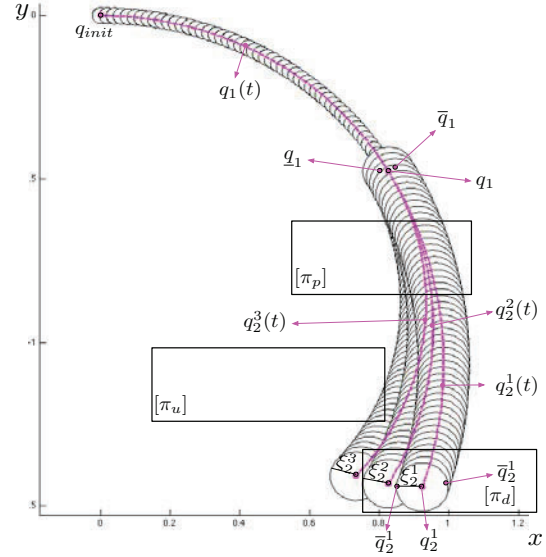


Fig. 3. Above: An example scenario corresponding to the MDP fragment shown below. For the state trajectory $q_2^i(t) = [x(t), y(t), \theta(t)]^T$, $t \in [\Delta t, 2\Delta t]$, when the uncertainty trajectory is $\xi_2^i(t) = \xi_2^i$ the following holds: (i) it can be guaranteed that a pick-up region is entered (i.e., $\exists t \in [\Delta t, 2\Delta t]$ s.t. $D((x(t), y(t)), \xi(t)) \subseteq [\pi_p]$), (ii) the end state is inside of a drop-off region (i.e., $D((x(2\Delta t), y(2\Delta t)), \xi(2\Delta t)) \subseteq [\pi_d]$), and (iii) R_u is not entered (i.e., $\forall t \in [\Delta t, 2\Delta t]$, $D((x(t), y(t)), \xi(t)) \cap [\pi_u] = \emptyset$). Thus, $\Theta_2^1 = \{\pi_p, \pi_d\}$. Similarly, $\Theta_2^2 = \{\pi_p, \pi_d\}$ but $\Theta_2^3 = \{\pi_p, \pi_u\}$. Below: A fragment of the MDP corresponding to the scenario shown above, where $[-\varepsilon_{max}, \varepsilon_{max}]$ is partitioned into $n = 3$ intervals. Action $u_2^1 \in A(s)$ enables three transitions, each w.p. $\frac{1}{3}$. This corresponds to applied control input being equal to $u_2^1 + \varepsilon_2^i$ w.p. $\frac{1}{3}$, $\varepsilon_2^i \in E$. The elements of the new states are: $q_2^i(t) = q_2^i(q_1, u_2^1 + \varepsilon_2^i, t)$; $\bar{q}_2^i(t) = \bar{q}_2^i(\bar{q}_1, u_2^1 + \varepsilon_2^i, t)$; $[\varepsilon_2^i, \bar{\varepsilon}_2^i] \in \mathcal{E}$ is s.t. $\varepsilon_2^i \in [\underline{\varepsilon}_2^i, \bar{\varepsilon}_2^i]$; and Θ_2^i is given above, where $i = 1, 2, 3$.

We generate S and P while building $G_K(q_{init})$ starting from q_{init} . Algorithm 1 takes as inputs a state $s \in S$ corresponding to some state trajectory $q_{k-1}(t)$ and an applied control input $u_k + \varepsilon_k$, and generates the new state of the MDP and updates S and P . First, given the end state of $q_{k-1}(t)$ and the applied control input $u_k + \varepsilon_k \in W_d$, the state trajectory at stage k , $q_k(t)$, is obtained (line 2). Then, using q_{k-1} and \bar{q}_{k-1} , and the fact that $\varepsilon_k \in [\underline{\varepsilon}_k, \bar{\varepsilon}_k] \in \mathcal{E}$, we obtain q_k and \bar{q}_k (line 3). The uncertainty trajectory along $q_k(t)$, $\xi_k(t)$ follows from Eqn. (5)

Algorithm 1: Generating S and P

Input: $s \in S, u_k \in U, \varepsilon_k \in E, S, P$ **Output:** S, P

- 1 $(q_{k-1}(t), \bar{q}_{k-1}, \bar{q}_{k-1}, \bar{\varepsilon}_{k-1}, \bar{\Theta}_{k-1}) = s;$
 - 2 $q_k(t) = [x_k(t), y_k(t), \theta_k(t)]^T = q_k(q_{k-1}, u_k + \varepsilon_k, t);$
 - 3 $\bar{q}_k(t) = \bar{q}_k(\bar{q}_{k-1}, u_k + \bar{\varepsilon}_k, t);$
 - 4 $\bar{\xi}_k(t) = \max_{[x', y', \theta']^T \in \{\bar{q}_k, \bar{q}_k\}} \|(x_k, y_k), (x', y')\|;$
 - 5 **if** $\exists t \in [(k-1)\Delta t, k\Delta t]$ s.t. $D((x_k(t), y_k(t), \bar{\xi}_k(t))) \subseteq [\pi_p]$ **then**
 - 6 $\bar{\Theta}_k = \bar{\Theta}_k \cup \{\pi_p\};$
 - 7 **if** $D((x_k, y_k), \bar{\xi}_k) \subseteq [\pi_d]$ **then**
 - 8 $\bar{\Theta}_k = \bar{\Theta}_k \cup \{\pi_d\};$
 - 9 **if** $\exists t \in [(k-1)\Delta t, k\Delta t]$ s.t. $D((x_k(t), y_k(t), \bar{\xi}_k(t))) \cap [\pi_u] \neq \emptyset$ **then**
 - 10 $\bar{\Theta}_k = \bar{\Theta}_k \cup \{\pi_u\};$
 - 11 $s' = (q_k(t), \bar{q}_k, \bar{q}_k, \bar{\varepsilon}_k, \bar{\Theta}_k); P(s, u_k, s') = \frac{1}{n}; S = S \cup \{s'\};$
-

(line 4). The algorithm checks if it can be guaranteed that a pick-up region is entered (lines 5–6), that the end state is inside of a drop-off region (lines 7–8), and if it is possible to enter R_u (lines 9–10) along $q_k(t)$ when the uncertainty trajectory is $\bar{\xi}_k(t)$. Finally, the newly generated state, s' , is added to S and the transition probability function is updated, i.e., $P(s, u_k, s') = \frac{1}{n}$ (line 11). This follows from the fact that given a control input $u_k \in U$ the applied control input will be $u_k + \varepsilon_k \in W_d$ with probability $\frac{1}{n}$, since $p_\omega(\omega(u_k) = u_k + \varepsilon_k) = \frac{1}{n}$, $\varepsilon_k \in E$ (see the MDP fragment in Fig. 3). In the technical report [CB12] we prove that M is a valid MDP.

VI. VEHICLE CONTROL STRATEGY

A. PCTL control policy generation

Formulas of PCTL are interpreted over states of an MDP and are constructed by connecting properties from Π using standard Boolean operators, the temporal operator \mathcal{U} (“until”), and the probabilistic operator \mathcal{P} . The requirement that the vehicle maximizes the probability of satisfying Specification 1 translates to the following PCTL formula:

$$\phi : \mathcal{P}_{\max=?}[\neg\pi_u \mathcal{U}(\neg\pi_u \wedge \pi_p \wedge \mathcal{P}_{>0}[\neg\pi_u \mathcal{U}(\neg\pi_u \wedge \pi_d)])]. \quad (6)$$

We use the PCTL control synthesis approach from [LWAB10] to generate a control policy for the MDP M . The tool takes as input an MDP and a PCTL formula ϕ and returns the control policy that maximizes the probability of satisfying ϕ , denoted μ , as well as the corresponding probability value, denoted V , where $V : S \rightarrow [0, 1]$. Specifically, for $s \in S$, $\mu(s) \in A(s)$ is the action to be applied at s and $V(s)$ is the probability of satisfying the specification at s under control policy μ . The tool is based on the off-the-shelf PCTL model-checking tool PRISM (see [KNP04]). We use Matlab to construct MDP M , which together with ϕ is passed to the PCTL control synthesis tool. The computational complexity of this step is as follows: Given U, E and K , the size of the MDP M is bounded above by $(|U| \times |E|)^K$. The time complexity of the control synthesis algorithm is polynomial in the size of the MDP and linear in the number of temporal operators in the formula.

B. Obtaining a vehicle control strategy

Let $[w_1, \bar{w}_1][w_2, \bar{w}_2] \dots [w_k, \bar{w}_k]$ be a sequence of measured intervals, where $[w_i, \bar{w}_i] = [u_i + \varepsilon_i, u_i + \bar{\varepsilon}_i]$, $u_i \in U$ and $[\varepsilon_i, \bar{\varepsilon}_i] \in \mathcal{E}$, $i = 1, \dots, k$. This corresponds to a unique path through the MDP: $s_0 \xrightarrow{u_1, [\varepsilon_1, \bar{\varepsilon}_1]} s_1 \xrightarrow{u_2, [\varepsilon_2, \bar{\varepsilon}_2]} s_2 \dots s_{k-1} \xrightarrow{u_k, [\varepsilon_k, \bar{\varepsilon}_k]} s_k$, where each transition is induced by a choice of action and the noise interval.

The desired vehicle control strategy is in the form of a finite sequence $\Gamma = \{\gamma_0, \gamma_1, \dots, \gamma_{K-1}\}$, where $\gamma_0 = \mu(s_0) \in U$ and $\gamma_k : (U \times \mathcal{E})^k \rightarrow U$, s.t. $\gamma_k((u_1, [\varepsilon_1, \bar{\varepsilon}_1]) \dots (u_k, [\varepsilon_k, \bar{\varepsilon}_k])) = \mu(s_k)$, for $k = 1, \dots, K-1$. At stage k , the control input is $u_k = \gamma_{k-1}((u_1, [\varepsilon_1, \bar{\varepsilon}_1]) \dots (u_{k-1}, [\varepsilon_{k-1}, \bar{\varepsilon}_{k-1}])) \in U$. Thus, given a sequence of measured intervals, Γ returns the control input for the next stage by mapping the sequence to the state of the MDP; the control input corresponds to the optimal action at that state.

Theorem 1: The probability that the system given by Eqn. (1), under the control strategy Γ , generates a trajectory that satisfies PCTL formula ϕ (Eqn. (6)) is bounded from below by $V(s_0)$, where $V(s_0)$ is the probability of satisfying ϕ on the MDP, under the control policy μ .

The result follows from the conservative approximation of the uncertainty region and the MDP construction (formal proof can be found in [CB12]). As a final remark, note that the bound obtained in this work approaches the true probability of satisfying ϕ in the theoretical limit, as $\Delta\varepsilon \rightarrow 0$.

VII. CASE STUDY

We considered the system given by Eqn. (1) and we used the following numerical values: $1/\rho = \pi/3$, $\Delta t = 1.2$, and $\varepsilon_{\max} = 0.06$ with $n = 3$, i.e., $\Delta\varepsilon = 0.04$. Thus, the maximum actuator noise was approximately 6% of the maximum control input.

Three case studies are shown in Fig. 4. The maximum probability of satisfying PCTL formula ϕ (Eqn. (6)) on the MDPs corresponding to cases A, B and C are 0.981, 0.874 and 0.892, respectively. For all three case studies, we found that $K = 6$ was enough as a terminal time, and we found the vehicle control strategies through the method described in Sec. VI. To verify our result, we simulated the original system under the obtained vehicle control strategies.

In Fig. 4, we show sample state trajectories and in Table 1 we compare the satisfaction probabilities obtained on the MDP with the simulation based satisfaction probabilities (number of satisfying trajectories over the number of generated trajectories). The results support Theorem 1, since the simulation based probabilities are bounded from below by the theoretical probabilities.

For each case study, the constructed MDP had approximately 20000 states. The Matlab code used to construct the MDP ran for approximately 4 minutes on a computer with a 2.5GHz dual processor. The control synthesis tool generated an optimal policy in about 1 minute.

VIII. CONCLUSION AND FUTURE WORK

We developed a feedback control strategy for a stochastic Dubins vehicle such that the probability of satisfying a

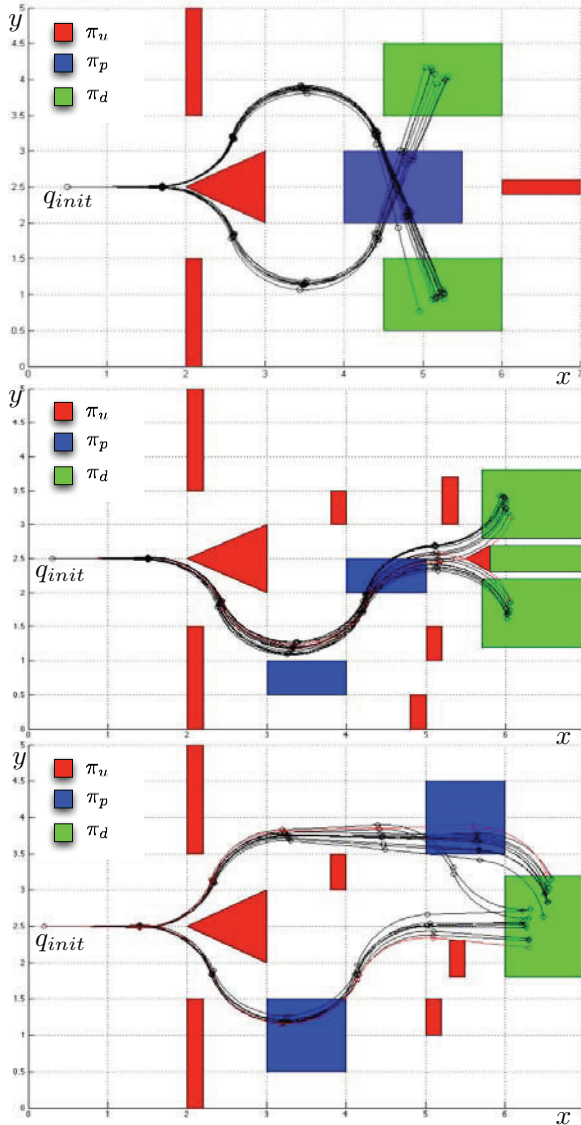


Fig. 4. 20 sample state (position) trajectories for cases A, B, and C (to be read top-to-bottom). The unsafe, pick-up, and the drop-off regions are shown in red, blue and green, respectively. Satisfying and violating trajectories are shown in black and red, respectively. In case A, all state trajectories were satisfying.

temporal logic statement over some environmental properties is maximized. Through discretization and quantization, we translated this problem to finding a control policy maximizing the probability of satisfying a PCTL formula on an MDP. We showed that the probability that the vehicle satisfies the specification in the original environment is bounded from below by the maximum probability of satisfying the specification on the MDP.

Future work includes extensions of this approach to controlling different types of vehicle models, allowing for richer temporal logic specifications, and experimental validations.

REFERENCES

[ADBS08] A. Abate, A. D’Innocenzo, Maria D. Di Benedetto, and S. Shankar Sastry. *Markov Set-Chains as Abstractions of*

TABLE I
THEORETICAL AND SIMULATION BASED PROBABILITIES OF SATISFYING
THE SPECIFICATION.

Case	Theoretical Probability ($V(s_0)$)	Number of generated trajectories		
		10^3	$5 \cdot 10^3$	10^4
A	0.981	1	1	1
B	0.874	0.902	0.926	0.928
C	0.892	0.913	0.931	0.935

Stochastic Hybrid Systems, volume 1. Springer Verlag, 2008.

[ASG07] R. Alterovitz, T. Simeon, and K. Goldberg. The stochastic motion roadmap: A sampling framework for planning with markov motion uncertainty. *Proceedings of Robotics: Science and Systems*, 2007.

[BK08] C. Baier and J. P. Katoen. *Principles of Model Checking*. MIT Press, 2008.

[CB12] I. Cizelj and C. Belta. Probabilistically safe control of noisy Dubins vehicles. Technical report, 2012. (available online at <http://arxiv.org/abs/1207.1280>).

[CGP99] E. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. The MIT Press, 1999.

[Dub57] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497–516, 1957.

[FMAG98] T. Fraichard, R. Mermond, and R. Alpes-Gravir. Path planning with uncertainty for car-like robots. In *IEEE Conf. on Robotics and Automation*, pages 27–32, 1998.

[Gir07] A. Girard. Approximately bisimilar finite abstractions of stable linear systems. In *Proceedings of the 10th international conference on Hybrid systems: computation and control, HSCC’07*, pages 231–244, Berlin, Heidelberg, 2007. Springer-Verlag.

[JP09] A. A. Julius and G. J. Pappas. Approximations of stochastic hybrid systems. *IEEE Transactions on Automatic Control*, 54(6):1193–1203, June 2009.

[KB08a] M. Kloetzer and C. Belta. Dealing with non-determinism in symbolic control. In M. Egerstedt and B. Mishra, editors, *Hybrid Systems: Computation and Control: 11th International Workshop, Lecture Notes in Computer Science*, pages 287–300. Springer Berlin / Heidelberg, 2008.

[KB08b] M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 2008.

[KF08] S. Karaman and E. Frazzoli. Vehicle routing problem with metric temporal logic specifications. In *IEEE Conf. on Decision and Control*, 2008.

[KGF07] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas. Where’s Waldo? Sensor-based temporal logic motion planning. In *IEEE International Conference on Robotics and Automation*, pages 3116–3121, 2007.

[KNP04] 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.

[LaV06] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006.

[LWAB10] M. Lahijanian, J. Wasniewski, S. B. Andersson, and C. Belta. Motion planning and control from temporal logic specifications with probabilistic satisfaction guarantees. In *Proc. 2010 IEEE International Conference on Robotics and Automation*, 2010.

[Maz04] M. Mazo. Robust area coverage using hybrid control. In *TELECOM’04, Santiago de Cuba, Cuba*, 2004.

[SWT09] A. Shkolnik, M. Walter, and R. Tedrake. Reachability-guided sampling for planning under differential constraints. In *IEEE International Conference on Robotics and Automation*, 2009.

[TP06] P. Tabuada and G. J. Pappas. Linear time logic control of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 2006.