2013 IEEE International Conference on Robotics and Automation (ICRA)
Karlsruhe, Germany, May 6-10, 2013

Control of Noisy Differential-Drive Vehicles from Time-Bounded Temporal Logic Specifications

Igor Cizelj and Calin Belta

***Abstract*— We address the problem of controlling a noisy differential drive mobile robot such that the probability of satisfying a specification given as a Bounded Linear Temporal Logic (BLTL) formula over a set of properties at the regions in the environment is maximized. We assume that the vehicle can precisely determine its 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 the angular velocity of its wheels using limited accuracy incremental encoders. Assuming the duration of the motion is finite, we map the measurements to a Markov Decision Process (MDP). We use recent results in Statistical Model Checking (SMC) to obtain 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 environment is bounded from below by the probability of satisfying the specification on the MDP. We illustrate our method with simulations and experimental results.**

I. Introduction

Temporal logics, such as Linear Temporal Logic (LTL) and Computational Tree Logic (CTL) have become increasingly popular for specifying robotic tasks (see, for example [KGFP07], [KB08b], [WTM09]). It has been shown that temporal logics 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".

In order to use existing model checking tools for motion planning (see [BK08]), many of the above-mentioned works rely on the assumption that the motion of the vehicle in the environment can be modeled as a finite 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]). If sensor and actuator noise models can be obtained from empirical measurements or an accurate simulator, then the robot motion can be modeled as a Markov Decision Process (MDP). 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 ([Gir07], [YTC$^+$12]). 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 consider a vehicle whose performance is measured by the completion of time constrained temporal logic tasks. In particular, we provide a conservative solution to the problem of controlling a stochastic differential drive mobile robot such that the probability of satisfying a specification given as a Bounded Linear Temporal Logic (BLTL) formula over a set of properties at the regions in the 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 a continuous probability distribution supported on a bounded interval, where the distribution is obtained through experimental trials. Also, we assume that the vehicle is equipped with two limited accuracy incremental encoders, each measuring the angular velocity of one of the wheels.

Assuming the duration of the motion is finite, through discretization, we map the incremental encoder measurements to an MDP. By relating the MDP to the vehicle motion in the environment, the vehicle control problem becomes equivalent to the problem of finding a control policy for an MDP such that the probability of satisfying the BLTL formula is maximized. Due to the size of the MDP, finding the exact solution is computationally too expensive. Therefore, we trade-off correctness for scalability and we use computationally efficient techniques based on system sampling. Specifically, we use recent results in Statistical Model Checking (SMC) for MDPs ([HMZ$^+$12]) to obtain an MDP control policy and a Bayesian Interval Estimation (BIE) algorithm ([ZPC10]) to estimate the probability of satisfying the specification. 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 under the obtained control policy.

The main contribution of this work lies in bridging the gap between a low level sensory inputs and a high level temporal logic specifications. We develop a framework for the synthesis of a vehicle feedback control strategy from such specifications based on a realistic model of an incremental encoder. This paper extends our previous work ([CB12b]) of controlling a stochastic version of Dubins vehicle such that the probability of satisfying a temporal logic statement, given as a Probabilistic CTL (PCTL) formula, over some

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

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

environmental properties, is maximized. Specifically, the approach presented here allows for richer temporal logic specifications, where the vehicle performance is measured by the completion of time constrained temporal logic tasks.

Due to page constraints, preliminaries are not included in this paper. We refer readers to [BK08] for information about MDPs and to [ZPC10] for detailed description of BLTL. Furthermore, we omit all proofs of all results. An extended version of this paper can be found in [CB12a].

II. Problem Formulation

A differential drive mobile robot ([LaV06]) is a vehicle having two main wheels, each of which is attached to its own motor, and a third wheel which passively rolls along preventing the robot from falling over. In this paper, we consider a stochastic version of a differential drive mobile robot, which captures actuator noise:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}(u_r+\varepsilon_r+u_l+\varepsilon_l)\cos(\theta) \\ \frac{r}{2}(u_r+\varepsilon_r+u_l+\varepsilon_l)\sin(\theta) \\ \frac{r}{L}(u_r+\varepsilon_r-u_l-\varepsilon_l) \end{bmatrix},\ u_r \in U_r,\ u_l \in U_l, \tag{1}$$

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_r and u_l are the control inputs (angular velocities before being corrupted by noise), U_r and U_l are control constraint sets, and ε_r and ε_l are random variables modeling the actuator noise with continuous probability density functions supported on the bounded intervals $[\varepsilon_r^{min}, \varepsilon_r^{max}]$ and $[\varepsilon_l^{min}, \varepsilon_l^{max}]$, respectively. L is the distance between the two wheels and r is the wheel radius. We denote the state of the system by $q = [x,y,\theta]^T \in SE(2)$.

Motivated by the fact that the time optimal trajectories for the bounded velocity differential drive robots are composed only of turns in place and straight lines ([BM00]), we assume U_r and U_l are finite, but we make no assumptions on optimality. We define

$$W_i = \{u+\varepsilon | u \in U_i, \varepsilon \in [\varepsilon_i^{min}, \varepsilon_i^{max}]\},\ i \in \{r,l\},$$

as the sets of applied control inputs, i.e, the sets of angular wheel 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 $K\Delta t$ (later in this section we explain how K is determined). We denote the control inputs and the applied control inputs at stage k as $u_i^k \in U_i,\ i \in \{r,l\}$, and $w_i^k \in W_i,\ i \in \{r,l\}$, respectively.

We assume that the vehicle is equipped with two incremental encoders, each measuring the applied control input (i.e., the angular velocity corrupted by noise) of one of the wheels. Motivated by the fact that the angular velocity is considered constant inside the given observation stage ([PTPZ07]), the applied controls are considered piecewise constant, i.e., $w_i : [(k-1)\Delta t, k\Delta t] \to W_i,\ i \in \{r,l\}$, are constant over each stage.

Incremental encoder model: As shown in [PTPZ07], the measurement resolution of an incremental encoder is constant and for encoder i we denote it as $\Delta\varepsilon_i,\ i \in \{r,l\}$. Given $\Delta\varepsilon_i$ and $[\varepsilon_i^{min}, \varepsilon_i^{max}],\ i \in \{r,l\}$, then the following holds: $\exists n_i \in \mathbb{Z}^+$ s.t. $n_i\Delta\varepsilon_i = |\varepsilon_i^{max} - \varepsilon_i^{min}|,\ i \in \{r,l\}$ (for more details see Sec. VII.) Then, $[\varepsilon_i^{min}, \varepsilon_i^{max}]$ can be partitioned[1] into n_i noise intervals of length $\Delta\varepsilon_i$: $[\underline{\varepsilon}_i^{j_i}, \overline{\varepsilon}_i^{j_i}],\ j_i = 1,\ldots,n_i,\ i \in \{r,l\}$. We denote the set of all noise intervals $\mathscr{E}_i = \{[\underline{\varepsilon}_i^1, \overline{\varepsilon}_i^1], \ldots, [\underline{\varepsilon}_i^{n_i}, \overline{\varepsilon}_i^{n_i}]\},\ i \in \{r,l\}$. At stage k, if the applied control input is $u_i^k + \varepsilon_i$, the incremental encoder i will return measured interval

$$[\underline{w}_i^k, \overline{w}_i^k] = [u_i^k + \underline{\varepsilon}_i, u_i^k + \overline{\varepsilon}_i],$$

where $\varepsilon_i \in [\underline{\varepsilon}_i, \overline{\varepsilon}_i] \in \mathscr{E}_i,\ i \in \{r,l\}$. The pair of measured intervals at stage k, $([\underline{w}_r^k, \overline{w}_r^k], [\underline{w}_l^k, \overline{w}_l^k])$, returned by the incremental encoders, is denoted $\mathbb{W}^k$.

The vehicle moves in a planar environment in which a set of non-overlapping regions of interest, denoted R, is present. Let Π be the set of propositions satisfied at the regions in the environment. One of these propositions, denoted by $\pi_u \in \Pi$, signifies that the corresponding regions are `unsafe`. We employ BLTL to describe high level motion specification.

Formulas of BLTL are constructed by connecting properties from a set of proposition Π using Boolean operators ($\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction)), and temporal operators ($\mathbf{U}^{\leq t}$ (bounded until), $\mathbf{F}^{\leq t}$ (bounded finally), and $\mathbf{G}^{\leq t}$ (bounded globally), where $t \in \mathbb{R}^{\geq 0}$ is the time bound parameter). The semantics of BLTL formulas are given over infinite traces $\sigma = (o_1,t_1)(o_2,t_2)\ldots,\ o_i \in 2^{\Pi},\ t_i \in \mathbb{R}^{\geq 0},\ i \geq 1$, where o_i is the set of satisfied propositions and t_i is the time spent satisfying o_i. A trace satisfies a BLTL formula ϕ if ϕ is true at the first position of the trace; $\mathbf{G}^{\leq t}\phi_1$ means that ϕ_1 will remain true for the next t time units; and $\phi_1 \mathbf{U}^{\leq t} \phi_2$ means that ϕ_2 will be true within the next t time units and ϕ_1 remains true until then. The fact that trace σ satisfies ϕ is denoted $\sigma \vDash \phi$.

In this work, the motion specification is expressed as a BLTL formula ϕ over Π:

$$\phi = \neg\pi_u \mathbf{U}^{\leq T_1}(\varphi_1 \wedge \neg\pi_u \mathbf{U}^{\leq T_2}(\varphi_2 \wedge \ldots \wedge \neg\pi_u \mathbf{U}^{\leq T_f}\varphi_f)), \tag{2}$$

$f \in \mathbb{Z}^+$, and $\varphi_j,\ \forall j \in \{1,\ldots,f\}$, is of the following form:

$$\varphi_j = \mathbf{G}^{\leq \tau_j^1}(\bigvee\nolimits_{\pi \in \Pi_j^1} \pi) \vee \ldots \vee \mathbf{G}^{\leq \tau_j^{n_j}}(\bigvee\nolimits_{\pi \in \Pi_j^{n_j}} \pi),$$

where $n_j \in \mathbb{Z}^+$, $\forall_{n=1,\ldots,n_j} \Pi_j^n \subset \Pi \setminus \pi_u$, $\forall_{n=1,\ldots,n_j} \tau_j^n \in \mathbb{R}^{\geq 0}$ and $T_j \in \mathbb{R}^{\geq 0}$.

Example 1: Consider the environment shown in Fig. 2 (Sec. IV). Let $\Pi = \{\pi_u, \pi_p, \pi_{t1}, \pi_{t2}, \pi_d\}$ where $\pi_u, \pi_p, \pi_{t1}, \pi_{t2}, \pi_d$ label the `unsafe`, `pick-up`, `test1`, `test2` and the `drop-off` regions, respectively. Let the motion specification be as follows:

Start from an initial state q_{init} and reach a `pick-up` region within 14 time units and stay in it at least 0.8 time units, to pick-up a load. After entering the `pick-up` region,

[1] Throughout the paper, we relax the notion of partition by allowing the endpoints of the intervals to overlap.

reach a `test1` *region within* 5 *time units and stay in it at least* 1 *time unit or reach a* `test2` *region within* 5 *time units and stay in it at least* 0.8 *time units. Finally, after entering the* `test1` *region or the* `test2` *region reach a* `drop-off` *region within* 4 *time units to drop off the load. Always avoid the* `unsafe` *regions.*

The specification translates to BLTL formula ϕ:

$$\phi = \neg\pi_u \mathbf{U}^{\leq 14}(\mathbf{G}^{\leq 0.8}\pi_p \wedge \neg\pi_u \mathbf{U}^{\leq 5}([\mathbf{G}^{\leq 1}\pi_{t1} \vee \mathbf{G}^{\leq 0.8}\pi_{t2}] \wedge \neg\pi_u \mathbf{U}^{\leq 4}\pi_d)). \quad (3)$$

We assume that the vehicle can precisely determine its initial state $q_{init} = [x_{init}, y_{init}, \theta_{init}]$, in a known map of the environment. While the vehicle moves, incremental encoder measurements $\mathbb{W}^k$ are available at each stage k. We define a *vehicle control strategy* as a map that takes as input a sequence of pairs of measured intervals $\mathbb{W}^1\mathbb{W}^2\dots\mathbb{W}^{k-1}$, and returns control inputs $u_r^k \in U_r$ and $u_l^k \in U_l$ at stage k. Let us formulate the main problem we consider in this paper:

Problem 1: Given a set of regions of interest R satisfying propositions from a set Π, a vehicle model described by Eqn. (1) with initial state q_{init}, a motion specification expressed as a BLTL formula ϕ over Π (Eqn. (2)), find a vehicle control strategy that maximizes the probability of satisfying the specification.

To fully specify Problem 1, we need to define the satisfaction of a BLTL formula ϕ by a trajectory $q : [0, K\Delta t] \rightarrow SE(2)$ of the system from Eqn. (1). A formal definition is given in [CB12a]. Informally, since the duration of the motion is finite, $q(t)$ produces a finite trace $\sigma = (o_1,t_1)(o_2,t_2)\dots(o_l,t_l)$, $o_i \in \Pi \cup \emptyset$, $t_i \in \mathbb{R}^{\geq 0}$, $i \geq 1$, where o_i is the satisfied proposition[2] and t_i is the time spent satisfying o_i, as time evolves. A trajectory $q(t)$ satisfies BLTL formula ϕ if and only if the generated trace satisfies the formula. We give an example in Fig. 2 (Sec. IV). Given ϕ, for the duration of the motion we use the smallest $K \in \mathbb{Z}^+$ for which model checking a trace is well defined, i.e., the smallest K for which the maximum nested sum of time bounds (see [ZPC10]) is at most $K\Delta t$.

III. Construction of an MDP Model

Recall that ε_i is a random variable with a continuous probability density function supported on the bounded interval $[\varepsilon_i^{min}, \varepsilon_i^{max}]$, $i \in \{r,l\}$. The functions are obtained through experimental trials and they are defined as follows:

$$\Pr(\varepsilon_i \in [\underline{\varepsilon}_i^{j_i}, \overline{\varepsilon}_i^{j_i}]) = p_i^{j_i}, \quad (4)$$

$[\underline{\varepsilon}_i^{j_i}, \overline{\varepsilon}_i^{j_i}] \in \mathscr{E}_i$, $j_i = 1,\dots,n_i$, s.t. $\sum_{j_i=1}^{n_i} p_i^{j_i} = 1$, $i \in \{r,l\}$.

An MDP M that captures every sequence realization of pairs of measurements returned by the incremental encoders is defined as a tuple (S, s_0, Act, A, P), where:

- $S = \cup_{k=1,\dots,K}\{([u_r + \underline{\varepsilon}_r, u_r + \overline{\varepsilon}_r], [u_l + \underline{\varepsilon}_l, u_l + \overline{\varepsilon}_l]) | u_r \in U_r, u_l \in U_l, [\underline{\varepsilon}_r, \overline{\varepsilon}_r] \in \mathscr{E}_r, [\underline{\varepsilon}_l, \overline{\varepsilon}_l] \in \mathscr{E}_l\}^k$ is the set of states with the following meaning: $(\mathbb{W}^1,\dots,\mathbb{W}^k) \in S$ means that at stage i, $1 \leq i \leq k$, the pair of measured intervals is $\mathbb{W}^i$.
- $s_0 = \emptyset$ is the initial state.
- $Act = \{U_r \times U_l\} \cup \varphi$ 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: if $|s| = K$, i.e., if the termination time is reached, $A(s) = \varphi$, otherwise $A(s) = \{U_r \times U_l\}$.
- $P : S \times Act \times S \rightarrow [0,1]$ is a transition probability function constructed by the following rules:

1) If $s = (\mathbb{W}^1,\dots,\mathbb{W}^k) \in S$ then $P(s,a,s') = p_r^m p_l^n$ iff $s' = (\mathbb{W}^1,\dots,\mathbb{W}^k, ([u_r + \underline{\varepsilon}_r^m, u_r + \overline{\varepsilon}_r^m], [u_l + \underline{\varepsilon}_l^n, u_l + \overline{\varepsilon}_l^n])) \in S$ and $a = (u_r, u_l) \in \{U_r \times U_l\}$ where $m = 1,\dots,n_r$, $n = 1,\dots,n_l$ and $k = 1,\dots,K$.;
2) If $|s| = K$ then $P(s,a,s') = 1$ iff $a = \varphi$ and $s' = s$;
3) $P(s,a,s') = 0$ otherwise.

In the technical report [CB12a] we prove that P is a valid transition probability function. Finally, we define a *control policy* μ of an MDP M as a function that resolves nondeterminism in each state s by providing a distribution over the set of actions enabled in s, i.e., $\mu(s,a) : S \times Act \rightarrow [0,1]$, s.t., $\sum_{a \in A(s)} \mu(s,a) = 1$ and $\mu(s,a) > 0$ only if a is enabled in s. A control policy for which either $\mu(s,a) = 1$ or $\mu(s,a) = 0$ for all pairs $(s,a) \in S \times Act$ is called deterministic.

IV. Position uncertainty

A. Nominal state trajectory

For each interval belonging to the set of noise intervals $\mathscr{E}_i$, we define a representative value $\varepsilon_i^{j_i} = (\underline{\varepsilon}_i^{j_i} + \overline{\varepsilon}_i^{j_i})/2$, $j_i = 1,\dots,n_i$, $i \in \{r,l\}$, i.e., $\varepsilon_i^{j_i}$ is the midpoint of interval $[\underline{\varepsilon}_i^{j_i}, \overline{\varepsilon}_i^{j_i}] \in \mathscr{E}_i$, $i \in \{r,l\}$. We denote the set of representative values as $E_i = \{\varepsilon_i^1,\dots,\varepsilon_i^{n_i}\}$, $i \in \{r,l\}$.

We use $q^k(t)$, w_r^k and w_l^k, $t \in [(k-1)\Delta t, k\Delta t]$, $k = 1,\dots,K$, to denote the state trajectory and the constant applied controls 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 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 the applied controls are constant and equal to w_r^k and w_l^k. Throughout the paper, we will also denote this trajectory by $q^k(q^{k-1}, w_r^k, w_l^k, t)$, when we want to explicitly capture the initial state q^{k-1} and the constant applied controls w_r^k and w_l^k.

Given a path through the MDP:

$$s_0 \xrightarrow{(u_r^1,u_l^1)} s_1 \xrightarrow{(u_r^2,u_l^2)} s_2 \dots s_{K-1} \xrightarrow{(u_r^K,u_l^K)} s_K, \quad (5)$$

where $s_k = (\mathbb{W}^1,\dots,\mathbb{W}^k)$, with $\mathbb{W}^k = ([u_r^k + \underline{\varepsilon}_r^k, u_r^k + \overline{\varepsilon}_r^k], [u_l^k + \underline{\varepsilon}_l^k, u_l^k + \overline{\varepsilon}_l^k])$, $k = 1,\dots,K$, we define the *nominal state trajectory* $q(t)$, $t \in [0, K\Delta t]$, as follows: $q(t) = q^k(q^{k-1}, u_r^k + \varepsilon_r^k, u_l^k + \varepsilon_l^k, t)$, $t \in [(k-1)\Delta t, \Delta t]$, $k = 1,\dots,K$, where $\varepsilon_i^k \in E_i$ is such that $\varepsilon_i^k \in [\underline{\varepsilon}_i^k, \overline{\varepsilon}_i^k]$, $i \in \{r,l\}$ and $q^0 = q_{init}$. For every path through the MDP, its nominal state trajectory is well defined. The next step is to define the uncertainty evolution, along the nominal state trajectory, since the applied controls can take any value within the measured intervals.

B. Position uncertainty evolution

Since a motion specification is a statement about the propositions satisfied by the regions of interest in the environment, in order to answer whether a state trajectory satisfies ϕ (Eq. (2)) it is sufficient to know its projection in $\mathbb{R}^2$.

[2]Since the regions of interest are non-overlapping it follows that $o_i \in \Pi \cup \emptyset$.

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 $d \in \mathbb{R}$, where d denotes the distance uncertainty:

$$D((x,y),d) = \{(x'y') \in \mathbb{R}^2 | ||(x,y),(x',y')|| \leq d\}, \quad (6)$$

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

First, let $\Delta\theta \in S^1$ denote the orientation uncertainty. Let $q(t)$, $t \in [0, K\Delta t]$, be the nominal state trajectory corresponding to a path through the MDP (Eqn. (5)). Then, $q(t)$ can be partitioned into K state trajectories: $q^k(t) = q^k(q^{k-1}, u_r^k + \varepsilon_r^k, u_l^k + \varepsilon_l^k, t)$, $t \in [(k-1)\Delta t, \Delta t]$, $k = 1,\ldots,K$, where $\varepsilon_i^k \in E_i$ is such that $\varepsilon_i^k \in [\underline{\varepsilon}_i^k, \overline{\varepsilon}_i^k] \in \mathscr{E}_i$, $i \in \{r,l\}$ and $q^0 = q_{init}$ (see Fig. 1). The distance and orientation uncertainty at state q^k are denoted as d^k and $\Delta\theta^k$, respectively. We set d^k and $\Delta\theta^k$ at state $q^k = [x^k, y^k, \theta^k]^T$ equal to:

$$\begin{aligned} &d^k = \max_{[x',y',\theta']^T \in \mathscr{R}^k}(||(x^k,y^k),(x',y')||) + d^{k-1} \text{ and} \\ &\Delta\theta^k = \max_{[x',y',\theta']^T \in \mathscr{R}^k}(|\theta^k - \theta'|), \end{aligned} \quad (7)$$

where

$$\begin{aligned} \mathscr{R}^k = \{ & q^k([x^{k-1}, y^{k-1}, \theta^{k-1} + \alpha]^T, u_r^k + \varepsilon_r', u_l^k + \varepsilon_l', k\Delta t) | \\ & \alpha \in \{\Delta\theta^{k-1}, -\Delta\theta^{k-1}\}, \varepsilon_r' \in \{\underline{\varepsilon}_r^k, \overline{\varepsilon}_r^k\}, \varepsilon_l' \in \{\underline{\varepsilon}_l^k, \overline{\varepsilon}_l^k\}\}, \end{aligned} \quad (8)$$

for $k = 1,\ldots,K$, where $d^0 = 0$ and $\Delta\theta^0 = 0$.

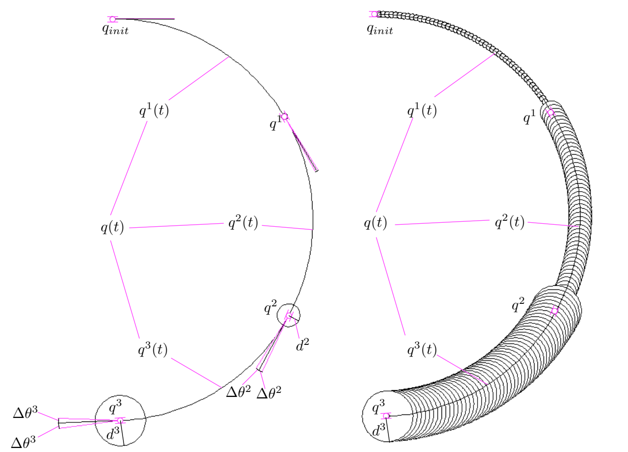

Fig. 1. Left: Evolution of the position uncertainty along the nominal state trajectory $q(t) = [x(t), y(t), \theta(t)]$, where $q(t)$ is partitioned into 3 state trajectories, $q^k(t)$, $k = 1,2,3$. Right: The conservative approximation of region $D((x(t),y(t)),d(t))$ along $q(t)$, where the distance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

Eqn. (7) and (8) are obtained using a worst scenario assumption. At stage k, the pair of measured intervals is $\mathbb{W}^k = ([u_r^k + \underline{\varepsilon}_r^k, u_r^k + \overline{\varepsilon}_r^k], [u_l^k + \underline{\varepsilon}_l^k, u_l^k + \overline{\varepsilon}_l^k])$ and we use the endpoints of the measured intervals to define set $\mathscr{R}^k$. $\mathscr{R}^k$ is the smallest set of points in $SE(2)$, at the end of stage k, guaranteed to contain (i) the state with the maximum distance (in Euclidian sense) from q^k given that the applied controls at stage i are within the measured intervals at stage i, and (ii) the state with the maximum orientation difference compared to q^k given that the applied controls at stage i are within the measured intervals at stage i, $i = 1,\ldots,k$. (for more details about $\mathscr{R}^k$ see [FMAG98]). An example is given in Fig. 1.

From Eqn. (7) and (8) it follows that, given a nominal state trajectory $q(t)$, $t \in [0, K\Delta t]$, the distance uncertainty increases as a function of time. The way it 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 $d : [0, K\Delta t] \to \mathbb{R}$ as an *approximate distance uncertainty trajectory* and we set $d(t) = d^k$, $t \in [(k-1)\Delta t, k\Delta t]$, $k = 1,\ldots,K$, i.e., we set the distance uncertainty along the state trajectory $q^k(t)$ equal to the maximum value of the distance uncertainty along $q^k(t)$, which is at state q^k (see Fig. 1).

C. Generating a trace under the position uncertainty

Let $q(t)$ be a nominal state trajectory with the distance uncertainty trajectory $d(t)$, $t \in [0, K\Delta t]$. In [CB12a] we introduce a set of conservative rules according to which the trace corresponding to the uncertainty region $D((x(t),y(t)),d(t))$ is generated. In Fig. 3 we show an uncertainty region and the corresponding trace. The rules guarantee that if the generated trace satisfies ϕ (Eqn. (2)) then any state (position) trajectory, inside $D((x(t),y(t)),d(t))$, will satisfy ϕ. Formal proof can be found in [CB12a].

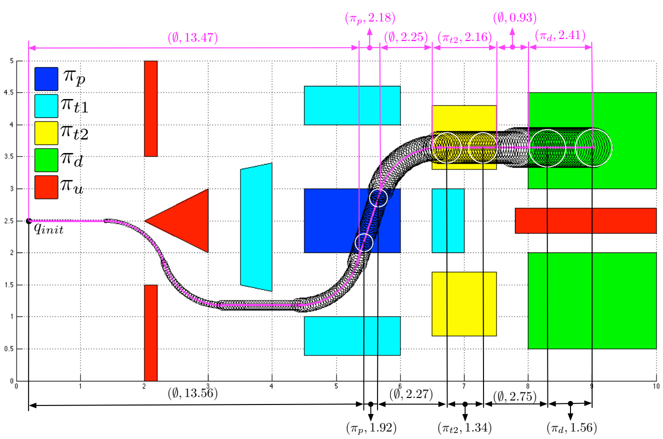

Fig. 2. An example environment with the regions of interest. The `unsafe`, `pick-up`, `test1`, `test2` and the `drop-off` regions are shown in red, blue, cyan, yellow and green, respectively. An uncertainty region and a sample state (position) trajectory, inside the uncertainty region, are shown in black and magenta, respectively. The corresponding generated traces are $\sigma^D = (\emptyset, 13.56)(\pi_p, 1.92)(\emptyset, 2.27)(\pi_{t2}, 1.34)(\emptyset, 2.75)(\pi_d, 1.56)$ and $\sigma^{q'} = (\emptyset, 13.47)(\pi_p, 2.18)(\emptyset, 2.25)(\pi_{t2}, 2.16)(\emptyset, 0.93)(\pi_d, 2.41)$. Let ϕ be as given by Eq. (3). Then, it follows that $\sigma^D \vDash \phi$ and $\sigma^{q'} \vDash \phi$.

V. Vehicle Control Strategy

Given the MDP M, the next step is to obtain a control policy that maximizes the probability of generating a path through M such that the corresponding trace is satisfying. Given U_r, U_l, n_r, n_l and K, the size of the MDP M is bounded above by $(|U_r| \times |U_l| \times n_r \times n_l)^K$. Even for a simple case study, due to the size of M, using exact methods to obtain a control policy is computationally too expensive. Therefore, we decide to trade-off correctness for scalability and use computationally efficient techniques based on sampling.

We obtain a suboptimal control policy by iterating over the *control synthesis* and the *probability estimation* procedure until the stopping criterion is met (see Sec. V-B). In the control synthesis procedure we use a *control policy optimization* algorithm from [HMZ+12] to incrementally improve a candidate control policy for the MDP M (control policy is initialized with a uniform distribution at each state). Next, in the probability estimation procedure we use SMC by BIE, as presented in [ZPC10]. We estimate the probability that the MDP M, under the candidate control policy, generates a path such that the corresponding trace satisfies BLTL formula ϕ. Finally, if the estimated probability converges, i.e., if the stopping criterion is met, we map the control policy to a vehicle control strategy. Otherwise, the control synthesis procedure is restarted using the latest update of the control policy.

A. *Control synthesis*

The details of the control policy optimization algorithm can be found in [HMZ+12] and here we only give an informal overview of the approach. In the control policy evaluation procedure we sample paths of the MDP M under the current control policy μ. Given a path $\omega = s_0 \xrightarrow{a^1} s_1 \xrightarrow{a^2} s_2 \ldots s_{K-1} \xrightarrow{a^K} s_K$, where $a^k = (u_r^k, u_l^k)$, the corresponding trace σ is generated as described in Sec. IV. Next, we check BLTL formula ϕ on each σ and estimate how likely it is for each action to lead to the satisfaction of ϕ, i.e., we obtain the estimate of the probability that a path crossing a state-action pair, (s^k, a^{k+1}), $k = 0, \ldots, K-1$, in ω will generate a trace that satisfies ϕ. These estimates are then used in the control policy improvement procedure, in which we update the control policy μ by reinforcing the actions that led to the satisfaction of ϕ most often. The authors ([HMZ+12]) show that the updated control policy is provably better than the previous one by focusing on more promising regions of the state space. In the next step, to estimate the probability of satisfaction, we use the deterministic version of the updated probabilistic control policy μ, denoted μ_{det} where: for all $s \in S$ and $a \in A$, $\mu_{det}(s,a) = I\{a = \arg\max_{a \in Act(s)} \mu(s,a)\}$.

B. *Probability estimation*

Next, we determine the estimate of the probability that the MDP M, under μ_{det}, generates a path such that the corresponding trace satisfies ϕ. To do so we use the BIE algorithm as presented in [ZPC10]. We denote the exact probability as p_M and the estimate as $\hat{p}_M$.

The algorithm generates traces by sampling paths through M under μ_{det} (as described in Sec. IV) and checks whether the corresponding traces satisfy ϕ, until enough statistical evidence has been found to support the claim that p_M is inside the interval $[\hat{p}_M - \delta, \hat{p}_M + \delta]$ with arbitrarily high probability, i.e., $\Pr(p_M \in [\hat{p}_M - \delta, \hat{p}_M + \delta]) \geq c$, where $c \in (\frac{1}{2}, 1)$ and $\delta \in (0, \frac{1}{2})$ are user defined parameters.

We stop iterating over the control synthesis and the probability estimation procedure when the difference between the two consecutive probability estimates converges to a neighborhood of radius $e \in (0,1)$. Let μ^*_{det} and $\hat{p}^*_M$ be the current control policy and the corresponding probability estimate, respectively, when the stopping criterion is met.

C. *Control strategy*

The vehicle control strategy is a function $\gamma : S \to \{U_r \times U_l\}$ that maps a sequence of pairs of measured intervals, i.e., a state of the MDP, to the control inputs: $\gamma((\mathbb{W}^1, \ldots, \mathbb{W}^k)) = \gamma(s_k) = \arg\max_{a \in Act(s_k)} \mu^*_{det}(s_k, a)$, $k = 1, \ldots, K-1$ with $\gamma(s_0) = \arg\max_{a \in Act(s_0)} \mu^*_{det}(s_0, a)$. At stage k, the control inputs are $(u_r^k, u_l^k) = \gamma((\mathbb{W}^1, \ldots, \mathbb{W}^{k-1})) \in \{U_r \times U_l\}$. Thus, given a sequence of pairs of measured intervals, γ returns the control inputs for the next stage; the control inputs are equal to the action returned by μ^*_{det} at the state of the MDP corresponding to that sequence.

Theorem 1: The probability that the system given by Eqn. (1), under the vehicle control strategy γ, generates a state trajectory that satisfies BLTL formula ϕ (Eqn. (2)) is bounded from below by p^*_M, where $\Pr(p^*_M \in [\hat{p}^*_M - \delta, \hat{p}^*_M + \delta]) \geq c$. The result follows from the conservative approximation of the uncertainty region, the result mentioned in IV-C and the MDP construction (formal proof can be found in [CB12a]).

VI. CASE STUDY

We considered the system given by Eqn. (1) and we used the numerical values corresponding to Dr. Robot's x80Pro mobile robot equipped with two incremental encoders. The parameters were $r = 0.085$m and $L = 0.295$m. To reduce the complexity, $\{U_r \times U_l\}$ was limited to $\{(\frac{1+L}{4r}, \frac{1-L}{4r}), (\frac{1}{4r}, \frac{1}{4r}), (\frac{1-L}{4r}, \frac{1+L}{4r})\}$, where the pairs of control inputs corresponded to a vehicle turning left at $\frac{1}{2}\frac{\text{rad}}{\text{s}}$, going straight, and turning right at $\frac{1}{2}\frac{\text{rad}}{\text{s}}$, respectively, when the forward speed is $\frac{1}{4}\frac{\text{m}}{\text{s}}$.

Measurement resolution and probability density functions: To obtain the angular wheel velocity, the frequency counting method [PTPZ07] was used, i.e., the encoder pulses inside a given sampling period were counted. The number of pulses per revolution (i.e., the number of windows in the code track of the encoders) was 378 and the sampling period was set to $\Delta t = 2.6$s. Thus, according to [PTPZ07] the measurement resolution was $\Delta\varepsilon_r = \Delta\varepsilon_l = \frac{2\pi}{378 \cdot 2.6} \approx 0.0064$. We obtained $-\varepsilon_r^{min} = \varepsilon_r^{max} = -\varepsilon_l^{min} = \varepsilon_l^{max} = 0.0096$ and the corresponding probabilities through experimental trials (details can be found in [CB12a]).

We considered the motion specification and the BLTL formula given in Example 1 (Sec. II). Two different environments are shown in Fig. 3. The estimated probability $\hat{p}^*_M$ corresponding to environment A and B was 0.664 and 0.719, respectively. From Eq. (3) it followed that $K = 9$. For both environments, we found the vehicle control strategy as described in Sec. V. To verify Theorem 1, we performed multiple runs of BIE algorithm by simulating the system under the vehicle control strategy. We denote the resulting probability estimate as $\hat{p}_S$ and we compare it to $\hat{p}^*_M$.

In Fig. 3 we show sample state trajectories and in Table I we compare the estimated probabilities obtained on the MDP, $\hat{p}^*_M$, with the estimated probabilities obtained by simulating the system, $\hat{p}_S$. The results support Theorem 1, since $\hat{p}_S$

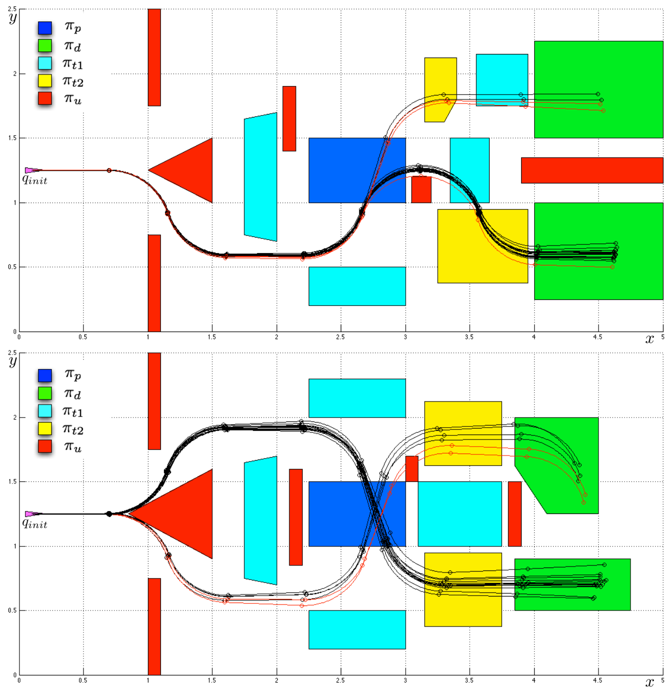

Fig. 3. 20 sample state (position) trajectories for cases A and B (to be read top-to-bottom). The `unsafe`, `pick-up`, `test1`, `test2`, and the `drop-off` regions are shown in red, blue, cyan, yellow and green, respectively. Satisfying and violating trajectories are shown in black and red, respectively. Note that, in case A, the upper two red trajectories avoid the `unsafe` regions and visit the `pick-up`, `test2`, and the `drop-off` region in the correct order, but they violate the specification because they do not stay long enough in the `test2` region.

is bounded from below by $\hat{p}_M^*$. The discrepancy in the probabilities is mostly due to the conservative approximation of the uncertainty region in Sec. IV. The Matlab code used to obtain the vehicle control strategy ran for approximately 2.2 hours on a computer with a 2.5GHz dual processor. A movie showing a robot motion produced by applying the vehicle control strategy for environment A is available online at http://people.bu.edu/icizelj/Igor_Cizelj/diff-bltl.html .

TABLE I

PROBABILITY ESTIMATES OF SATISFYING THE SPECIFICATION

Environment	$\hat{p}_M^*$	$\hat{p}_S$		
		Run 1	Run 2	Run 3
A	0.664	0.847	0.832	0.826
B	0.719	0.891	0.898	0.879

VII. CONCLUSION

We developed a feedback control strategy for a stochastic differential drive mobile robot such that the probability of satisfying a time constrained specification given in terms of a temporal logic statement is maximized. By mapping sensor measurements to an MDP we translate the problem to finding a control policy maximizing the probability of satisfying a BLTL formula on the MDP. The solution is based on SMC for MDPs and we show that the probability that the vehicle satisfies the specification is bounded from below by the probability of satisfying the specification on the MDP.

The key limitation of the proposed approach is the computation time. Since sampling accounts for the majority of our runtime, future work includes improving the sampling performance and making the implementation fully parallel, as well as developing a less conservative uncertainty model.

REFERENCES

[ADBS08] A. Abate, A. D'Innocenzo, Maria D. Di Benedetto, and S. Shankar Sastry. *Markov Set-Chains as Abstractions of Stochastic Hybrid Systems*, volume 1. Springer Verlag, 2008.

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

[BM00] D. Balkcom and M. T. Mason. Time Optimal Trajectories for Bounded Velocity Differential Drive Robots. In *IEEE International Conference on Robotics and Automation (ICRA) 2000*, volume 3, pages 2499 – 2504, 2000.

[CB12a] I. Cizelj and C. Belta. Control of Noisy Differential-Drive Vehicles from Time-Bounded Temporal Logic Specifications. Tecnhical report, 2012. (available online at http://arxiv.org/abs/1209.1139v4).

[CB12b] I. Cizelj and C. Belta. Probabilistically Safe Control of Noisy Dubins Vehicles. In *IEEE Intelligent Robots and Systems (IROS) Conference, 2012*, pages 2857 –2862, October 2012.

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

[FMAG98] T. Fraichard, R. Mermond, and R. Alpes-Gravir. Path Planning with Uncertainty for Car-Like Robots. In *IEEE International Conference on Robotics and Automation (ICRA) 1998*, pages 27–32, 1998.

[Gir07] A. Girard. Approximately Bisimilar Finite Abstractions of Stable Linear Systems. In *International Conference on Hybrid Systems: Computation and Control (HSCC) 2007*, pages 231–244, Berlin, Heidelberg, 2007. Springer-Verlag.

[HMZ+12] D. Henriques, J. Martins, P. Zuliani, A. Platzer, and E. M. Clarke. Statistical Model Checking for Markov Decision Processes. In *9th International Conference on Quantitative Evaluation of SysTems (QEST) 2012*, September 2012.

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

[KB08a] M. Kloetzer and C. Belta. Dealing with Non-Determinism in Symbolic Control. In *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*, 53(1):287 –297, 2008.

[KGFP07] 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 (ICRA) 2007*, pages 3116–3121, 2007.

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

[PTPZ07] R. Petrella, M. Tursini, L. Peretti, and M. Zigliotto. Speed Measurement Algorithms for Low-Resolution Incremental Encoder Equipped Drives: A Comparative Analysis. In *International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) 2007*, pages 780 –787, 2007.

[WTM09] T. Wongpiromsarn, U. Topcu, and R. M. Murray. Receding Horizon Temporal Logic Planning for Dynamical Systems. In *48th IEEE Conference on Decision and Control (CDC) 2009*, pages 5997 –6004, 2009.

[YTC+12] B. Yordanov, J. Tumova, I. Cerna, J. Barnat, and C. Belta. Temporal Logic Control of Discrete-Time Piecewise Affine Systems. *IEEE Transactions on Automatic Control*, 57(6):1491 –1504, 2012.

[ZPC10] P. Zuliani, A. Platzer, and E. M. Clarke. Bayesian Statistical Model Checking with Application to Simulink/Stateflow Verification. In *International Conference on Hybrid systems: Computation and Control (HSCC) 2010*, pages 243–252, 2010.