Probabilistic Control from Time-Bounded Temporal Logic Specifications in Dynamic Environments

A. I. Medina Ayala, S. B. Andersson, and C. Belta

Abstract— **The increasing need for real time robotic systems capable of performing tasks in changing and constrained environments demands the development of reliable and adaptable motion planning and control algorithms. This paper considers a mobile robot whose performance is measured by the completion of temporal logic tasks within a certain period of time. In addition to such time constraints, the planning algorithm must also deal with changes in the robot's workspace during task execution. In our case, the robot is deployed in a partitioned environment subjected to structural changes in which doors shift from open to closed and vice-versa. The motion of the robot is modeled as a Continuous Time Markov Decision Process and the robot's mission is expressed as a Continuous Stochastic Logic (CSL) temporal logic specification. An approximate solution to find a control strategy that satisfies such specifications is derived for a subset of probabilistic CSL formulae. Simulation and experimental results are provided to illustrate the method.**

I. INTRODUCTION

The interest in finding robust and reliable motion planning algorithms has substantially increased since the arena of applications in which robots are being used is becoming more constrained and complex. Planning in dynamic environments represents an extension to the basic path planning problem, in which, besides stationary obstacles, moving obstacles are present. As in static environments, the problem of planning in dynamic environments has many different approaches, with different assumptions about the input, and different requirements on the output. Starting with [1], [2], most of the vast number of planning algorithms considering dynamic environments have attempted to solve the reachability problem of moving from an initial to a goal position while staying within specific regions and dealing with changes in the workspace. However, while these methods solve this basic path planning problem, they fail to consider planning with temporally extended goals. In order to express these constraints, model checking based planning techniques appear to be a viable choice [3]–[8].

Research in verification and formal synthesis has significantly grown in recent years. Besides the considerable theoretical and experimental advances in verification, there is an ongoing diversification of this field into areas that at least initially appeared to be unrelated to logic and model checking. As an example, synthesis of discrete time mobile robotic systems using formal methods has been an actively explored subject by many researchers, including the authors of this paper [9], [10]. Despite the progress in this area, researchers have not explicitly considered synthesis in continuous time systems.

The dynamics of real time systems can be effectively modeled using probabilistic Markov models. Continuous Time Markov Decision Processes (CTMDPs) [11] are stochastic models that may exhibit nondeterminism between transitions in which time follows an exponential distribution. In the field of model checking, the Continuous Stochastic Logic (CSL) was introduced in [12] to specify temporal logic properties of continuous time Markov chains (CTMCs). Later on, in [13], the maximum reachability problem in uniform CTMDPs was solved using a backward greedy algorithm. More recently, a discretization technique to solve the maximum reachability problem in locally uniform CTMDPS was introduced in [14].

In this paper, we consider a changing environment with doors that open and close during a time-constrained robot mission. We solve this problem under the assumptions that the robot's elapsed time in a given region of the environment and the transition time in which the doors switch between open and closed are governed by exponential distributions. Moreover, the exponential rate at which the doors switch is given to the robot *a priori*. A CTMDP is used to model the interaction between the robot and the changing environment under these settings. We consider specifications given as CSL formulas to represent the tasks to be accomplished by the robot. The main contribution of this paper is the development of a framework for the synthesis of control strategies from such specifications to be applied in robotics applications. Although the algorithms presented are based on existing model checking and step-reachability probability algorithms, the explicit formulation and solution of a CSL synthesis problem as a maximum reachability problem are, to the best of our knowledge, novel and general. While this paper focuses on a structured environment with doors, the problem and the methods developed can be generalized to arbitrary environments that involve structural and ambient changes. To illustrate these methods, we use the Dynamic Indoor Concurrent Environment (DICE) Simulator to generate the CTMDP model for a robot moving in a dynamic environment and to show the planning of the robot.

II. PROBLEM FORMULATION AND APPROACH

Consider a mobile robot moving in an indoor environment consisting of static and moving components. The static part of the environment corresponds to regions whose topology is

This work is partially supported at Boston University by the NSF under grants CNS-0834260 and CMMI-0928776, the ARO under grant W911NF-09-1-0088, the AFOSR under grant FA9550-09-1-0209, and the ONR MURI under grant N00014-09-1051.

The authors are with the Department of Mechanical Engineering, Boston University, MA, USA, E-mail: duvinci@bu.edu, sanderss@bu.edu, cbelta@bu.edu

A. Medina Ayala is the corresponding author.

assumed to be known. The changing part of the environment is represented by the occurrence of discrete events. A discrete event is the outcome of any change in the dynamics of the environment that the robot has no control over. Such changes can correspond to transitions of the movable structures present in the environment such as doors or gates opening and closing. We call these events *switching events*.

We assume that the robot can determine its current region exactly and that it is programmed with a set of control primitives. These primitives allow the robot to move inside each region in the environment and from one region to an adjacent one provided the region is not blocked by a closed door. To account for noisy actuators, if in a given region a control designed to take the robot to a specific adjacent region is used, it is possible that the robot will instead transition to a different adjacent region. In practice, the success and failure rates of these controllers can be determined. The robot only gets information on whether a given door is open or closed when the current region contains that door.

In order to analyze the motion of the robot in the environment, a probabilistic model is used. Consequently, a change in the state of any door (open or closed) is assumed to follow a Poisson distribution [15] so that the time between two subsequent switching events is an exponential random variable. It is assumed that the rate parameters associated to these events are known. The exponential distribution is a reasonable choice to represent the time between two consecutive switching events given that it depicts the following features of the dynamic nature of the environment. First, the number of events in a given time interval is independent of the number of events in any other non-overlapping time interval. Second, the number of events in a time interval is proportional to the length of the time interval. And third, the probability that an event occurs in a time interval becomes arbitrarily small if the time interval is sufficiently small [2].

The motion of the robot is considered to evolve according to the following dynamic process. The robot starts in an initial region of the environment. After applying a control primitive, it remains in this region for a random amount of time before making a transition to a different region. The time spent in a given region and the next region to be visited are independent random variables. We assume that given a set of previous regions visited at earlier times, the robot 'forgets' all but the region visited at the most recent time. Intuitively, this feature can be captured by the memoryless property of the exponential distribution. Hence, the time that the robot stays in a particular region is modeled to be exponentially distributed. The rate parameters associated to these distributions can be determined with a combination of experiments and simulations.

The mission given to the robot will be a temporal logic statement over a fragment of the Continuous Stochastic Logic (CSL) [12]. With this setting, we consider the following problem:

Problem 1. *Given a motion specification in the form of a temporal logic statement over a set of properties satisfied by the regions in a changing indoor environment with known topology, find a control strategy that maximizes the probability that the robot satisfies the specification before t time units.* □

As an example, consider the environment shown in Fig. 1, which consists of 9 corridors (marked as C_1, ..., C_9), six intersections (I_1, ..., I_6), and four doors (d_1, ...,d_4). The static part of the environment corresponds to corridors and intersections that have the following properties: *Safe* (the robot can safely drive through a region with this property), *Relatively safe* (the robot can pass through the region but should avoid it if possible), *Unsafe* (the corresponding region should be avoided), *Fire Extinguisher* (there is an extinguisher in this region that the robot can use later in its mission), and *Destination* (a region the robot should visit). On the other hand, the dynamic part is defined by the different states that the doors can be in (open or closed). A task based on the properties of interest in the considered environment is:

Specification 1. *"In less than 3 minutes, reach a Destination region while going only through either Safe or Relatively safe regions that have a Fire Extinguisher available".*

Our approach to *Problem 1* relies on modeling the motion of the robot in the changing environment as a CTMDP, (see Sec. III-A). A control strategy will be specified as a motion primitive to be performed by the robot at each region in the environment. Since the result of each control primitive will be characterized probabilistically, the satisfaction of the specification will be defined probabilistically as well. The details of the construction of the CTMDP model are described in Sec. V. Solving *Problem 1* under the assumption that the robot has perfect knowledge of the region it is currently in reduces the problem to finding the strategy that generates the maximum probability of satisfying a CSL formula. Depending on the formula being considered, the obtained probability may admit an upper bound. This topic is developed in Sec. IV.

III. PRELIMINARIES

In this section, some concepts and notation used through the paper are introduced.

Let AP be a set of atomic propositions.

Definition 1. *[13] A Continuous Time Markov decision process (CTMDP) $\mathcal{M}$ is a tuple $(S, Act, \mathbf{R}, T, L)$ where S is a finite set of states, Act is a finite nonempty set of actions, $\mathbf{R} : S \times Act \times S \to \mathbb{R}_{\geq 0}$ is a rate function such that for each $s \in S$ there is a pair $(\alpha, s') \in Act \times S$ with $\mathbf{R}(s, \alpha, s') > 0$, $T : S \times Act \times S \to [0, 1]$ is a transition probability function such that for each $s \in S$ and $\alpha \in Act$, either $T(s, \alpha, .)$ is a probability distribution on S or $T(s, \alpha, .)$ is the null function (i.e. $T(s, \alpha, s') = 0$ for any $s' \in S$), and $L : S \to 2^{AP}$ is a labeling function.*

Given $s \in S$, with a slight abuse of notation, we use $Act(s)$ to denote the set of actions available at state s. We assume that each state has at least one outgoing transition.

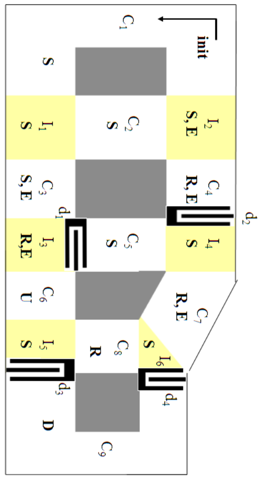

Fig. 1. Schematic representation of the dynamic environment from Fig. 2. The properties satisfied at the regions are: **S** = Safe, **R** = Relatively safe, **U** = Unsafe, **E** = fire Extinguisher, and **D** = Destination.

A CTMDP $\mathcal{M}$ is called *locally uniform* [14] if $\forall s \in S$ and $\forall \alpha, \beta \in Act(s)$, $E(s,\alpha) = E(s,\beta)$, where $E(s,\alpha) = \sum_{s' \in S} \mathbf{R}(s,\alpha,s')$ is the exit rate. In the sequel, we use $\mathbf{E}(s)$ to denote the exit rate of state s of a locally uniform CTMDP.

For a given CTMDP $\mathcal{M}$, the embedded locally uniformized discrete time Markov decision process (DTMDP) with uniformization rate $\mathbf{E}(s) = \max_{s \in S} \max_{\alpha \in Act} E(s,\alpha)$, is $(S, Act, T^{u(\mathcal{M})}, L)$, where for all $\alpha \in Act(s)$,

$$T^{u(\mathcal{M})}(s,\alpha,s') = \begin{cases} \frac{E(s,\alpha)}{\mathbf{E}(s)} T(s,\alpha,s') & \text{if } s \neq s' \\ \frac{E(s,\alpha)}{\mathbf{E}(s)} T(s,\alpha,s') + 1 - \frac{E(s,\alpha)}{\mathbf{E}(s)} & \text{if } s = s'. \end{cases}$$

A path ω of a CTMDP $\mathcal{M}$ is an infinite sequence $s_0, \alpha_0, t_0, s_1, \alpha_1, t_1, s_2, \alpha_2, t_2, \ldots$ where $s_i \in S$ is a state, $\alpha_i \in Act$ is an action and $t_i \in \mathbb{R}_{\geq 0}$ is the sojourn time in state s_i. The sequence can be written as:

$$s_0 \xrightarrow{\alpha_0, t_0} s_1 \xrightarrow{\alpha_1, t_1} s_2 \xrightarrow{\alpha_2, t_2} \ldots.$$

Any finite prefix of ω that ends in a state is a finite path of $\mathcal{M}$. The set of all non-empty finite sequences of states is denoted by $Path^{fin}$ and that of infinite ones by $Path^{inf}$.

In order to resolve the nondeterminism that occurs in the states of a CTMDP in which more than one action is allowed, we introduce the concept of timed measurable policies.

Definition 2. *[14] A timed measurable policy for a CTMDP $\mathcal{M}$ is defined as a mapping $\pi : Path^{fin} \times \mathbb{R}_{\geq 0} \times Act \to [0,1]$, such that for all $t_i \in \mathbb{R}_{\geq 0}$ and $\omega \in Path^{fin}$, the functions $\pi(\omega, t, .) : Path^{fin} \times \mathbb{R}_{\geq 0} \to [0,1]$ yield a probability distribution over all $\alpha \in Act$.*

Time measurable policies that are based on the current state and the total elapsed time are known as *late total time positional* (late TTP) policies [14]. Within this class, we consider a special set of policies. A late TTP policy is *piecewise constant and non-Zeno* if for any state $s \in S$ and any time bound t, the frequency at which actions change is finite. Such policies ensure that a finite number of decision epochs occurs up to time t. Considering time intervals of equal length, we define the concept of periodic policies.

Definition 3. *Any piecewise constant and non-Zeno late TTP policy is a periodic policy if for all $s \in S$ and $k \in \mathbb{N}$ there exists an action $\alpha \in Act$ such that for the elapsed time within the interval $[k\mathbf{T}, (k+1)\mathbf{T})$ of period $\mathbf{T}$, the chosen action is α.*

Continuous Stochastic Logic (CSL) [12] is a branching time temporal logic based on Computation Tree Logic (CTL) [16]. A CSL formula sets conditions on a state of a CTMDP. Besides the standard propositional temporal logic operators, CSL includes the probabilistic operator $\mathbb{P}_\lambda(\phi)$ where ϕ is a path formula and λ is a probability threshold. CSL also admits the steady-state probabilistic operator $\mathbb{S}_\lambda(\phi)$. $\mathbb{S}_\lambda(\phi)$ expresses that in the long-run, the steady-state probability of a ϕ-state satisfies the threshold λ. The path formulae ϕ are defined as for CTL, except that a bounded next operator $\mathcal{X}^I\ \Phi$ and a bounded until operator, $\Phi\ \mathcal{U}^I\ \Psi$ for the compact interval $I \subseteq \mathbb{R}_{\geq 0}$ are included. Formally, we have:

Definition 4. *The syntax of CSL state formulas are defined according to the following grammar rules:*

$$\Phi ::= true \mid a \mid \neg\Phi \mid \Phi_1 \vee \Phi_2 \mid \Phi_1 \wedge \Phi_2 \mid \mathbb{P}_{\sim\lambda}[\phi] \mid \mathbb{S}_{\sim\lambda}[\phi]$$

where $a \in AP$, ϕ is a path formula, $\sim \in \{<, \leq, >, \geq\}$ is a comparison operator, and $\lambda \in [0,1]$ is a probability threshold. CSL path formulae are given by:

$$\phi ::= \mathcal{X}^I\ \Phi \mid \Phi_1\ \mathcal{U}^I\ \Phi_2$$

where Φ, Φ_1, and Φ_2 are state formulae and $I \subseteq \mathbb{R}_{\geq 0} \cup \{\infty\}$.

The semantics of CSL formulae is defined on CTMDPs analogously to the semantics of PCTL formulae on MDPs [12].

As an example of the properties that can be expressed using CSL formulae, *Specification 1* on the environment in Fig. 1 translates to the formula:

$$\phi :: \mathbf{S} \vee (\mathbf{R} \wedge \mathbf{E})\ \mathcal{U}^{[0,3]}\ \mathbf{D}.$$

IV. CONTROL SYNTHESIS OF CTMDPS WITH TIMED BOUNDED SPECIFICATIONS

In this paper we restrict our attention to path CSL formulae.

A. Next Optimal Operator ($P_{\max=?}\mathcal{X}^I\ \Phi$)

Let s denote the current state of the system. The problem of finding the optimal action to satisfy the formula within the interval $I \subseteq \mathbb{R}_{\geq 0}$ in exactly one transition can be reduced to

$$\pi^*(s) = \arg\max_{\alpha \in Act(s)} (e^{-\mathbf{E}(s) \inf I} - e^{-\mathbf{E}(s) \sup I}) \sum_{s' \models \Phi} T(s,\alpha,s'). \tag{1}$$

The computation of the policy that satisfies the formula with maximum probability can be achieved by defining an indexed probability vector $\overline{\Phi}$, whose entries are equal to $e^{-\mathbf{E}(s) \inf I} - e^{-\mathbf{E}(s) \sup I}$ if $s \models \Phi$ and 0 otherwise. Then, a vector with the probabilities of satisfying $\mathcal{X}^I\ \Phi$ from each state is calculated by multiplying the transition probability matrix T and the vector $\overline{\Phi}$.

The solution of $P_{\min=?}[\mathcal{X}^I\ \Phi]$ is solved as for the $P_{\max=?}$ case. However, instead of a maximization problem, a minimization problem is solved.

B. Until ϵ-Optimal Operator ($P_{max=?}[\Phi\ \mathcal{U}^I\ \Psi]$)

The maximum probability of satisfying $\Phi\,\mathcal{U}^I\ \Psi$ is achieved by a fixed point characterization. In the sequel, assume that $I = [0,t]$ and let $\mathbf{P}(s,\alpha,s',x) = \mathbf{E}(s)e^{-\mathbf{E}(s)x}T(s,\alpha,s')$ represent the probability that a transition to state s' results after applying action α at state s within x time units for $x \leq t$. Let $p_{\max}(s,t)$ denote the maximum probability of satisfying the formula in at most t time units. The function $(s,t) \mapsto p_{\max}(s,t)$ is the fixed point of the higher-order operator $\Omega : (S\times\mathbb{R}_{\geq 0} \to [0,1]) \to (S\times\mathbb{R}_{\geq 0} \to [0,1])$, defined for all $s \in S$, $t \in \mathbb{R}_{\geq 0}$ and the measurable function $F : S\times\mathbb{R}_{\geq 0}$, such that:
$\Omega(F)(s,t) =$

$$\begin{cases} 1 & \text{if } s \models \Psi \\ \int_0^t \max\limits_{\alpha\in Act(s)} \sum\limits_{s'\in S} \mathbf{P}(s,\alpha,s',x)F(s',t-x)\mathrm{d}x & \text{if } s \models \Phi\wedge\neg\Psi \\ 0 & \text{otherwise.} \end{cases}$$

The solution to the nontrivial case when $s \models \Phi\wedge\neg\Psi$ implies solving recursively a set of Volterra integral equations. Numerical integration and transformation of the integral equation into a system of differential equations are two possible approaches that can be used to solve the equation. However, these methods have been shown to be time consuming and numerically unstable [17]. Alternatively, if we consider the first $\mathbf{T}$ units of time in the interval $[0,t]$, we can split the integral into two summands as follows:

$$\begin{aligned} p_{\max}(s,t) = &\int_0^{\mathbf{T}} \max_{\alpha\in Act(s)} \mathbf{P}(s,\alpha,s',x)\cdot p_{\max}(s',t-x)\mathrm{d}x \\ &+ \int_{\mathbf{T}}^{t} \max_{\alpha\in Act(s)} \mathbf{P}(s,\alpha,s',x)\cdot p_{\max}(s',t-x)\mathrm{d}x. \end{aligned} \tag{2}$$

The integration range of the second summand in (2) can be shifted by $-\mathbf{T}$. Additionally, for a sufficiently small $\mathbf{T}$, the first summand can be made independent of x. Solving the integrals and replacing $\mathbf{P}(s,\alpha,s',x)$ accordingly, we obtain:

$$\begin{aligned} p_{\max}(s,t) \approx &\max_{\alpha\in Act(s)} (1-e^{-\mathbf{E}(s)\mathbf{T}}) \sum_{s'\in S} T(s,\alpha,s')\cdot \\ &p_{\max}(s',t-\mathbf{T}) + e^{-\mathbf{E}(s)\mathbf{T}}\cdot p_{\max}(s,t-\mathbf{T}). \end{aligned} \tag{3}$$

The first summand in (3) is an approximation of the first integral in (2) representing the probability that only one transition occurs during the interval $[0,\mathbf{T}]$. Clearly, (3) can be interpreted as a lower bound for $p_{\max}(s,t)$. In order to define an upper bound for (2), let $\mathbf{E} = \max_{s\in S}\mathbf{E}(s)$ be the maximum exit rate of the system. In the worst case, the maximum ϵ error will be equal to $\frac{(\mathbf{ET})^2}{2}$. This result follows from the derivation of the Taylor expansion of the exponential function when considering that the number of transitions in the $[0,\mathbf{T}]$ interval follows a Poisson distribution. For a fixed ϵ upper bound, the number of steps k that satisfy $\epsilon \geq \frac{(\mathbf{E}\cdot t)^2}{2k}$ can be defined.

Based on the above approximation, a discrete time MDP $\widetilde{\mathcal{M}} = (S, Act, \widetilde{T}, L)$ is induced, where

$$\widetilde{T}(s,\alpha,s') = \begin{cases} (1-\exp^{-\mathbf{E}(s)\mathbf{T}})\cdot T(s,\alpha,s') & \text{if } s\neq s' \\ (1-\exp^{-\mathbf{E}(s)\mathbf{T}})\cdot T(s,\alpha,s') & \\ +\exp^{-\mathbf{E}(s)\mathbf{T}} & \text{if } s = s'. \end{cases}$$

For a given upper bound on the approximation error, ϵ, an ϵ-optimal policy is found solving the maximization problem:

$$\begin{aligned} \pi^*(s,k) = \arg\max_{\alpha\in Act} \Bigg(&\sum_{s'\models\Phi\wedge\neg\Psi} \widetilde{T}(s,\alpha,s')\cdot p(s',k-1) \\ &+ \sum_{s'\models\neg\Phi\wedge\Psi} \widetilde{T}(s,\alpha,s')\Bigg), \end{aligned} \tag{4}$$

where $\pi^*(s,k)$ denotes the ϵ-optimal periodic policy that applied at s generates the probability of reaching a Ψ-state starting from state s in at most k steps in the induced discrete MPD $\widetilde{\mathcal{M}}$. Therefore, computing $p^{\max}(s,t)$ up to an ϵ error can be found using dynamic programming techniques such as the value iteration method [11]. The solution for $\mathbb{P}_{\min=?}[\Phi\ \mathcal{U}^I\ \Psi]$ is found in a similar way.

C. Complexity

As for PCTL, the overall time complexity for CSL control synthesis is linear in the size of the formula and polynomial in the size of the model. Let $M = \sum_{s_i\in S}|Act(s_i)|$ be the size of the CTMDP $\mathcal{M}$. Obtaining the optimal policy for formulae of the form $P_{max=?}[\mathcal{X}^I\ \Phi]$ has $\mathcal{O}(M)$ time complexity, as only one matrix-vector multiplication and one maximization step are needed. On the other hand, for formulas of the form $P_{max=?}[\Phi\ \mathcal{U}^I\ \Psi]$, the complexity is more expensive. Assuming we start with a locally uniform or uniformized CTMDP, the discretized MDP $\widetilde{\mathcal{M}}$ used in the maximization process is built adding at the most one self-loop for each $s_i \in S$ and $\alpha \in Act(s_i)$. Therefore, in the worst case, the size of $\widetilde{\mathcal{M}}$ is $2M$. For a given error bound ϵ, the smallest number of steps k to be used while applying the value iteration algorithm is given by $\frac{(\mathbf{E}t)^2}{\epsilon}$. Hence, the time complexity of the algorithm is $\mathcal{O}(M\cdot\frac{(\mathbf{E}t)^2}{\epsilon})$ [14].

V. CTMDP MODEL OF ROBOT MOTION IN A CHANGING ENVIRONMENT

A. CTMDP Model Construction

To capture the dynamics of the environment, we use the following definition:

Definition 5. *[10] A changing environment is a tuple $\mathcal{E} = (R, D, A, C, H)$, where*

- $R = \{r_1, r_2, \ldots, r_k\}$ *is a set of k mutually disjoint regions;*
- $D = \{d_1, d_2, \ldots, d_N\}$ *is a set of N doors;*
- $A \subseteq R\times R$ *is a binary relation representing the adjacency between two regions.* $(r_1, r_2) \in A$ *denotes*

that r_1 and r_2 are adjacent and there is no door in between;

- *$C = \{c_1, c_2, \ldots, c_N\}$ is a set of Boolean variables indicating if a door is closed ($c_i = 1$) or open ($c_i = 0$), and;*
- *$H \subseteq D \times R$ is a binary relation representing the adjacency between regions and doors. A region r has a door d if $(d, r) \in H$.*

Using this definition, we define R_d as the set of regions that have a door, i.e., $R_d = \{r \in R | (d, r) \in H$ for some $d \in D\}$, and R_{nd} as the set of regions with no doors, i.e., $R_{nd} = R \setminus R_d$. The state space of the CTMDP is the union of the set of regions with no doors with the union of the set of pairs of regions containing a door and the states of that door, i.e., $S = R_{nd} \cup \bigcup_{r \in R_d} \{(r, 0), (r, 1)\}$. The action space of the CTMDP includes the set of control primitives available at each one of the static components (regions) in the environment and the decisions that allow the robot to cope with the occurrence of a switching event.

In order to represent the CTMDP rate and transition functions, we assume the following timing of events. After choosing an action in a given state, the robot remains there for an action-dependent random period of time during which the robot is moving according to the action. Then a transition occurs and the next action is chosen. We assume that the probabilities of these transitions depend only on the region the robot is currently in. Since transitions only occur at the end of a sojourn in a region, we specify the transition probabilities for the Markov model by $T(s, \alpha, s') = Pr(s'|s, \alpha)$. We assume that the sojourn time in $s \in S$ under $Act(s) \in Act$ is exponentially distributed on the interval $[0, x]$ i.e., $E(s, \alpha)e^{-E(s,\alpha)x}$, with exit rate $E(s, \alpha) = \sum_{s \in S} \mathbf{R}(s, \alpha, s')$.

As already outlined, each switching event is governed by a Poisson distribution whose rates are assumed to be known. The knowledge of these rates allows us to capture the behavior of the dynamic components of the environment and integrate it into the model of the system.

B. Dynamic Indoor Concurrent Environment (DICE)

In order to capture the motion of the robot in an indoor changing environment, the Dynamic Indoor Concurrent Environment (DICE) was developed. DICE is a simulation/experimental platform based on the Robotic InDoor Environment (RIDE) simulator [9]. It consists of a mobile robot moving in an environment with corridors and intersections delimited by RFID tags. The robot is an iRobot Create equipped with a Hokoyu URG-04LX laser range finder, an APSX RW-210 RFID reader, and a Dell Latitude 2120 netbook. The RFID tags mark the boundaries between the regions and serve as transition indicators for the robot. DICE contains doors that randomly change from open to closed and vice-versa. The time at which a door changes was programmed to follow an exponential distribution. The motion of the doors is activated by servos mounted on top of each door through a USB servo controller, which receives the commands from an off-board computer by a wireless USB hub. The samples from the exponential distributions were generated in Matlab. The simulation component of DICE was designed to capture the most important characteristics of the environment and the robot, including the robot dynamics and sensors, door transitions, transition times, and reasonable models of the noise affecting all sensors and actuators.

For this work, we utilized the environment in Fig. 1. The sojourn time in a given state was approximated to be exponential given the assumptions described in Sec. II. To enforce Markovianity in the transitions, the states of the CTMDP were defined as adjacent pairs of regions (e.g. I_2C_4 represented the state in which the robot was in I_2 and is now in C_4). As described in Sec. V.A, states containing a region with a door were duplicated to account for the two possible states the doors could be in. The model had 48 states.

The actions available at these states were: FollowRoad, GoRight, GoLeft, GoStraight, TurnAway, and Wait. The transition probabilities and rates associated to each action (with the exception of Wait) were computed after performing 1000 simulation trials. In each trial, the robot was initialized at the beginning of the region representing each state. If this region was a road, then the FollowRoad controller was applied until the system transitioned to the next state. On the other hand, if this region was an intersection, each one of the actions allowed at this state was applied and the resulting transitions and the sojourn times were recorded. The results were then integrated into the transition probabilities and rate matrices, respectively. The entries of the rate matrices were the inverse of the average sojourn times calculated in the simulations. To corroborate the accuracy of the CTMDP model obtained through simulation, five of the transition probabilities and their time averages were determined experimentally by performing 35 trials of each transition. The simulated and experimental results were then compared using the Fisher exact test [18] and determined to be statistically equivalent with a confidence of 99%.

Finally, each state of the CTMDP was labeled with the property that is satisfied at the second region of the pair representing such state.

VI. Case Study

Consider the environment shown in Fig. 1 and the motion specification given by *Specification 1* in Sec. II. This specification can be translated to the following CSL formula:

$$\phi_1 \quad :: \quad \mathbb{P}_{max=?}[\mathbf{S} \vee (\mathbf{R} \wedge \mathbf{E}) \ \mathcal{U}^{[0,3]} \ \mathbf{D}] \tag{5}$$

Discretizing the CTMDP representing the motion of the robot in the environment allows us to find an ϵ-optimal control policy for the specification given in (5). Defining an error bound equal to 0.027 and using the approach described in Sec. V for the CSL Until operator, the maximum probability for (5) was 0.345. To validate the computed probabilities, 500 simulations and 35 experimental trials were performed. The simulations demonstrated that the probability of satisfying (5) was 0.298 while the experimental trials show that this probability was 0.371. The discrepancy in the results is likely due to several factors, including statistical variation

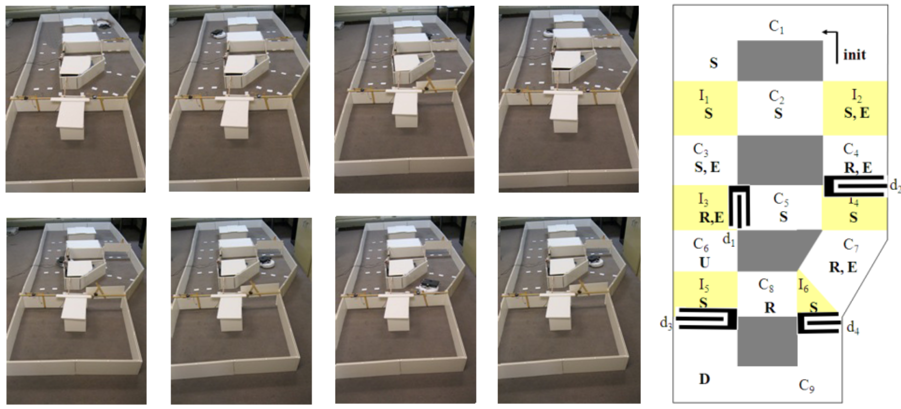

Fig. 2. Snapshots (to be read left-to-right and top-to-bottom) from a movie showing the robot's motion produced by applying the control strategy maximizing the probability of satisfying *Specification 1*.

due to the finite number of runs, differences between the real-world and simulation dynamics, and remaining non-Markovian behavior in the system.

Fig. 2 depicts scenes from a single successful experimental run in DICE according to the ϵ-optimal policy determined by the proposed approach. The robot starts its mission at C_1. In this particular run, the robot followed the path $C_1 I_1 C_2 I_2$ at which point it found door d_2 closed. It then turned away and followed the route $I_2 C_2 I_1 C_3 I_3$. Upon reaching I_3, the robot found door d_1 open and it continued driving through $C_5 I_4 C_7 I_6$. Door d_4 was closed when the robot arrived at I_6 and the robot chose to wait for it to open. Once it did, the robot passed through to arrive at C_9, achieving the task in the specification in less than three minutes.

VII. CONCLUSIONS

We presented a method for the automatic deployment of a mobile robot moving in a changing indoor environment subject to a time constrained task given in terms of a temporal logic specification. The robot's motion capabilities and the knowledge of the time distributions between the transitions of the changing elements of the environment were captured using a CTMDP. This allowed us to implement CSL model checking and dynamic programming techniques to find an optimal (ϵ-optimal) control strategy that maximizes the probability of satisfying the specification as a CSL formula.

VIII. ACKNOWLEDGMENTS

The authors would like to thank P. Galvin and K. Barry for their help with the construction of DICE and programming of the robot used in the experiments.

REFERENCES

[1] R. Sharma, "Locally efficient path planning in an uncertain, dynamic environment using a probabilistic model," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, pp. 105–110, 1992.

[2] S. LaValle and R. Sharma, "On motion planning in changing, partially-predictable environments," *International Journal of Robotics Research*, vol. 16, no. 6, pp. 775–805, 1997.

[3] S. G. Loizou and K. J. Kyriakopoulos, "Automatic synthesis of multiagent motion tasks based on LTL specifications," *Proc. IEEE Conference on Decision and Control*, pp. 153–158, 2004.

[4] M. M. Quottrup, T. Bak, and R. Izadi-Zamanabadi, "Multi-robot motion planning: A timed automata approach," *Proc. IEEE Conference on Decision and Control*, pp. 4417–4422, 2004.

[5] G. E. Fainekos, H. Kress-Gazit, and G. J. Pappas, "Hybrid controllers for path planning: A temporal logic approach," *Proc. IEEE Conference on Decision and Control*, pp. 4885–4890, 2005.

[6] ——, "Where's Waldo? sensor-based temporal logic motion planning," *Proc. IEEE Conference on Robotics and Automation*, pp. 3116–3121, 2007.

[7] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 287–297, 2008.

[8] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.

[9] M. Lahijanian, S. B. Andersson, and C. Belta, "Temporal logic motion planning and control with probabilistic satisfaction guarantees," *IEEE Transactions on Robotics*, 2010.

[10] A. Medina Ayala, S. B. Andersson, and C. Belta, "Temporal logic control in dynamic environments with probabilistic satisfaction guarantees," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 241–266, 2011.

[11] M. L. Puterman, *The Markov Decision Processes*. Wiley Series in Probability and Mathematical Statistics, 1994.

[12] C. Baier, B. Haverkort, H. Hermanns, and J. P. Katoen, "Model checking algorithms for continuous time Markov chains," *IEEE Transactions on Software Engineering*, vol. 29, no. 7, pp. 524–541, 2003.

[13] ——, "Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision process," *Theoretical Computer Science - Tools and algorithms for the construction and analysis of systems*, vol. 345, no. 1, pp. 2–26, 2005.

[14] M. R. Neuhäußer and L. Zhang, "Time-bounded reachability probabilities in continuous-time Markov decision processes," *In Quantitative Evaluation of Systems (QEST), IEEE CS Press*, pp. 209–218, 2010.

[15] S. M. Ross, *Introduction to Probability Models*. Elsevier Inc., 2007.

[16] E. A. Emerson and E. M. Clarke, "Using branching time temporal logic to synthesize synchronization skeletons," *Science of Computer Programming*, vol. 2, no. 3, pp. 241–266, 1982.

[17] H. Hermanns, J. P. Katoen, J. Meyer-Kayser, and M. Siegle, "A markov chain model checker," *Software Tools and Technology Transfer*, vol. 4, no. 2, pp. 153–172, 2003.

[18] A. Trujillo-Ortiz, R. Hernandez-Walls, A. Castro-Perez, L. Rodriguez-Cardozo, N. Ramos-Delgado, and R. Garcia-Sanchez, "Fisher's exact probability test. a Matlab file," 2004. [Online]. Available: http://www.mathworks.com/matlabcentral/