Robot Navigation Under MITL Constraints Using Time-Dependent Vector Field Based Control

Christos N. Mavridis[†], Constantinos Vrohidis[⋆], John S. Baras[†] and Kostas J. Kyriakopoulos[⋆]

***Abstract*— In this work, we consider the problem of robot navigation, under spatial and temporal constraints, modeled as Metric Interval Temporal Logic (MITL) formulas. We introduce appropriate control schemes, driven by time-dependent vector fields, that satisfy both the problems of (a) entering an arbitrary neighborhood of the workspace within a given time interval, and, (b) avoiding collision with any given obstacle. We model the problems (a) and (b) as MITL formulas, defined upon a specific class of atomic propositions, and proceed in building more complex MITL expressions that can be decomposed into a conjunction of the former formulas. Finally, we propose a way to generate a hybrid automaton, whose execution satisfies the given MITL formula, by appropriately composing the control schemes. We validate our methodology via a numerical simulation.**

I. INTRODUCTION

Motion and task planning constitute a fundamental problem in robotics, and still remain an active research topic in many respects. Many efficient approaches have been proposed, ranging from standard artificial intelligence, and temporal logic planning methods [1], [2], to methods based on artificial potential functions [3], [4], [5].

However, complex planning objectives, incorporating spatial and temporal constraints, are becoming all the more essential in robot navigation. For this reason motion planning under timed temporal logic has been recently studied [6], [7], [8]. In [6], the navigation problem is formulated as a mixed-integer optimization problem while, in [7] the system is discretized and approximated by a complex timed automaton, which, in turn, is analyzed be model checking tools.

On the other hand, methods based on the closed-loop evaluation of vector fields [3], [4], [9], although being popular amongst researchers – owing to their low complexity, and their ability to simultaneously tackle both the motion planning and control problems – they are not able to handle temporal constraints. However, the authors in [10] and [11] recast the aforementioned problem by requiring that a robot be driven to a predefined neighborhood of the desired configuration in predetermined time. They proposed a novel vector field that ensures obstacle avoidance and facilitates the use of the Prescribed Performance Control technique [12], [13] to impose predetermined convergence to the desired configuration, thus resulting in a time-varying vector field planner.

[†] Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland, College Park, MD 20742, USA. Email: {mavridis@umd.edu, baras@isr.umd.edu}.

[⋆] Control Systems Laboratory, School of Mechanical Engineering, National Technical University of Athens, Greece. Email: {vroh@mail.ntua.gr, kkyria@mail.ntua.gr}.

In order to tackle timed tasks in real-time while avoiding the increased computational complexity of timed temporal logic, the authors in [14] propose the construction of a hybrid automaton [15] that enables the decoupling of the navigation problem and the task sequencing. The hybrid automaton consists of appropriate control schemes that change subject to certain events, created by the sequence generator.

In this work, we consider the problem of real-time robot navigation in sphere world configuration spaces, which can be extended to generalized sphere worlds, upon global knowledge of the environment. We introduce a time-dependent vector field function, which results in a prescribed performance control scheme as defined in [12], such that the robot is driven from any initial configuration to an arbitrary neighborhood of the workspace within a given time interval. Additionally, we borrow from [10], and define a vector field driven control scheme, such that, collision avoidance with any given obstacle is established. We introduce a class of atomic propositions to describe the above main problems as MITL formulas, and proceed in building more complex MITL expressions that can be decomposed into these. Finally, we propose a way to generate a hybrid automaton, whose execution satisfies the given MITL formula, by appropriately composing the control schemes associated with the two main formulas. The proposed methodology is guaranteed to satisfy the MITL specifications, and is validated by a non-trivial numerical simulation.

II. PRELIMINARIES

A. Workspace and Robot Kinematics

We consider a point robot[1] operating in a bounded workspace $\mathcal{W} \subset \mathbb{R}^n$ with $n \in \mathbb{N}_{\geq 2}$ and denote its position by $\mathrm{x} \in \mathcal{W}$. The workspace is assumed to be an open ball centered at the origin

$$\mathcal{W} \triangleq \{q \in \mathbb{R}^n : \|q\| < r_{\mathcal{W}}\} \tag{1}$$

where $r_{\mathcal{W}} \in \mathbb{R}_{>0}$ is the workspace radius. The workspace can be populated with $m \in \mathbb{N}$ closed sets O_i, $i \in \mathcal{J} \triangleq \mathbb{N}_{\leq m}$, corresponding to obstacles. In particular, each obstacle $i \in \mathcal{J}$ is a ball centered at p_i with radius $r_i \in \mathbb{R}_{>0}$,

$$O_i \triangleq \{q \in \mathcal{W} : \|q - \mathrm{p}_i\| \leq r_i\}, \quad \forall i \in \mathcal{J}. \tag{2}$$

Assumption 1. *The obstacles are assumed to be static, i.e.* p_i *and* r_i *do not depend on time. This assumption will*

[1]Treating a robot with volume can be achieved by "transferring" its volume to the other workspace entities and considering it as a point.

simplify further analysis, but is not restrictive, since it can be alleviated as shown in [11].

Assumption 2 (Sphere World Assumption)**.** *The free space* $\mathcal{F} \triangleq \mathcal{W} \setminus \bigcup_{i\in\mathcal{J}} O_i$ *is assumed to be a sphere world, i.e., each obstacle* O_i *is contained in workspace* $\mathcal{W}$ *and the obstacle sets are pairwise disjoint, i.e.,* $O_i \cap O_j = \emptyset$*, for all* $i, j \in \mathcal{J}, i \neq j$.

Remark 1. *Given the global knowledge of the environment, the proposed methodology can be extended to generalized sphere worlds* $\mathcal{M}$ *provided that there exists a diffeomorphism* $\mathbf{T} : \mathcal{M} \to \mathcal{F}$ *(see [9]). The construction of such transformations is beyond the scope of this paper.*

Assumption 3 (Single Integrator Kinematics)**.** *The robot kinematics are assumed to be given by the first order holonomic kinematic model*

$$\dot{\mathrm{x}} = \mathrm{u} \tag{3}$$

where $\mathrm{x}(0) = \mathrm{x}_0 \in \mathcal{W}$ *and* $u \in U \subseteq \mathbb{R}^n$.

Remark 2. *The proposed methods can be extended to handle general known and holonomic robot kinematics, and can be potentially extended to non-holonomic or unknown systems, as well. This is beyond the scope of this paper.*

B. Metric Interval Temporal Logic (MITL)

Definition 1. *An atomic proposition* $\pi : \mathcal{W} \to \{\top, \bot\}$ *is a statement about the system variables (x) that takes the Boolean constant* **True**$(\top)$ *or* **False**$(\bot)$ *for some given values of the state variables.*

Definition 2. *The syntax of MITL formulas are defined according to the following grammar rules:*

$$\phi ::= \top \mid \pi \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \mathbf{U}_I \phi_2,$$

where $I \subseteq [0, \infty]$ *is an interval with end points in* $\mathbb{Q}_{\geq 0} \cup \{\infty\}$, π *is an atomic proposition, and* $\top$ *and* $\bot(= \neg\top)$ *are the Boolean constants true and false, respectively.*

Definition 3. *The semantics of an MITL formula* ϕ *is recursively defined over a trajectory* x_t *as:*
$x_t \models \pi$ *iff* $\pi(x_t) = \top$
$x_t \models \neg\pi$ *iff* $\pi(x_t) = \bot$
$x_t \models \phi_1 \wedge \phi_2$ *iff* $x_t \models \phi_1$ *and* $x_t \models \phi_2$
$x_t \models \phi_1 \mathbf{U}_I \phi_2$ *iff* $\exists s \in I$ *s.t.* $x_{t+s} \models \phi_2$ *and* $\forall\, s' \leq s,\ x_{t+s'} \models \phi_1$.

Thus, the expression $\phi_1 \mathbf{U}_I \phi_2$ means that ϕ_2 will be true within time interval I, and ϕ_1 must be true until ϕ_2 becomes true. In this regard, other Boolean operators can also be expressed [16], and the following MITL operators of special interest can be defined:

- $\Diamond_I \phi \equiv \top \mathbf{U}_I \phi$ meaning that ϕ will eventually become true within the time interval I, and
- $\Box_I \phi \equiv \neg\Diamond_I \neg\phi$ meaning that ϕ is always true for the time interval I.

Finally, composition of two or more of the above MITL operators can express very sophisticated specifications; for example $\Diamond_{I_1} \Box_{I_2} \phi$ means that within time interval I_1, ϕ will be true and from that instance it will always hold true for a duration equal to that of I_2.

Remark 3. *Note that MITL is defined over a dense time domain [16], here* $\mathbb{Q}_{\geq 0} \cup [\infty]$*, and that it prohibits the interval* I *to be singular, i.e., of the form* $[a, a]$, $a \in \mathbb{R} \cup [\infty]$*. As a result, MITL has no next-time operator, and formulas such as* $\phi = \Diamond_{[\tau,\tau]} \pi$*, stating that "*$\pi$ *will be* true *at precisely* $t = \tau$*" do not constitute an MITL formula.*

C. Hybrid Automaton

A Hybrid Automaton is a dynamical system that describes the evolution in time of the values of a set of discrete and continuous variables [14], [15].

Definition 4 (Hybrid Automaton)**.** *A hybrid automaton* H *is an eleven tuple* $H = (Q, X, E, U, f, \delta, Inv, guard, \rho, q_0, x_0)$*, where*

- Q *is a set of discrete states or modes;*
- X *is a set of continuous state space (normally* $X = \mathbb{R}^n$*);*
- E *is a finite set of events;*
- U *is a set of admissible controls (normally* $U \subseteq \mathbb{R}^m$*);*
- $f : Q \times X \times U \to X$ *is a vector field;*
- $\delta : Q \times X \times E \to Q$ *is a discrete state transition function;*
- $Inv \subseteq Q \times X$ *is a set defining an invariant condition (also called domain);*
- $guard \subseteq Q \times Q \times X$ *is a set defining a guard condition;*
- $\rho : Q \times Q \times X \times E \to X$ *is a reset function;*
- q_0 *is an initial discrete state;*
- x_0 *is an initial continuous state.*

III. Safe Navigation in Prescribed Time

In this section, we deal with the problem of navigation under temporal and spatial constraints, and in particular with the two main specifications met in the problem, i.e. within a given time interval, (a) enter a predefined neighborhood of the workspace, and (b) avoid any obstacles. We formulate (a) and (b) as MITL expressions, by acknowledging that the atomic propositions used in the syntax of MITL (see Section II-B), should be restricted to propositions that are useful in motion planning problems.

In this regard, we introduce a set $\mathcal{P}$ of propositions that describe the presence of the robot in a certain area of the workspace:

Definition 5. *The set* $\mathcal{P}$ *is a set of atomic propositions* p*, each described by a pair of parameters* $(\mathrm{x}_p, r_p) \in \mathcal{W} \times \mathbb{R}_{\geq 0}$*, which describe the presence of the robot in a predefined neighborhood of a point* $\mathrm{x}_p \in \mathcal{W}$*, i.e.,*

$$\text{if } p \in \mathcal{P} \text{ then}$$
$$p(x) = \top \Leftrightarrow \left\|\mathrm{x} - \mathrm{x}_p\right\| \leq r_p.$$

In what follows, we propose control laws, driven by appropriately defined time-dependent vector fields, such that for $p \in \mathcal{P}$, the MITL expressions $\phi_a = \Diamond_J p$, which states that "eventually within a time interval J enter an area of $\mathcal{W}$", and $\phi_b = \Box_I \neg p$, which states that "always on the time interval I avoid an area of $\mathcal{W}$", are satisfied.

A. Navigation within given time interval

Consider the satisfiability problem:

Problem 1 (Navigation within a given time interval)**.** *Let* $\mathrm{x}(t) \in \mathcal{W},\ t \in [0, \tau_p]$ *be a trajectory satisfying the kinematics (3). Then the problem of navigating from* $\mathrm{x}(0)$ *to a predefined neighborhood of* $\mathcal{W}$ *within a given time interval* $J = [0, \tau_p]$*, can be formulated as:*

$$\begin{aligned} \mathrm{x}_0 &\models \Diamond_J p \\ s.t. \quad p &= p_{(\mathrm{x}_p, r_p)} \in \mathcal{P} \\ J &= [0, \tau_p] \\ \mathrm{x}_0 &\triangleq \mathrm{x}(0) \in \mathcal{W}, \end{aligned}$$

which states that $\left\|\mathrm{x}(t) - \mathrm{x}_p\right\| \leq r_p$ *for some* $t \in [0, \tau_p]$.

Consider also the following problem studied in [10]:

Problem 2 (Prescribed Time Scale Navigation Problem)**.** *Assuming single integrator robot kinematics,*

$$\dot{\mathrm{x}} = \mathrm{u}$$

and for any pair of initial and final configurations $(\mathrm{x}_0, \mathrm{x}_d) \in \mathcal{W}^2$*, and any pair* (r, τ) *with* $r, \tau > 0$*, determine a time-varying controller* $\mathrm{u} : \mathbb{R}_{\geq 0} \times \mathcal{W} \to \mathbb{R}^n$ *such that the workspace space* $\mathcal{W}$ *is forward invariant and*

$$\|\mathrm{x}(t) - \mathrm{x}_d\| < r, \quad \forall t \geq \tau. \tag{4}$$

Intuitively, equation (4) means that by time τ the robot will have entered a ball of radius r centered at the desired configuration, and remain inside it thereafter.

Then, unraveling the definitions, we have that:

Proposition 1. *A solution* u*, of Problem 2 yields a closed-loop trajectory* $x(\cdot)$*, such that Problem 1 is satisfied for* $(\mathrm{x}_p, r_p, \tau_p) = (\mathrm{x}_d, r, \tau)$.

Next, we proceed in finding a control u that solves Problem 2 such that Problem 1 is satisfied. As discussed in [10], we adopt the notion of prescribed performance control technique [12], [13], and introduce a time-dependent vector field that achieves practical convergence of the robot to the desired configuration in predefined time.

By defining the squared Euclidean distance of the robot from the desired configuration as

$$\gamma(\mathrm{x}) = \|\mathrm{x} - \mathrm{x}_d\|^2 \tag{5}$$

the prescribed performance is achieved when $\gamma(\mathrm{x}(t))$ is bounded above by a strictly decreasing function of time $\rho : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$,

$$\gamma(\mathrm{x}(t)) < \rho(t), \quad \forall t \in \mathbb{R}_{\geq 0} \tag{6}$$

where ρ is a designer-specified, smooth, bounded and decreasing function of time. In this work we consider

$$\rho(t) = (\rho_0 - \rho_\infty) e^{-lt} + \rho_\infty$$

satisfying

$$\rho_0 > \gamma(\mathrm{x}_0) \tag{7}$$

$$\rho_\infty < r^2 \tag{8}$$

$$l = l(r, \tau) \geq -\frac{1}{\tau} \ln\left(\frac{r^2 - \rho_\infty}{\rho_0 - \rho_\infty}\right). \tag{9}$$

Note that by using ρ as defined above we impose $\rho(\tau) \leq r^2$ which implies $\|\mathrm{x} - \mathrm{x}_d\| < r$. We, furthermore, define

$$\xi(t, \mathrm{x}) \triangleq \frac{\gamma(\mathrm{x})}{\rho(t)} \tag{10}$$

and an increasing bijective mapping $\mathrm{T} : \mathbb{R}_{<1} \to \mathbb{R}$ of the performance domain $\mathbb{R}_{<1}$, which defines the transformed squared distance from the desired configuration $\varepsilon(\xi) \in \mathbb{R}$, as

$$\varepsilon(\xi) \triangleq T(\xi) = \ln\left(\frac{1}{1 - \xi}\right),\ \xi \in \mathbb{R}_{<1} \tag{11}$$

Our objective now is to find a control u_γ such that ε remains bounded over the trajectory $\dot{\mathrm{x}} = u_\gamma$.Taking the time derivative of ε yields

$$\dot{\varepsilon} = J_T(t, \xi)\big(\dot{\gamma} + \alpha(t)\gamma\big) \tag{12}$$

where

$$J_T(t, \xi) \triangleq \frac{\partial T(\xi)}{\partial \xi} \frac{1}{\rho(t)} > 0 \tag{13}$$

$$\alpha(t) \triangleq -\frac{\dot{\rho}(t)}{\rho(t)} > 0. \tag{14}$$

The proposed controller is then defined as

$$u_\gamma(t, \mathrm{x}) = -\left(k\varepsilon(\xi) + \frac{1}{2}\alpha(t)\right)(\mathrm{x} - \mathrm{x}_d), \quad k > 0 \tag{15}$$

and can be expressed as $u_\gamma = -\nabla U_\gamma$ where

$$U_\gamma(t, \mathrm{x}) = \frac{1}{2}\left[\left(k + \frac{\alpha(t)}{2}\right)\gamma(\mathrm{x}) + k\varepsilon(t, \mathrm{x})(\gamma(\mathrm{x}) - \rho(t))\right],\ k > 0 \tag{16}$$

An experienced reader can see that u_γ is defined such that $\dot{\gamma} + \alpha(t)\gamma = -2k\varepsilon\gamma$, which simplifies the stability analysis of (12).

Proposition 2. *The control law* u_γ *defined in (15) is a solution to Problem 2.*

Proof. It suffices to show that

$$\xi(t) \in \Omega_\xi,\ \forall t \in [0, \infty)$$

where $\Omega_\xi \triangleq [0, \bar{\xi}],\ \bar{\xi} \in [0, 1)$.

We first take the derivative of (10) with respect to time

$$\dot{\xi} = \frac{1}{\rho(t)}\left(\dot{\gamma} - \dot{\rho}(t)\xi\right). \tag{17}$$

where

$$\begin{aligned} \dot{\gamma} &= \nabla\gamma^{\mathrm{T}}\dot{\mathrm{x}} \\ &= 2(\mathrm{x} - \mathrm{x}_d)^{\mathrm{T}}\dot{\mathrm{x}} \\ &\overset{(15)}{=} -2k\varepsilon(\xi)\gamma - \alpha(t)\gamma \end{aligned} \tag{18}$$

$$\therefore \dot{\xi} = -2k\varepsilon(\xi)\xi, \ \xi(0) = \frac{\gamma(x_0)}{\rho_0} \tag{19}$$

Since the right-hand side of the initial value problem (19) is Lipschitz continuous, Theorem 54 [17] can be invoked as in [12], establishing the existence of a maximal solution $\xi(t)$ on a time interval $[0, \tau_{\max})$ such that $\xi(t) \in \mathbb{R}_{<1}$ for all $t \in [0, \tau_{\max})$. As an immediate consequence, $\varepsilon(\xi)$ is well-defined for all $t \in [0, \tau_{\max})$, so that we can define the following Lyapunov function

$$V = \frac{1}{2}\varepsilon^2 \tag{20}$$

which yields

$$\begin{aligned} \dot{V} &\overset{(12)}{=} J_T(t,\xi)\varepsilon(\xi)(\dot{\gamma} + \alpha(t)\gamma) \\ &\overset{(18)}{=} J_T\varepsilon(-2k\varepsilon\gamma) \leq 0 \end{aligned} \tag{21}$$

Thus, the positive definiteness and radial unboundedness of V with respect to ε concludes that

$$\varepsilon(\xi(t)) \leq \varepsilon_0 \triangleq \varepsilon(0) < +\infty, \quad \forall t \in [0, \tau_{\max}). \tag{22}$$

As an immediate consequence $u_\gamma(\cdot,\cdot)$ remains bounded in the interval of existence

$$\|u_\gamma(t, \mathrm{x}(t))\| \leq \bar{u}_\gamma \triangleq (k\varepsilon_0 + \bar{\alpha})\sqrt{\rho_0}, \quad \forall t \in [0, \tau_{\max}) \tag{23}$$

with $\bar{\alpha} \triangleq \alpha(0) = \frac{l(\rho_0 - \rho_\infty)}{\rho_0}$, and, by taking the inverse logarithmic function in (22), we get:

$$0 \leq \xi(t) \leq \bar{\xi} \triangleq 1 - \exp(-\varepsilon_0) < 1, \quad \forall t \in [0, \tau_{\max}). \tag{24}$$

Thus, we have shown that

$$\xi(t) \in \Omega_\xi, \ \forall t \in [0, \tau_{\max})$$

Finally, since Ω_ξ is a compact subset of $\mathbb{R}_{<1}$, invoking Proposition C.3.6 [17], the solution $\xi(t)$ of (19) can be extended for $\tau_{\max} = +\infty$, which completes the proof. $\square$

B. Obstacle Avoidance

Consider the satisfiability problem:

Problem 3 (Obstacle Avoidance)**.** *Let* $\mathrm{x}(t) \in \mathcal{W}$, $t \in [0, \tau_p]$ *be a trajectory satisfying the kinematics (3). Then the problem of not entering a predefined neighborhood of* $\mathcal{W}$ *throughout a given time interval* $J = [0, \tau_p]$*, can be formulated as:*

$$\begin{aligned} & \mathrm{x}_0 \models \square_J \neg p, \\ s.t. \quad & p = p_{(\mathrm{x}_p, r_p)} \in \mathcal{P} \\ & J = [0, \tau_p] \\ & \mathrm{x}_0 \triangleq \mathrm{x}(0) \in \mathcal{W} \setminus \{q \in \mathcal{W} : \|q - \mathrm{x}_p\| \leq r_p\} \end{aligned}$$

which states that $\|\mathrm{x}(t) - \mathrm{x}_p\| > r_p$ *for all* $t \in [0, \tau_p]$.

Consider also the following problem:

Problem 4 (Obstacle Avoidance)**.** *Assuming single integrator robot kinematics,*

$$\dot{\mathrm{x}} = \mathrm{u}$$

and for any obstacle $O_i \triangleq \{q \in \mathcal{W} : \|q - \mathrm{p}_i\| \leq r_i\}$, $i \in \mathcal{J}$, *any initial configuration* $\mathrm{x}_0 \in \mathcal{F}_i \triangleq \mathcal{W} \setminus O_i$, *and any* $\tau > 0$, *determine a time-varying controller* $\mathrm{u} : \mathbb{R}_{\geq 0} \times \mathcal{F}_i \to \mathbb{R}^n$ *such that the free space* $\mathcal{F}_i$ *is forward invariant.*

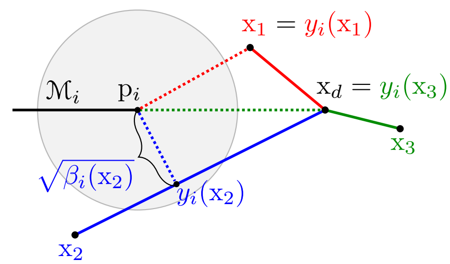

Fig. 1. An illustration of the three cases present in (28). Red, blue, and green correspond to values of λ_i in $\mathbb{R}_{\leq 0}$, $(0,1)$, and $\mathbb{R}_{\geq 1}$, respectively.

Intuitively, $\mathcal{F}_i$ being forward invariant means that for every $t \in [0, \tau]$ the robot will avoid obstacle O_i.

Proposition 3. *A solution* u*, of Problem 4 yields a closed-loop trajectory* $x(\cdot)$*, such that Problem 3 is satisfied for* $(\mathrm{x}_p, r_p, \tau_p) = (\mathrm{p}_i, r_i, \tau)$.

We adopt the vector field β_i, $i \in \mathcal{J}$ first introduced in [10] to define a feedback law u_{β_i} that solves Problem 4 such that Problem 3 is satisfied.

Given a robot configuration $\mathrm{x} \in \mathcal{F}_i$ and for a given desired configuration $\mathrm{x}_d \in \mathcal{F}_i$, by denoting

$$\mathcal{S}(\mathrm{x}) \triangleq \{q \in \mathbb{R}^n : q = (1-\lambda)\mathrm{x} + \lambda \mathrm{x}_d, \ \lambda \in [0,1]\} \subset \mathcal{W}, \tag{25}$$

we define the locally Lipschitz continuous vector field (as proved in [10]) $\beta_i : \mathcal{W} \setminus \{\mathrm{x}_d\} \to \mathbb{R}_{\geq 0}$ by

$$\beta_i(\mathrm{x}) = \inf\left\{\|q - \mathrm{p}_i\|^2 : q \in \mathcal{S}(\mathrm{x})\right\}, \quad i \in \mathcal{J} \tag{26}$$

which is the squared Euclidean distance of the set $\mathcal{S}(\mathrm{x})$ from the obstacle center p_i. Due to $\mathcal{S}(x)$ being closed and convex, there exists a *unique point* $y_i(\mathrm{x}) \in \mathcal{S}(\mathrm{x})$ such that $\beta_i(\mathrm{x}) = \|y_i(\mathrm{x}) - \mathrm{p}_i\|^2$ and, by (25), a unique $\lambda_i(\mathrm{x}) \in [0,1]$, such that

$$y_i(\mathrm{x}) = \mathrm{x} - \lambda_i(\mathrm{x})(\mathrm{x} - \mathrm{x}_d). \tag{27}$$

where

$$\lambda_i(\mathrm{x}) = \begin{cases} 0, & \tilde{\lambda}(\mathrm{x}) \in \mathbb{R}_{\leq 0} \\ \tilde{\lambda}_i(\mathrm{x}), & \tilde{\lambda}(\mathrm{x}) \in (0,1) \\ 1, & \tilde{\lambda}(\mathrm{x}) \in \mathbb{R}_{\geq 1} \end{cases} \tag{28}$$

with $\tilde{\lambda}_i(\mathrm{x}) = \frac{(\mathrm{x} - \mathrm{p}_i)^{\mathrm{T}}(\mathrm{x} - \mathrm{x}_d)}{\|\mathrm{x} - \mathrm{x}_d\|^2}$.

Now we are in place to define the controller u_{β_i} using the vector field β_i from (26):

$$u_{\beta_i}(\mathrm{x}) \triangleq \frac{\sigma_{i,\delta}(\mathrm{x})}{d_i(\mathrm{x})}\left(\nabla\beta_i(\mathrm{x}) + \mathbb{1}_{\mathcal{M}_i}(\mathrm{x})\,(\mathrm{x} - \mathrm{x}_d)^{\perp}\right) \tag{29}$$

where

- $d_i(x) = \|\mathrm{x} - \mathrm{p}_i\|^2 - r_i^2$

- $\sigma_{i,\delta}(\mathrm{x}) : \mathbb{R} \to [0,1]$ are C^1 switches that make the effect of the obstacle O_i local, and are defined as:

$$\sigma_{i,\delta}(x) = \begin{cases} 1, & d_i(x) \in \mathbb{R}_{\leq 0} \\ 2\left(\frac{d_i(x)}{\delta_i}\right)^3 - 3\left(\frac{d_i(x)}{\delta_i}\right)^2 + 1, & d_i(x) \in (0, \delta_i) \\ 0, & d_i(x) \in \mathbb{R}_{\geq \delta_i} \end{cases}$$

where $\delta_i = \delta_i(\delta) \triangleq \delta(\delta + 2r_i)$,
- $\delta \in (0, \bar{r})$, for some $\bar{r}$ such that

$$\|\mathrm{p}_i - \mathrm{p}_j\| > r_i + r_j + 2\bar{r}, \quad \forall i, j \in \mathcal{J}, i \neq j$$

and

$$\inf_{q \in \partial \mathcal{W}} \|q - \mathrm{p}_i\| > r_i + 2\bar{r}, \quad \forall i \in \mathcal{J}.$$

- $\nabla \beta_i(\mathrm{x})$ is given by

$$\nabla \beta_i(\mathrm{x}) = \begin{cases} 2(\mathrm{x} - \mathrm{p}_i), & \lambda_i(\mathrm{x}) \in \{0\} \\ 2\big(1 - \lambda_i\big)\big(\mathrm{x} - \mathrm{p}_i - \lambda_i(\mathrm{x} - \mathrm{x}_d)\big), & \lambda_i(\mathrm{x}) \in (0,1) \\ 0, & \lambda_i(\mathrm{x}) \in \{1\} \end{cases} \tag{30}$$

- the term $\mathbb{1}_{\mathcal{M}_i}(\mathrm{x} - \mathrm{x}_d)^{\perp}$, where $(\mathrm{x} - \mathrm{x}_d)^{\perp}$ denotes a vector normal to $\mathrm{x} - \mathrm{x}_d$, and $\mathbb{1}_{\mathcal{M}_i}$ denotes the indicator function of the set $\mathcal{M}_i \triangleq \{q \in \mathcal{W} \setminus \{\mathrm{x}_d\} : q = \mathrm{p}_i + \mu(\mathrm{p}_i - \mathrm{x}_d), \mu \in \mathbb{R}_{\geq 0}\}$, introduces a discontinuity necessary to prevent the robot from remaining in the set $\mathcal{M}_i$ which would imply that forward invariance of the free space would be impossible to establish for all initial configurations in $\mathcal{F}_i$. These discontinuities are confined to a set of zero measure and are both unstable and non-attractive and thus the existence of the closed-loop system solution can be established [10].

Proposition 4. *The control law u_{β_i} defined in (29) is a solution to Problem 4.*

Proof. The detailed proof is given in [10]. □

C. Composition of the proposed controllers

In this section we investigate both the problems of navigation and obstacle avoidance in prescribed time, by appropriately composing the controllers defined in (15), and (29).

Consider the satisfiability problem:

Problem 5 (Safe navigation in prescribed time)**.** *Let $\mathrm{x}(t) \in \mathcal{W}$, $t \in [0, t_f]$ be a trajectory satisfying the kinematics (3), and $O_i \triangleq \{q \in \mathcal{W} : \|q - \mathrm{p}_i\| \leq r_i\}$, $\forall i \in \mathcal{J}$, represent the obstacles of the workspace $\mathcal{W}$, as defined in (2). Then the problem of safe navigation in prescribed time can be formulated as:*

$$\begin{aligned} & \mathrm{x}_0 \models (\Diamond_J p) \wedge \left(\bigwedge_{i \in \mathcal{J}} \Box_I \neg p_i \right) \\ s.t. \quad & p = p_{(\mathrm{x}_p, r_p)} \in \mathcal{P} \\ & p_i = p_{(\mathrm{x}_i, r_i)} \in \mathcal{P}, \; i \in \mathcal{J} \\ & J = [0, \tau] \\ & I = [0, t_f] \\ & \mathrm{x}_0 \triangleq \mathrm{x}(0) \in \mathcal{W} \setminus \bigcup_{i \in \mathcal{J}} O_i \\ & \tau \leq t_f \end{aligned}$$

Then it follows from the work of Vrohidis et. al in [10] that, given the controllers u_γ and u_{β_i}, $i \in \mathcal{J}$ as defined in (15), and (29), the following proposition holds

Proposition 5. *The controller*

$$u(t, \mathrm{x}) \triangleq u_\gamma(t, \mathrm{x}) + u_\beta(\mathrm{x}) \tag{31}$$

where

$$u_\beta(\mathrm{x}) \triangleq \sum_{i \in \mathcal{J}} u_{\beta_i}(\mathrm{x}) \tag{32}$$

along with system (3), yields $\mathrm{x}(t)$, $t \in I$ that satisfies Problem 5, for $(\mathrm{x}_p, r_p) = (\mathrm{x}_d, r)$, and $(\mathrm{x}_i, r_i) = (\mathrm{p}_i, r_i)$, $\forall i \in \mathcal{J}$, as long as the obstacle sets are pairwise disjoint, i.e., $O_i \cap O_j = \emptyset$, for all $i, j \in \mathcal{J}$, $i \neq j$, and $\|\mathrm{p}_i - \mathrm{p}_j\| > r_i + r_j + 2\bar{r}$, $\forall i, j \in \mathcal{J}$, $i \neq j$.

IV. Complex MITL Expressions in Robot Navigation

In this section, we investigate more complex MITL expressions that are useful in robot navigation, and can be decomposed into a conjunction of the simpler formulas $\phi_a = \Diamond_J p$ and $\phi_b = \Box_I \neg p$ as defined in Section III. Using the controllers u_γ and u_{β_i}, $i \in \mathcal{J}$ defined in (15), and (29), we define time-varying control schemes, modeled as hybrid automata (II-C), such that the original MITL expressions are satisfied,

A. Task Execution

Because of our adherence to the set of atomic propositions $\mathcal{P}$ (Definition 5), some MITL expressions become indifferent in robot navigation applications. For example, given the robot kinematics (3), the formula $\Box_I p = \top$, $p \in \mathcal{P}$, is trivially satisfied by $u \equiv 0$ if $\mathrm{x}(0) \models p$, or not at all if $\mathrm{x}(0) \not\models p$.

In this regard, it would be useful to request that the robot, once it is close to a desired configuration, execute a specific task, e.g. move in a certain way or grasp an object. We describe the execution of a task as an atomic proposition q belonging to the following set $\mathcal{Q}$:

Definition 6. *The set $\mathcal{Q}$ is a set of atomic propositions q, each described by a pair of parameters (x_q, r_q), and a function $f_q : \mathbb{R}_{\geq 0} \times C^n[0, \infty) \times \mathcal{W} \times \mathbb{R}_{>0} \to \mathcal{W}$ which describe the presence*

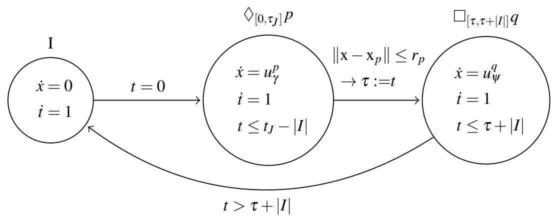

Fig. 2. A hybrid automaton satisfying the expression $\phi_{ea}=\Diamond_J\Box_I q,\ q\in\mathcal{Q},\ J=[0,t_J]$. Note that ϕ_{ea} can, equivalently, be written as $\phi_{ea}=\Diamond_{[0,\tau_J]}p\wedge\Box_{[\tau,\tau+|I|]}q$ where $\tau_J=t_J-|I|],\ \tau=\min\{t\in[0,\tau_J]:\mathrm{x}(t)\models p\}$.

of the robot in a predefined neighborhood of a point $\mathrm{x}_q\in\mathcal{W}$*, and the execution of a given task, respectively, i.e.,*

$$\begin{gathered} if\ \ q\in\mathcal{Q}\ \ then \\ q(x)=\top\Leftrightarrow\begin{cases}\left\|\mathrm{x}-\mathrm{x}_q\right\|\le r_q\\ f_q(t,\mathrm{x}(\cdot),\mathrm{x}_q,r_q)=0\end{cases}\end{gathered}$$

where t *denotes time,* x *denotes the position of the robot at the given time, and* $\mathrm{x}(\cdot)\in C^n(\mathbb{R})$ *denotes the trajectory of the robot as a function of time.*

Assumption 4. *We assume that every atomic proposition* $q\in\mathcal{Q}$ *used, is associated with a time interval* $I=[0,t_f]$ *and a controller* $u_\psi(\cdot)$ *(that may depend on the function* $\mathrm{x}(\cdot)$*), such that*

$$\dot{\mathrm{x}}(t)=u_\psi(t)\Leftrightarrow\begin{cases}\left\|\mathrm{x}(t)-\mathrm{x}_q\right\|\le r_q\\ f_q(t,\mathrm{x},\mathrm{x}_q,r_q)=0\end{cases}\quad ,t\in I$$

i.e., $\phi_q=\Box_I q$ *is satisfied, provided that* $\left\|\mathrm{x}(0)-\mathrm{x}_q\right\|\le r_q$*. In other words, we request that the task can be executed by a closed loop system defined by the designer of the motion planning specifications, and ensures that* $\left\|\mathrm{x}-\mathrm{x}_q\right\|\le r_q$ *will always hold in* I*.*

Therefore, in view of Definition 6, the expression

$$\Diamond_J\Box_I q,\ q\in\mathcal{Q}$$

now takes the meaning that within time interval $J=[0,t_J]$, the robot will have entered an area of $\mathrm{x}_q\in\mathcal{W}$, and from that time, say $\tau\in J$, and until $\tau+|I|$, where $|I|=\int_{t\in I}dt$ is the duration of I, the robot will be executing the task $f_q(t,\mathrm{x},\mathrm{x}_q,r_q)=0$. Noting that $\mathcal{Q}$ reduces to $\mathcal{P}$ for the trivial choice of $f_q(t,\mathrm{x},\mathrm{x}_s,r_q)\equiv 0$, i.e. when there is no additional task to be executed, we get the following useful relation:

$$\Diamond_J\Box_I q\equiv\Diamond_{[0,\tau_J]}p\wedge\Box_{[\tau,\tau+|I|]}q \tag{33}$$

where $\tau_J=t_J-|I|$, $\tau=\min\{t\in[0,\tau_J]:\mathrm{x}(t)\models p\}$, and $p\in\mathcal{P}$ and $q\in\mathcal{Q}$ are associated by $\left\{\mathrm{x}_q,r_q\right\}=\left\{\mathrm{x}_p,r_p\right\}$.

It follows from (33) and Assumption 4 that

Proposition 6. *The expression*

$$\phi_{ea}=\Diamond_J\Box_I q,\ q\in\mathcal{Q},\ J=[0,t_J]$$

is satisfied by $\mathrm{x}(0)$*, if* $\dot{\mathrm{x}}=u_{ea}$*, with*

$$u_{ea}=\begin{cases}u_\gamma^p(t,\mathrm{x}), & t\in[0,\tau]\\ u_\psi^q(t), & t\in[\tau,\tau+|I|]\end{cases} \tag{34}$$

where $\tau=\min\{t\in[0,\tau_J]:\mathrm{x}(t)\models p\}$*,* $\tau_J=t_J-|I|$*,* u_γ^p *is defined as in (15) for the proposition p which is associated with* q *by* $\left\{\mathrm{x}_q,r_q\right\}=\left\{\mathrm{x}_p,r_p\right\}$*, and* u_ψ^q *is designed for proposition* q *and is known (Assumption 4).*

We can use the controller defined in (34) to construct a hybrid automaton that satisfies the formula $\phi_{ea}=\Diamond_J\Box_I q,\ q\in\mathcal{Q},\ J=[0,t_J]$. Because of space limitation, we omit the formal definition of the hybrid automaton and proceed with the usual graphical representation in Figure 2.

Remark 4. *We note that the time instance* τ *in (33) and (34) is not known a priori, but results from the execution of the system (3) with* $u=u_{ea}$*. In fact, time* τ *essentially corresponds to the event* $\mathrm{x}(t)\models p$ *triggering an appropriately defined state transition function of the hybrid automaton, as shown in Fig. 2. As a result, such time* τ *may not exist at all, in which case, the problem is not satisfiable (see Section IV-D).*

B. Precedence Constraints

It is easy to see that, by definition:

$$\phi_1\mathbf{U}_I\phi_2\equiv\Box_{[0,\tau]}\phi_1\wedge\Diamond_{\left[0,t_f\right]}\phi_2$$

where $I=\left[0,t_f\right]$, ϕ_1,ϕ_2 are MITL formulas, and $\tau\in I$ is such that $\tau=\min\{t\in I:\mathrm{x}(t)\models\phi_2\}$.

An important use of the timed Until operator in motion planning is to impose *precedence constraints*, i.e., a proposition p_1 be satisfied prior to another, p_2, which is captured in the MITL expression:

$$\neg p_2\mathbf{U}_I p_1\wedge\Diamond_J p_2\equiv\Diamond_I p_1\wedge\Box_{[0,\tau]}\neg p_2\wedge\Diamond_{\left[\tau,t_{Jf}\right]}p_2 \tag{35}$$

where $I=\left[0,t_{If}\right]$, $J=\left[0,t_{Jf}\right]$, $t_{Jf}>t_{If}$, $p_1,p_2\in\mathcal{P}$, and $\tau=\min\{t\in I:\mathrm{x}(t)\models p_1\}$ (see Remark 4).

It follows from eq. (35) and Proposition 5 that

Proposition 7 (Precedence Constraints)**.** *The expression*

$$\neg p_2\mathbf{U}_I p_1\wedge\Diamond_J p_2,\ p_1,p_2\in\mathcal{P}$$

where $I=\left[0,t_{If}\right]$*, and* $J=\left[0,t_{Jf}\right]$ *is satisfied by* $\mathrm{x}(0)$*, if* $\dot{\mathrm{x}}=u_{pr}$*, with*

$$u_{pr}=\begin{cases}u_\gamma^{p_1}(t,\mathrm{x})+u_\beta^{p_2}(\mathrm{x}), & t\in[0,\tau_1]\\ u_\gamma^{p_2}(t,\mathrm{x}), & t\in[\tau_1,\tau_2]\end{cases} \tag{36}$$

where $\tau_1=\min\{t\in I:\mathrm{x}(t)\models p_1\}$*, and* $\tau_2=\min\left\{t\in[\tau_1,t_{Jf}]:\mathrm{x}(t)\models p_2\right\}$ *(see Remark 4),* $u_\gamma^{p_1}$*,* $u_\gamma^{p_2}$ *are defined as in (15) for propositions* p_1 *and* p_2*. respectively, and,* $u_\beta^{p_2}$ *is defined as in (32) for* $\mathcal{J}=\{1\}$*, and an obstacle* O_1 *defined by the parameters of proposition* p_2*.*

We can use the controller defined in (36) to construct a hybrid automaton that satisfies the formula $\phi_{pr}=\neg p_2\mathbf{U}_I p_1\wedge\Diamond_J p_2,\ p_1,p_2\in\mathcal{P}$. Due to space limitation, we only provide the graphical representation in Figure 3.

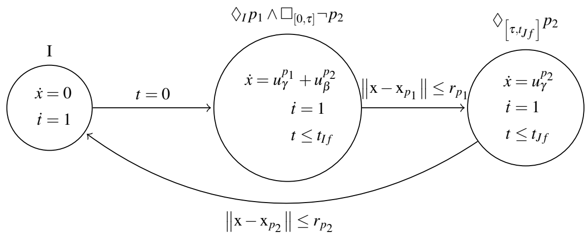

Fig. 3. A hybrid automaton satisfying the expression $\phi_{pr} = \neg p_2 \mathbf{U}_I p_1 \wedge \Diamond_J p_2,\ \ p_1, p_2 \in \mathcal{P}$, where $I = \left[0, t_{If}\right]$, $J = \left[0, t_{Jf}\right]$. Note that ϕ_{pr} can, equivalently, be written as $\phi_{pr} = \Diamond_I p_1 \wedge \Box_{[0,\tau]} \neg p_2 \wedge \Diamond_{\left[\tau, t_{Jf}\right]} p_2$ where $\tau = \min\left\{t \in I : \mathrm{x}(t) \models p_1\right\}$.

C. A complex example

Consider the following motion planning specification:

"Until time t_u, enter the sphere $\{\mathrm{x} \in \mathcal{W} : \|\mathrm{x} - \mathrm{x}_a\| \leq r_a\}$, and, for time duration of $t_a < t_u$, do the task $f_a(t, \mathrm{x}, \mathrm{x}_a, r_a) = 0$. After this is finished $(t \leq t_u)$, enter the sphere $\{\mathrm{x} \in \mathcal{W} : \|\mathrm{x} - \mathrm{x}_b\| \leq r_b\}$ before time $t_e > t_u$, and, for time duration of t_b, do the task $f_b(t, \mathrm{x}, \mathrm{x}_b, r_b) = 0$. Meanwhile, at all times, avoid the obstacles $\{\mathrm{x} \in \mathcal{W} : \|\mathrm{x} - \mathrm{p}_i\| \leq r_i\},\ i \in \mathcal{J}$."

This is captured by the following MITL formula:

$$\phi = \left(\Diamond_{[0, t_e + t_b]} \Box_{[0, t_b]} B^q\right) \wedge \left(\neg B^q \mathbf{U}_{[0, t_u]} \Box_{[0, t_a]} A^q\right) \wedge \left(\bigwedge_{i \in \mathcal{J}} \Box \neg O_i\right) \tag{37}$$

where $t_a < t_u < t_e$, $A^q, B^q \in \mathcal{Q}$ are associated with $A^p, B^p \in \mathcal{P}$ via $\left(x_p^A, r_p^A\right) = \left(x_q^A, r_q^A\right)$, and $\left(x_p^B, r_p^B\right) = \left(x_q^B, r_q^B\right)$, and $O_i \in \mathcal{P},\ i \in \mathcal{J}$.

Following the same methodology we can write:

$$\phi = \Diamond_{[0, t_u - t_a]} A^p \wedge \Box_{[\tau_1, \tau_1 + t_a]} A^q \wedge \Box_{[0, \tau_1 + t_a]} \neg B^p \wedge \Diamond_{[\tau_1 + t_a, t_e]} B^p \wedge \Box_{[\tau_2, \tau_2 + t_b]} B^q \wedge \left(\bigwedge_{i \in \mathcal{J}} \Box \neg O_i\right)$$

where $t_a < t_u < t_e$, $\tau_1 = \min\{t \in [0, t_u - t_a] : \mathrm{x}(t) \models A^p\}$, and $\tau_2 = \min\{t \in [\tau_1 + t_a, t_e] : \mathrm{x}(t) \models B^p\}$ (see Remark 4).

Finally we can construct a hybrid automaton that satisfies the formula ϕ. Due to space limitation, we provide only the graphical representation in Figure 4.

D. Limitations

At first, we note that we are not addressing a decision, or optimization, problem, but instead propose a control scheme that satisfies a given MITL formula. As a result, formulas such as $\phi_1 = \Diamond_I (p_1 \vee p_2),\ \ p_1, p_2 \in \mathcal{P}$, are not of particular interest and cannot be handled by the given approach. We assume that this has been taken care of by the designer of the MITL specifications.

Secondly, we stress that, as discussed in [14], our methodology reduces the time-constrained navigation problem into a scheduling problem for the generated hybrid automaton. We generate the hybrid automaton by decomposing the given MITL expression ϕ, into $\phi = \phi_1 \wedge \ldots \wedge \phi_k$ such that the controllers u_i that make the system (3) satisfy ϕ are

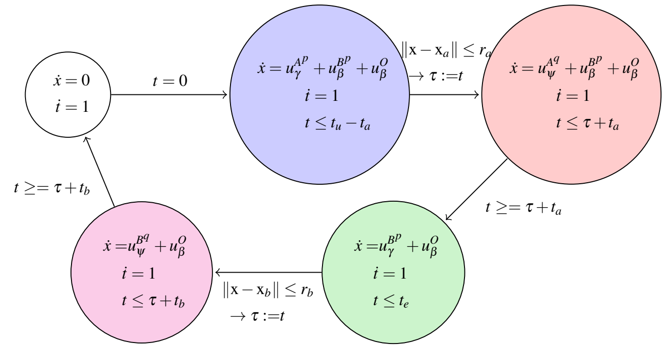

Fig. 4. A hybrid automaton satisfying the expression $\phi = \left(\Diamond_{[0, t_e + t_b]} \Box_{[0, t_b]} B^s\right) \wedge \left(\neg B^s \mathbf{U}_{[0, t_u]} \Box_{[0, t_a]} A^s\right) \wedge \left(\bigwedge_{i \in \mathcal{J}} \Box \neg O_i\right)$.

known for each $\phi_i,\ i = 1, \ldots, k$. In order to do that, we address the scheduling problem, using the auxiliary variables $\tau_i \in \{t : \mathrm{x}(t) \models \phi_i\}, i = 1, \ldots, k$, to define $\phi_j, j \neq i$, as, for example, in (33) and (34). It follows from the above and from Remark 4, that the time instances $\tau_i,\ i = 1, \ldots, k$ are not uniquely defined and may not exist. Therefore, the task sequencing problem can be further addressed on its own, especially under constraints on the robot dynamics, or the control input.

V. Simulation Results

To demonstrate the applicability of the proposed methodology, we simulate the hybrid automaton generated for the MITL formula (37), as shown in Figure 4.

The workspace is defined as the open ball $\mathcal{W} \triangleq \{q \in \mathbb{R}^2 : \|q\| < 15\}$. The parameter values are chosen such that $t_a = t_b = 3$, $t_u = 10$, $t_e = 22$, $\left(x_q^A, r_q^A\right) = \left(x_p^A, r_p^A\right) = ([0, 0], 1)$, and $\left(x_q^B, r_q^B\right) = \left(x_p^B, r_p^B\right) = ([3, 4], 2)$. The parameters of the obstacles $O_i,\ i \in \mathcal{J} \triangleq \{1, 2, 3, 4, 5\}$ satisfying the sphere world assumption (Assumption 2) are provided in Table I. The task $f_a(t, \mathrm{x}, \mathrm{x}_a, r_a) = 0$ is defined such that the robot moves in a clockwise circular trajectory in a constant distance $d_a \leq r_a = r_q^A$ around $x_a = x_q^A$ with a constant speed of 2. This is achieved by the closed-loop control law

$$u_\psi^{A^q}(\mathrm{x}) = 2 \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \frac{\mathrm{x} - x_a}{\|\mathrm{x} - x_a\|},$$

where $d_a = \|x(\tau) - x_a\|$ is the distance from x_a calculated at time τ when the controller first gets activated.

The task $f_b(t, \mathrm{x}, \mathrm{x}_b, r_b) = 0$ is defined such that the robot converges to $x_b = x_q^B$ in finite time. This is achieved by the closed-loop control law

$$u_\psi^{B^q}(\mathrm{x}) = -k_b \|\mathrm{x} - x_b\|^{-\frac{1}{2}} (\mathrm{x} - x_b),$$

for $k_b = \frac{2}{3}\sqrt{2}$. The robot kinematics are given by (3), and the controllers u_γ and u_β are as defined in (15), and (32). The parameter δ of (32) was set equal to 10. The parameters of controller u_γ are provided in Figure 6. The trajectory of the closed-loop hybrid system is depicted in Figure 5. The coloring of the trajectory is in correspondence with the coloring of the states of the hybrid automaton (see Figure 4

TABLE I

NUMERICAL VALUES OF OBSTACLES PARAMETERS

Center (p_i)	$[0,5]$	$[6,3]$	$[-6,3]$	$[0,-7]$	$[6,-5]$	$[-6,-5]$	$[10,0]$	$[-10,0]$	$[3,-0.9]$	$[-3,-0.9]$
Radius (r_i)	3	2	2	3	2	2	2	2	2	2

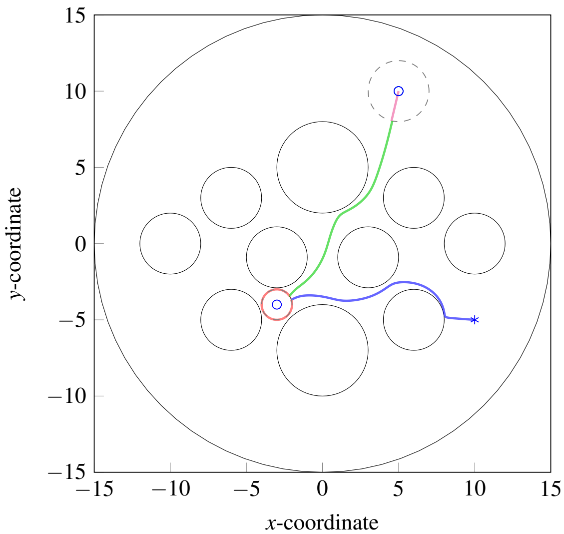

Fig. 5. Workspace and continuous solution of the hybrid automaton. Black circles correspond to the boundary of the obstacles. The asterisk indicates the initial configuration, and the blue circles correspond to the points x_q^A and x_q^B. The coloring of the trajectory is in correspondence with the coloring of the states of the hybrid automaton (Fig. 4).

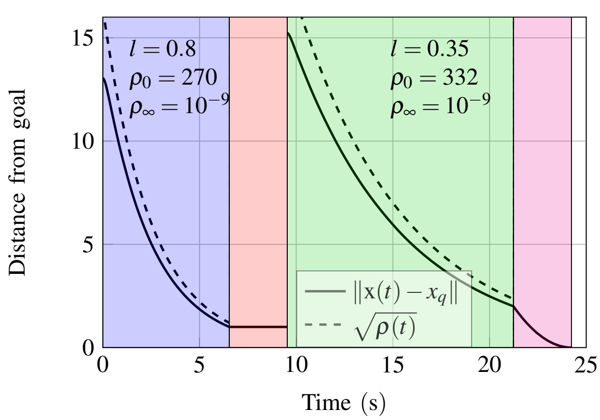

Fig. 6. Temporal evolution of the distance from points x_q^A and x_q^B. The coloring of the trajectory is in correspondence with the coloring of the states of the hybrid automaton (Fig. 4).

for comparison). Finally, Figure 6 illustrates the satisfaction of the temporal specifications described by the associated MITL formula.

VI. CONCLUSION

We have considered the problem of robot navigation, under spatial and temporal constraints, modeled as MITL formulas. We introduced appropriate control schemes, driven by time-dependent vector fields, and proposed a way to generate hybrid automata, whose execution satisfies the given MITL formulas, by appropriately composing the control schemes. Our methodology was validated via a complex numerical simulation. Future research efforts will be focused on the limitations of this work discussed in Section IV-D, and in particular on handling constraints in control input and robot kinematics.

VII. ACKNOWLEDGMENT

The work of Christos Mavridis and John Baras was partially supported by ONR grant N00014-17-1-2622.

REFERENCES

[1] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.

[2] G. E. Fainekos, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for mobile robots," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 2020–2025.

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[4] W. Xi, X. Tan, and J. S. Baras, "Gibbs sampler-based path planning for autonomous vehicles: Convergence analysis," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 49–54, 2005.

[5] X. Tan, W. Xi, and J. S. Baras, "Decentralized coordination of autonomous swarms using parallel gibbs sampling," *Automatica*, vol. 46, no. 12, pp. 2068–2076, 2010.

[6] Y. Zhou, D. Maity, and J. S. Baras, "Optimal mission planner with timed temporal logic constraints," pp. 759–764, July 2015.

[7] ——, "Timed automata approach for motion planning using metric interval temporal logic," in *2016 European Control Conference (ECC)*, June 2016, pp. 690–695.

[8] J. Fu and U. Topcu, "Computational methods for stochastic control with metric interval temporal logic specifications," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 7440–7447.

[9] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential fields," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[10] C. Vrohidis, P. Vlantis, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Prescribed time scale robot navigation," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1191–1198, April 2018.

[11] ——, "Prescribed time scale robot navigation in dynamic environments," pp. 1803–1808, June 2018.

[12] C. P. Bechlioulis and G. A. Rovithakis, "A low-complexity global approximation-free control scheme with prescribed performance for unknown pure feedback systems," *Automatica*, vol. 50, no. 4, pp. 1217 – 1226, 2014.

[13] C. N. Mavridis, K. Alevizos, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Human-robot collaboration based on robust motion intention estimation with prescribed performance," pp. 249–254, June 2018.

[14] C. C. Constantinou and S. G. Loizou, "Automatic controller synthesis of motion-tasks with real-time objectives," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 403–408.

[15] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Berlin, Heidelberg: Springer-Verlag, 2006.

[16] R. Alur, T. Feder, and T. A. Henzinger, "The benefits of relaxing punctuality," *J. ACM*, vol. 43, no. 1, pp. 116–146, Jan. 1996. [Online]. Available: http://doi.acm.org/10.1145/227595.227602

[17] E. D. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems (2nd Ed.)*. New York, NY, USA: Springer-Verlag New York, Inc., 1998.