Reconfigurable Motion Planning and Control in Obstacle Cluttered Environments under Timed Temporal Tasks

Christos K. Verginis[1], Constantinos Vrohidis[2], Charalampos P. Bechlioulis[2], Kostas J. Kyriakopoulos[2], and Dimos V. Dimarogonas[1]

***Abstract*— This work addresses the problem of robot navigation under timed temporal specifications in workspaces cluttered with obstacles. We propose a hybrid control strategy that guarantees the accomplishment of a high-level specification expressed as a timed temporal logic formula, while preserving safety (*i.e.,* obstacle avoidance) of the system. In particular, we utilize a motion controller that achieves safe navigation inside the workspace in predetermined time, thus allowing us to abstract the motion of the agent as a finite timed transition system among certain regions of interest. Next, we employ standard formal verification and convex optimization techniques to derive high-level timed plans that satisfy the agent's specifications. A simulation study illustrates and clarifies the proposed scheme.**

I. Introduction

Temporal-logic-based motion planning has gained significant attention in recent years, as it provides a fully automated correct-by-design control synthesis approach for autonomous robots. Temporal logics such as linear or metric temporal logic (LTL, MTL) provide formal high-level languages that can describe planning objectives more complex than the well-studied point-to-point navigation ("never enter a dangerous regions", "keep visiting regions A and B infinitely often") [1]–[7]. Firstly, the task specification is expressed as a temporal logic formula over a discrete representation of the system (*e.g.,* a transition system), which is obtained through the procedure of abstraction (*e.g.,* [2], [8]–[12]). Then a high-level discrete plan is found by off-the-shelf model-checking algorithms [13], which is executed by low-level controllers.

Ultimately, however, we are interested in complex tasks over predefined time horizons ("collect data in region C every 50 seconds and upload it in region D after at most 10 seconds"), which can be encompassed by timed temporal languages (*e.g.,* Metric Temporal Logic) [14]. Timed temporal tasks have been considered in a variety of works in the related literature [12], [15]–[23]. The authors in [21], [22] focus on multi-agent planning of global and local tasks expressed as Metric Interval Temporal Logic (MITL) formulas, based on reachibility analysis performed on the discretized state space. The authors of [20], [24] resort to facet reachability analysis under timed bounds by discretizing the space of available controls. The vehicle-routing problem under Metric Temporal Logic (MTL) specifications is tackled in [17] and [15] via Mixed-Integer Linear Programms (MILP), and tree-search/stochastic dynamic-programming (SDP) algorithms, respectively, without employing standard automata-based techniques. In [19] the authors use predetermined open-loop controllers to transition in a partitioned workspace according to a high-level plan satisfying an MITL formula determined through formal verification techniques. MITL specifications are also considered in [12], [23], where robust closed-form controllers that guarantee the timed navigation of a cooperatively manipulated object in a partitioned workspace are derived. [18] proposes a roadmap abstraction-based algorithm to address the problem of multi-goal motion planning under time windows.

Most of the related works on temporal logic-based motion planning and control resort to full workspace partitioning in order to encode safety specifications (*e.g.,* obstacle avoidance) and to facilitate the synthesis of controllers that implement the transitions of the abstracted discrete system. Regarding timed temporal tasks, many related works (*e.g.,* [20], [24]) use optimization techniques that yield maximum velocity controllers, to obtain upper bounds on the transition times between the states of the abstracted system. This approach is, in general, conservative and might lead to unnecessarily high control effort. Works that focus solely on task assignment neglect the continuous dynamics entirely [15], [17], [21].

In this work, we consider the robot motion planning under timed temporal tasks. Firstly, given predefined regions of interest[1] in an obstacle-cluttered workspace, we employ our previous results on feedback closed-loop[2] navigation [25] to design robot transitions among the regions under strict time constraints. This allows us to abstract the motion of the robot as a timed transition system over the regions of interest, without requiring any further workspace partition refinement. Subsequently, we employ formal verification techniques to

[1]KTH Center of Autonomous Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden. Email: {cverginis, dimos}@kth.se. This work was supported by the H2020 ERC Starting Grant BUCOPHSYS, the European Union's Horizon 2020 Research and Innovation Programme under the GA No. 731869 (Co4Robots), the Swedish Research Council (VR), the Knut och Alice Wallenberg Foundation (KAW) and the Swedish Foundation for Strategic Research (SSF)

[2] Control Systems Laboratory, School of Mechanical Engineering, National Technical University of Athens, 9 Heroon Polytechniou Street, Zografou 15780, Greece. Electronic addresses: {vroh, chmpechl, kkyria}@mail.ntua.gr.

[1]Regions of interest are also employed in [7] where solely untimed specifications are considered.

[2]The closed loop nature of the feedback control renders the scheme robust against modeling uncertainties/external disturbances, compared to open loop controllers (*e.g.,* [19])

derive a plan that satisfies the untimed specification and recast the assignment of the transition times as a convex optimization problem thereby achieving satisfaction of the timed specification. The transition times are recalculated after each transition, incorporating newly acquired information, and resulting in decreased control effort.

II. Preliminaries

In this section, we summarize some preliminary notions that are used in the sequel.

A. Timed Logics

Definition 1 [26] A time sequence $t_0 t_1 t_2 \dots$ is an infinite sequence of time values $t_j \in \mathbb{R}_{\geq 0}$, $j \in \mathbb{N}_0$, satisfying $t_{j+1} = t_j + t_{j,j+1}$, for some constants $t_{j,j+1} \in \mathbb{R}_{\geq 0}$.

An *atomic proposition* is a statement over the problem variables and parameters that is either True ($\top$) of False ($\bot$) at a given time instance.

Definition 2 Let $\mathcal{AP}$ be a finite set of atomic propositions. A timed word w over $\mathcal{AP}$ is an infinite sequence $w = (w_0, t_0)(w_1, t_1), \dots$, where $w_0 w_1 w_2 \dots$ is an infinite word over $2^{\mathcal{AP}}$ and $t_0 t_1 t_2 \dots$ is a time sequence according to Def. 1.

Definition 3 A *Weighted Transition System (WTS)* is a tuple $\mathcal{T} \coloneqq (\Pi, \Pi_0, \longrightarrow, \mathcal{AP}, \mathcal{L}, \gamma)$,, where Π is a finite set of states, $\Pi_0 \subseteq \Pi$ is a set of initial states, $\longrightarrow \subseteq \Pi \times \Pi$ is a transition relation, $\mathcal{AP}$ is a finite set of atomic propositions, $\mathcal{L} : \Pi \to 2^{\mathcal{AP}}$ is a labeling function, and $\gamma : \longrightarrow \to \mathbb{R}_{\geq 0}$ is a map that assigns a weight to each transition.

For convenience, we use $\pi \to \pi'$ to denote the fact that $(\pi, \pi') \in \longrightarrow$.

Definition 4 A *timed run* of a WTS is an infinite sequence $\mathsf{R} \coloneqq (\pi_0, t_0)(\pi_1, t_1)(\pi_2, t_2) \dots$ such that $\pi_0 \in \Pi_0$, $\pi_j \in \Pi$, and $\pi_j \to \pi_{j+1}$, for all $j \in \mathbb{N}_0$. The sequence of the time stamps $t_0 t_1 \dots$ is a time sequence according to Def. 1. The timed run r generates a *timed word* $\mathsf{w}(\mathsf{R}) \coloneqq \mathsf{w}_0(\pi_0)\mathsf{w}_1(\pi_1)\mathsf{w}_2(\pi_2) \dots \coloneqq (\mathcal{L}(\pi_0), t_0)(\mathcal{L}(\pi_1), t_1)(\mathcal{L}(\pi_2), t_2) \dots$ over the set $2^{\mathcal{AP}}$, where for each $j \in \mathbb{N}_0$, $\mathcal{L}(\pi_j)$ is the subset of atomic propositions that are true at state π_j at time t_j.

The *syntax of timed logics* over a set of atomic propositions $\mathcal{AP}$ is defined by a grammar that has the form

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc_I \varphi \mid \Diamond_I \varphi \mid \Box_I \varphi \mid \varphi_1 \mathcal{U}_I \varphi_2, \quad (1)$$

where $p \in \mathcal{AP}$, and $\bigcirc, \Diamond, \Box$ and $\mathcal{U}$ are the next, future, always and until operators, respectively; I is a nonempty time interval in one of the following forms: $[i_1, i_2], [i_1, i_2), (i_1, i_2], (i_1, i_2), [i_1, \infty), (i_1, \infty)$ with $i_1, i_2 \in \mathbb{Q}_{\geq 0}, i_2 > i_1$. Several languages have the form (1), such as Metric Temporal Logic (MTL) and Metric Interval Temporal Logic (MITL). Other variants also exist, like Bounded-MTL (where I needs to be bounded), or coFlat-MTL. More details can be found in [27]. In the following, we define the generalized semantics of (1), interpreted in point-wise semantics, *i.e.,* over discrete observations.

Definition 5 [28], [29] Given a sequence $\mathsf{R} = (\pi_0, t_0)(\pi_1, t_1)(\pi_2, t_2) \dots$ and a timed formula φ, we define $(\mathsf{R}, j) \models \varphi$, $j \in \mathbb{N}_0$ (R satisfies φ at j) as follows:

$$\begin{aligned}
&(\mathsf{R}, j) \models p \Leftrightarrow p \in \mathcal{L}(\pi_j),\\
&(\mathsf{R}, j) \models \neg\varphi \Leftrightarrow (\mathsf{R}, j) \not\models \varphi,\\
&(\mathsf{R}, j) \models \varphi_1 \wedge \varphi_2 \Leftrightarrow (\mathsf{R}, j) \models \varphi_1 \text{ and } (\mathsf{R}, j) \models \varphi_2,\\
&(\mathsf{R}, j) \models \textstyle\bigcirc_I \varphi \Leftrightarrow (\mathsf{R}, j+1) \models \varphi \text{ and } t_{j+1} - t_j \in I,\\
&(\mathsf{R}, j) \models \varphi_1 \mathcal{U}_I \varphi_2 \Leftrightarrow \exists k \geq j \text{ such that } (\mathsf{R}, k) \models \varphi_2,\\
&\qquad\qquad t_k - t_j \in I \text{ and } (\mathsf{R}, m) \models \varphi_1, \forall m \in \{j, \dots, k\}.
\end{aligned}$$

Also, $\Diamond_I \varphi = \top \mathcal{U}_I \varphi$ and $\Box_I \varphi = \neg\Diamond_I \neg\varphi$. The sequence R satisfies φ, denoted as $\mathsf{R} \models \varphi$, if and only if $(\mathsf{R}, 0) \models \varphi$.

B. Timed Büchi Automata

We provide here a description of *Timed Büchi Automata (TBA)*, originally proposed in [26]. Let $\mathsf{CL} \coloneqq \{\mathsf{cl}_1, \dots, \mathsf{cl}_{|\mathsf{CL}|}\}$ be a finite set of *clocks*. The set of *clock constraints* $\Phi(\mathsf{CL})$ is defined by the grammar:

$$\phi ::= \top \mid \neg\phi \mid \phi_1 \ \wedge \ \phi_2 \mid \mathsf{cl} \bowtie \psi,$$

where $\mathsf{cl} \in \mathsf{CL}$ is a clock, $\psi \in \mathbb{Q}$ is a clock constraint, and $\bowtie \in \{<, >, \geq, \leq, =\}$. A *clock valuation* is a mapping $v : \mathsf{CL} \to \mathbb{R}$ that assigns a value to each clock. A clock cl_i has valuation v_i for $i \in \{1, \dots, |\mathsf{CL}|\}$. Given $v \coloneqq (v_1, \dots, v_{|\mathsf{CL}|})$ and $t \in \mathbb{R}_{\geq 0}$, we denote by $v \models \phi$ and $t \models \phi$ the fact that the valuation v and the time instant t, respectively, satisfy the clock constraint ϕ.

Definition 6 A *Timed Büchi Automaton* is a tuple $\mathcal{A}_t \coloneqq (Q, Q_0, \mathsf{CL}, \mathcal{AP}, E, F)$, where Q is a finite set of locations, $Q_0 \subseteq Q$ is the set of initial locations, CL is a finite set of clocks, $\mathcal{AP}$ is a finite set of atomic propositions that defines the input alphabet $2^{\mathcal{AP}}$, $E \subset Q \times \Phi(\mathsf{CL}) \times 2^{\mathsf{CL}} \times 2^{\mathcal{AP}} \times Q$ gives the set of edges of the form $e = (q, g, R, \alpha, q')$, where q, q' are the source and target locations, g is the guard of edge, R is a set of clocks to be reset upon executing the edge, and α is an input string; finally, $F \subseteq Q$ is a set of accepting locations.

A state of $\mathcal{A}_t$ is a pair $(q, v) \in \mathbb{R} \times \mathbb{R}^{|\mathsf{CL}|}$. The initial state of $\mathcal{A}_t$ is $(q_0, 0_{|\mathsf{CL}|})$, with $q_0 \in Q_0$. Given two states (q, v), (q', v'), and an edge $e = (q, g, R, \alpha, q')$, there exists a *discrete transition* $(q, v) \xrightarrow{e} (q', v')$ if $v \models g$. Moreover, $v'_i = 0$, $\forall \mathsf{cl}_i \in R$, and $v'_i = v_i$, $\forall \mathsf{cl}_i \notin R$. Given $\delta \in \mathbb{R}$, there exists a *time transition* $(q, v) \xrightarrow{\delta} (q', v')$ if $q = q'$ and $v' = v + \delta$ (component-wise summation). We write $(q, v) \xrightarrow{\delta} \xrightarrow{e} (q', v')$ if there exists q'', v'' such that $(q, v) \xrightarrow{\delta} (q'', v'')$ and $(q'', v'') \xrightarrow{e} (q', v')$, with $q'' = q$.

An infinite run of $\mathcal{A}_t$ starting at a state (q_0, v_0) is an infinite sequence of time and discrete transitions $(q_0, v_0) \xrightarrow{\delta_0}$

$(q_0', v_0') \xrightarrow{e_0} (q_1, v_1) \xrightarrow{\delta_1} (q_1', v_1') \ldots$, where $e_i = (q_i, g_i, R_i, \sigma_i, q_i')$, $\forall i \in \mathbb{N}_0$. This run corresponds to the timed word $\mathsf{w}_t = (\sigma_0, \tau_0)(\sigma_1, \tau_1)$, with $\tau_{i+1} = \tau_i + \delta_i$, $\forall i \in \mathbb{N}_0$. The run is called accepting if $q_j \in F$ for infinitely many $j \in \mathbb{N}_0$. A timed word is called *accepting* if there exists an accepting run associated with it. The problem of deciding the language emptiness of a given TBA is PSPACE-complete [26]. In other words, an accepting run of a given TBA can be synthesized, if one exists. Any timed formula φ over $\mathcal{AP}$ originating from the decidable fragment of timed logics (*e.g.,* MITL, discrete-time MTL, finite MTL[3], coFlat-MTL, Bounded-MTL [27], [29]) can be algorithmically translated into a TBA with input alphabet $2^{\mathcal{AP}}$, such that the language of timed words that satisfy φ is the language of timed words produced by the TBA. Finally, given a TBA $\mathcal{A}_t$ with clock constraints of the form $\mathsf{cl} \bowtie \psi$, $\psi \in \mathbb{Q}_{\geq 0}$, we denote by $\mathcal{A}_{t+\tau}$, with $\tau \in \mathbb{Q}_{\geq 0}$, the automaton whose clock constraints are shifted to the left by τ, *i.e.,* they are of the form $\mathsf{cl} \bowtie (\psi - \tau)$, $\bowtie \in \{<, >, \geq, \leq, =\}$.

III. Problem Formulation

Consider a robotic agent operating in an open bounded subset $\mathcal{W}$ of the 2-dimensional Euclidean space. In addition, the workspace is populated with $m \in \mathbb{N}$ connected, closed sets $\{O_i\}_{i \in \mathcal{J}}$, indexed by the set $\mathcal{J} \coloneqq \{1, \ldots, m\}$, corresponding to obstacles. Accordingly, we define the free space as

$$\mathcal{F} \coloneqq \mathcal{W} \backslash \bigcup_{i \in \mathcal{J}} O_i,$$

Remark 1. To facilitate exposition, we assume that all the data describing the workspace are known *a priori*. The analysis remains the same for the case of initially unknown workspaces where obstacles are discovered along the way.

The agent is assumed to be a point[4] described by the position variable $x \in \mathbb{R}^2$ which is governed by the single integrator dynamics,

$$\dot{x} = u, \quad u \in \mathbb{R}^2. \tag{2}$$

Moreover, we consider that there exist K points of interest in the free space, denoted by $c_k \in \mathcal{F}$, for every $k \in \mathcal{K} \coloneqq \{1, \ldots, K\}$, with $\Pi \coloneqq \{c_1, \ldots, c_K\}$, that correspond to satisfy certain properties of interest (*e.g.,* gas station, obstacle region, repairing area, etc.) These properties of interest are expressed as boolean variables via the finite set of atomic propositions $\mathcal{AP}$. The properties satisfied at each point are provided by the labeling function $\mathcal{L} : \Pi \to 2^{\mathcal{AP}}$. Informally, $\mathcal{L}$ assigns to each point $c_k, k \in \mathcal{K}$, the subset of the atomic propositions that hold true in that point.

Since, in practice, the aforementioned properties shared by a point of interest are naturally inherited to some neighborhood of that point we define for each $k \in \mathcal{K}$, the *region of interest* π_k corresponding to the point of interest c_k as the set

$$\pi_k \coloneqq \bar{\mathcal{B}}(c_k, r_{\pi_k}) \cap \mathcal{F}, \quad r_k \in \mathbb{R}_{>0},$$

where $\bar{\mathcal{B}}(c, r)$ denotes the closed ball of radius $r > 0$ centered at $c \in \mathbb{R}^2$. We also let $\pi_{\mathcal{W}} \coloneqq \mathcal{F} \backslash (\cup_{k \in \mathcal{K}} \pi_k)$ be the subset of the free space outside the regions of interest. We define thus the set $\widetilde{\Pi} \coloneqq \{\pi_k\}_{k \in \mathcal{K}} \cup \{\pi_{\mathcal{W}}\}$ as well as the corresponding labeling function as $\widetilde{\mathcal{L}} : \widetilde{\Pi} \to 2^{\mathcal{AP}}$, with $\mathcal{L}(c_k) = \{p\} \Leftrightarrow \widetilde{\mathcal{L}}(\pi_k) = \{p\}$, $\forall k \in \mathcal{K}$, and $\widetilde{\mathcal{L}}(\pi_{\mathcal{W}}) = \emptyset$. The agent is assumed to be in a region π_k, $k \in \mathcal{K}$, in $\pi_{\mathcal{W}}$, simply when $x \in \pi_k$ and $x \in \pi_{\mathcal{W}}$, respectively. We assume that, for all $k \in \mathcal{K}$, the location of the points c_k as well as the radii r_{π_k} are known.

We make the following standard assumptions [30] regarding the geometry of the workspace and the regions of interest.

Assumption 1 *The collection of sets comprised of all obstacles and regions of interest is pairwise disjoint.*

The aforementioned assumption simply states that the obstacles/regions of interest are sufficiently away from each other as well as the workspace boundary.

As already mentioned, we are interested in defining timed temporal formulas over the atomic propositions $\mathcal{AP}$, and hence, over the regions of interest Π of $\mathcal{F}$. To that end, we need to discretize the system using a finite set of states. We will achieve that by guaranteeing timed transitions between the regions of interest in Π and by building a well-defined timed transition system among them. We first need the following definition regarding the transitions of the agent.

Definition 7 Assume that $x(t_k) \in \mathcal{F}$, for a $t_k \in \mathbb{R}_{\geq 0}$, *i.e.,* the agent is either in a region π_k, for some $k \in \mathcal{K}$, or in $\pi_{\mathcal{W}}$. Then, given $\delta \in \mathbb{R}_{>0}$, there exists a *timed transition* to π_ℓ, $\ell \in \mathcal{K}$, denoted as $\pi_k \to \pi_\ell$ (or $\pi_{\mathcal{W}} \to \pi_\ell$), if there exists a time-varying feedback control law $u : \mathcal{F} \times [t_k, t_\ell] \to \mathbb{R}^2$, with $t_\ell \geq t_k + \delta$, such that the solution x of the closed loop system (2) satisfies the following:

(i) $x(t) \in \pi_l$, for all $t \in [t_k + \delta, t_l)$,

(ii) $x(t) \in \mathcal{F}$, for all $t \in [t_k, t_\ell]$,

(iii) $x(t) \notin \pi_m$, for all $m \in \mathcal{M},\ t \in [t_k, t_\ell]$,

where $\mathcal{M} \coloneqq \mathcal{K} \backslash \{k, \ell\}$ if $x(t_k) \in \pi_k$ and $\mathcal{M} \coloneqq \mathcal{K} \backslash \{\ell\}$ if $x(t_k) \in \pi_{\mathcal{W}}$.

Intuitively, according to Def. 7, the agent has to transit between two regions π_k, π_ℓ (or $\pi_{\mathcal{W}}$ and π_ℓ), while avoiding all other regions of interest, obstacles, as well as the workspace boundary. In what follows, we sometimes use $\pi_k \xrightarrow{\delta} \pi_\ell$ instead of $\pi_k \to \pi_\ell$ to emphasize the transition time δ. We have included the space outside the regions $\pi_{\mathcal{W}}$ to account for initial conditions that might satisfy $x(t_k) \notin \cup_{k \in \mathcal{K}} \pi_k$. Next, we define the behavior of the agent, in order to formulate the problem of timed specifications.

Definition 8 Consider an agent trajectory $x : [t_0, \infty) \to \mathcal{F}$

[3]In this case, the generated Timed Automaton will have finite accepting runs.

[4]Treating a robot with volume can be achieved by initially "transferring" its volume to the other workspace entities (*e.g.,* obstacles) and subsequently considering it as a point.

of (2), where $t_0 \in \mathbb{R}_{\geq 0}$. Then, a *timed behavior* of x is the infinite sequence $\mathsf{b} \coloneqq (x(t_0), \sigma_0, t_0)(x(t_1), \sigma_1, t_1)\ldots$, where $t_0 t_1 \ldots$ is a time sequence according to Def. 1, $x(t_0) \in \widetilde{\Pi}$, $x(t_l) \in \pi_{j_l}$, $j_l \in \mathcal{K}$, for $l \in \mathbb{N}_0$, and $\sigma_l = \mathcal{L}(\pi_{j_l}) \subseteq 2^{\mathcal{AP}}$, *i.e.,* the subset of atomic propositions that are true when $x(t_j) \in \pi_{j_l}$, for $l \in \mathbb{N}_0$. The timed behavior b satisfies a timed formula φ if and only if $\mathsf{b}_\sigma \coloneqq (\sigma_0, t_0)(\sigma_1, t_1)\ldots \models \varphi$.

We are now ready to state the problem addressed in the present work.

Problem 1 *Consider a robot with dynamics governed by* (2)*, operating in the workspace $\mathcal{W}$, with initial position $x(0) \in \mathcal{F}$. Given a timed formula φ over $\mathcal{AP}$ and a labeling function $\widetilde{\mathcal{L}}$, develop a control strategy that results in a solution $x : [0, \infty) \to \mathcal{F}$, which achieves a timed behavior b that yields the satisfaction of φ.*

IV. Methodology

In this section we present the proposed solution, which consists of two layers: (i) a tuning-free continuous control law that guarantees the navigation of the agent to a desired point from *all* obstacle-collision-free configurations, and (ii) a discrete time plan over the regions of interest for the robot to follow, which employs formal verification and optimization techniques and is updated on-line.

A. Motion Controller

The first part of the proposed solution is the design of a control protocol such that a transition to a region of interest is established, according to Def. 7. Assume, therefore, that $x(t_k) \in \mathcal{F}$, and more specifically, $x(t_k) \in \pi_k$ ($x(t_k) \in \pi_{\mathcal{W}}$) for some $t_k \in \mathbb{R}_{\geq 0}$ and $k \in \mathcal{K}$. Given $\delta \in \mathbb{R}_{>0}$, we wish to find a time-varying state-feedback control law $u : \mathcal{F} \times [t_k, t_\ell]$, with $t_\ell \geq t_k + \delta$, such that $\pi_k \xrightarrow{\delta} \pi_\ell$ ($\pi_{\mathcal{W}} \xrightarrow{\delta} \pi_\ell$). To that end, we first redefine the free space as

$$\mathcal{F} \coloneqq \mathcal{W} \backslash \left(\bigcup_{i \in \mathcal{J}} O_i \cup \bigcup_{l \in \mathcal{M}} \pi_m \right),$$

so that regions of interest that shall not be crossed during the transition are regarded as obstacles.

Following our previous work [25], the tuple $(x(t_k), c_l, \rho_{\pi_l}, \delta)$ constitute a well-defined instance of the *Prescribed Time Scale Navigation Problem* [25, Problem 1] in $\mathcal{F}$. Theorem 2 of the aforementioned work suggests that the construction of the required feedback law $u : \mathcal{F} \times [t_k, t_\ell] \to \mathbb{R}^2$ reduces to the problem of smoothly transforming the free space $\mathcal{F}$ to a topologically equivalent, yet geometrically simpler, space.

More specifically, we require a diffeomorphism $\mathrm{T} : \mathcal{F} \to \mathcal{P}$ where $\mathcal{P}$ is a point world [31]; an open disk modulo a finite set with cardinality equal to the number of obstacles and regions of interest $|\mathcal{J}| + |\mathcal{M}|$. Under the prevailing Assumption 1, [32, Theorem 1] provides a computationally efficient method to determine the space $\mathcal{P}$ and the mapping T.

This allows us to apply the conclusions of [25, Theorems 1, 2] which yield a feedback law $u : \mathcal{F} \times [t_k, t_\ell] \to \mathbb{R}^2$ such that the closed-loop system satisfies the properties of Def. 7 therefore establishing the existence of the required timed transition. Presenting the analytical expression for the resulting feedback law is deemed prohibitive due to space limitations. However, we encourage the reader to consult [25] for a detailed exposition.

B. High-Level Plan Generation

The second part of our solution is the derivation of a high-level timed plan over the regions of interest, which satisfies the given timed formula φ. This plan will be generated using standard techniques from automata-based formal verification and optimization methodologies. Thanks to the proposed control law of the previous section that allows the transitions in the set $\widetilde{\Pi}$ in predefined time intervals, we can abstract the motion of the robotic agent as a finite transition system $\mathcal{T} \coloneqq \{\widetilde{\Pi}, \widetilde{\Pi}_0, \longrightarrow, \mathcal{AP}, \widetilde{\mathcal{L}}, \gamma\}$, where $\widetilde{\Pi}$ is the set of states defined in Section III, $\widetilde{\Pi}_0 \in \widetilde{\Pi}$ is the initial state, $\longrightarrow \coloneqq \widetilde{\Pi} \times \widetilde{\Pi}$ is a transition relation according to Def. 7, $\mathcal{AP}$ and $\widetilde{\mathcal{L}}$ are the atomic propositions and the labeling function, respectively, as defined in Section III, and $\gamma : (\longrightarrow) \to \mathbb{R}_{>0}$ is a cost associated with each transition. More specifically, we consider as cost the distance the agent has to cover from a region π_k (or $\pi_{\mathcal{W}}$) to a region π_ℓ. However, this cost is highly dependent on the initial robot configuration and the number and position of the obstacles between the initial and the goal regions, and cannot be computed explicitly. Therefore, we initially set $\gamma(\pi_k \to \pi_\ell) = \|c_k - c_\ell\|$, $\gamma(\pi_k \to \pi_k) = 0$, and $\gamma(\pi_{\mathcal{W}} \to \pi_k) = \gamma(\pi_k \to \pi_{\mathcal{W}}) = \|c_k - x(0)\|$, for all $k, \ell \in \mathcal{K}$ with $k \neq \ell$, and proceed with the derivation of the timed plan as a timed sequence of regions in Π.

Firstly, the timed formula φ over the atomic propositions $\mathcal{AP}$ is translated to the TBA $\mathcal{A}_t = (Q, Q_0, \mathsf{CL}, \mathcal{AP}, E, F)$ using off-the-shelf tools [33]. Secondly, we calculate the product Büchi Automaton $\mathcal{A}_\mathcal{P}$ as $\mathcal{A}_\mathcal{P} \coloneqq \mathcal{T} \otimes \mathcal{A}_t = (S, S_0, \longrightarrow_\mathcal{P}, F_\mathcal{P}, \gamma_\mathcal{P})$, where

- $S = \widetilde{\Pi} \times Q$,
- $S_0 = \widetilde{\Pi} \times Q_0$,
- $\longrightarrow_\mathcal{P} \subset S \times \Phi(\mathsf{CL}) \times 2^C \times S$ gives the set of edges; $e \coloneqq (s, g, R, s') \in \longrightarrow_\mathcal{P}$, with $s \coloneqq (\pi, q)$, $s' \coloneqq (\pi', q') \in S$ if and only if (i) $(q, g, R, \mathcal{L}(\pi), q') \in E$ and (ii) $q = q'$, $(\pi, \pi') \in \longrightarrow$, with $\pi \neq \pi'$, *or* $\pi = \pi'$.
- $F_\mathcal{P} \subseteq \widetilde{\Pi} \times F$ with $s \coloneqq (\pi, q) \in F_\mathcal{P}$ if and only if $q \in F$ and $(s, \Phi(\mathsf{CL}), R, S) \in E$ for some state in S, *i.e.,* there is always a transition from s, for all the the possible valuations of the clocks CL.
- $\gamma_\mathcal{P} : (\longrightarrow_\mathcal{P}^*) \to \mathbb{R}_{>0}$, with $\gamma_\mathcal{P}((s, g, R, s')) = \gamma(\pi \to \pi')$, where $(\longrightarrow_\mathcal{P}^*) \coloneqq \{((\pi, q), g, R, (\pi', q')) \in \longrightarrow_\mathcal{P} : \pi \neq \pi')\}$.

We use the abbreviation $s \xrightarrow{\mathcal{I}} s'$ for $(s, g, R, s') \in \longrightarrow_\mathcal{P}$,

where $\mathcal{I} \coloneqq \{g, R\}$. Note that the product $\mathcal{A}_\mathcal{P}$ consists of a finite number of states, and therefore we can employ graph-search techniques to find the optimal timed path, with respect to the cost $\gamma_\mathcal{P}$, from the initial states S_0 to the accepting states $F_\mathcal{P}$, which will satisfy the given timed formula φ [7]. This path will contain a finite prefix — a finite sequence of states to be visited — and a infinite suffix — a specific sequence of states to be visited infinitely many times [7], [13]. Moreover, note that the motion controller developed in Section Sec. IV-A can guarantee the safe navigation among two regions of interest in any predefined time interval.

By viewing $\mathcal{A}_\mathcal{P}$ as a graph, we can find a path that starts at the initial states S_0 and traverses an accepting state in $F_\mathcal{P}$ infinitely many times. Such a path has the form

$$\bar{s}_{p_0} \xrightarrow{I_{0,1}} \bar{s}_{p_1} \xrightarrow{I_{1,2}} \dots \xrightarrow{I_{L-1,L}} \bar{s}_{p_L} \xrightarrow{I_{L,L+1}} \left(\bar{s}_{p_{L+1}} \xrightarrow{I_{L+1,L+2}} \dots \xrightarrow{I_{L+Z-1,L+Z}} \bar{s}_{p_{L+Z}}\right)^\omega$$

Here, $\bar{s}_{p_j}$, for $j \in \{0, \dots, L+Z\}$, denotes the sequence of states

$$\bar{s}_{p_j} \coloneqq (\pi_{p_j}, q_{j_0}) \xrightarrow{I_{j_{0,1}}} \dots \xrightarrow{I_{j_{(\ell_j-1),\ell_j}}} (\pi_{p_j}, q_{j_{\ell_j}}),$$

with $\pi_{p_j} \in \widetilde{\Pi}$, $q_{j_\iota} \in Q$, for $j \in \{0, \dots, L+Z\}$, $\iota \in \{1, \dots, \ell_j\}$, and $\ell_j \in \{0, \dots, |S|\}$. Moreover, $q_{(j+1)_0} = q_{j_{\ell_j}}$, $q_{(L+Z)_{\ell_{(L+Z)}}} = q_{(L+1)_0}$ and

$$I_{j,j+1} \coloneqq \Big\{g_{j,j+1}, R_{j,j+1}\Big\}, \quad I_{j_{\iota,\iota+1}} \coloneqq \Big\{g_{j_{\iota,\iota+1}}, R_{j_{\iota,\iota+1}}\Big\},$$

indicating the corresponding guards and reset maps, for $j \in \{0, 1, \dots, L+Z-1\}, \iota \in \{0, \dots, \ell_j - 1\}$. The transition set $I_{L+Z,L+1}$ is defined similarly. Loosely speaking, the path consists of consecutive (at most $|S|$) transitions of the form $(\pi_j, q_{j_\iota}) \xrightarrow{(\cdot)} (\pi_j, q_{j_{(\iota+1)}})$ among states in $\mathcal{A}_t$, where π_j is fixed, and transitions of the form $(\pi_{p_j}, q_{j_{\ell_j}}) \xrightarrow{(\cdot)} (\pi_{p_{(j+1)}}, q_{(j+1)_0})$ among the states of $\mathcal{T}$, where $q_{(j+1)_0} = q_{j_{\ell_j}}$ is fixed.

Note that we have not yet associated any time intervals with the transitions $(\pi_{p_j}, q_{j_{\ell_j}}) \xrightarrow{(\cdot)} (\pi_{p_{(j+1)}}, q_{(j+1)_0})$, which correspond to physical transitions among the regions of interest. We do that now by using the transition guards $g_{j,j+1}, g_{j_{\iota,\iota+1}}$. More specifically, consider the transitions

$$(\pi_{p_j}, q_{j_0}) \xrightarrow{I_{j_{0,1}}} \dots \xrightarrow{I_{j_{(\ell_j-1),\ell_j}}} (\pi_{p_j}, q_{j_{\ell_j}}) \xrightarrow{I_{j,j+1}} (\pi_{p_{j+1}}, q_{(j+1)_0}),$$

that encode the physical transition from π_{p_j} to $\pi_{p_{j+1}}$ in $\mathcal{A}_\mathcal{P}$. The intersection of the respective guards $g_{j,j+1}, g_{j_{\iota,\iota+1}}$, $\iota \in \{0, \dots, \ell_j - 1\}$, provides a time interval of the form $\mathcal{I}_{j,j+1} \in \{[a,b], [a,b), (a,b], (a,b), [a,\infty), (a,\infty)\}$, with $a, b \in \mathbb{Q}_{>0}$, $b > a$, such that, $t_{j,j+1} \in \mathcal{I}_{j,j+1} \Rightarrow t_{j,j+1} \models g_{j,j+1}, t_{j,j+1} \models g_{j_{\iota,\iota+1}}$, for $\iota \in \{0, \dots, \ell_j - 1\}$, where $t_{j,j+1}$ is the time duration of the navigation $\pi_j \xrightarrow{t_{j,j+1}} \pi_{j+1}$. Note that $\mathcal{I}^i_{j,j+1}$ might be a function of the previous transition duration $t_{j-1,j}$.

Since $\mathcal{I}^i_{j,j+1}$ is, in general, an infinite set, and we have, thus, infinitely many choices for $t_{j,j+1}$, we propose a procedure for assigning the time durations $t_{j,j+1}$, for each $j \in \{0, L+Z, -1\}$, and $t_{L+Z,L+1}$. In particular, we formulate the transition times assignment as a convex optimization problem. To that end, let $t_p \coloneqq [t_{0,1}, \dots, t_{L+Z-1,L+Z}, t_{L+Z,L+1}]^\top \in \mathbb{R}^{L+Z+1}_{>0}$ be the concatenation of the transition times constituting the variable of the following optimization problem:

$$\begin{aligned} \underset{t_p}{\text{minimize}} \; & \sum_{j=0}^{L+Z-1} \left(\frac{\gamma(\pi_{p_j} \to \pi_{p_{j+1}})}{t_{j,j+1}}\right) + \frac{\gamma(\pi_{p_{L+Z}} \to \pi_{p_{L+1}})}{t_{j,j+1}}, \\ \text{subject to } & t_{j,j+1} \in \mathcal{I}_{j,j+1}, \text{ for all } j \in \{0, L+Z, -1\}, \\ & t_{L+Z,L+1} \in \mathcal{I}_{L+Z,L+1}. \end{aligned} \tag{3}$$

Note that the objective function is a convex function of t_p and the constraints can be expressed as linear inequalities on the problem variables. Thus, the above optimization problem is convex and can be efficiently solved using off-the-shelf software. The choice of this particular cost function is motivated by the following two observations: (i) the time assigned to a transition is an increasing function of the transition cost, and (ii) brief transition times are penalized. Furthermore, the imposed constraints guarantee the satisfaction of the formula provided that transitions are executed within the specified transition times. We also report empirical evidence from numerical simulations suggesting that reduction in control effort is achieved.

After solving the aforementioned optimization problem and obtaining the time durations t_p the robot performs the first transition using the motion controller presented in Sec. IV-A where δ taken equal to the corresponding transition time. Once transition $\pi_{p_j} \xrightarrow{t_{j,j+1}} \pi_{p_{j+1}}$is completed, the corresponding transition cost $\gamma(\pi_j \to \pi_{j+1})$ is updated by being set equal to the length of the integral curve of the closed-loop system for the duration of the transition. The updated value of the transition cost is in some sense more accurate than the initial estimate based on the Euclidean distance since the existence of obstacles can potentially obstruct the straight line path between two regions of interest.

Having acquired this new information the associated optimization problem can be solved to acquire new values for the transition times. We note that after each transition the constraints of the optimization problem are altered. In particular, the TBA of the formula has to be shifted forward by an amount of time equal to the last performed transition which induces a change in the guards and, therefore, to the optimization problem constraints. We assume that the time needed for solving the optimization problem is short enough so that satisfaction of the formula is not jeopardized. This assumption is reasonable enough primarily owing to the problem's low computational complexity and, secondarily, the fact that the previously computed values of the transition times are a good prior for initiating the numerical solver. Nevertheless, computational overhead can be accounted for in the constraints by allocating the required time, or opti-

mization could be performed en route to the next region of interest with the transition times adjusted in an any-time fashion.

V. Simulation Results

To demonstrate the proposed scheme, we consider a task and motion planning problem for a robot operating in a planar office environment. In particular, we consider three points of interest and therefore have $\Pi = \{c_k\}_{k\in\mathcal{K}}$, where $\mathcal{K} = \{1, 2, 3\}$. The corresponding regions of interest $\pi_k = \bar{\mathcal{B}}(c_k, r)$, where $r_{\pi_k} = 0.2$ for $k \in \mathcal{K}$, define the set $\Pi = \{\pi_k\}_{k\in\mathcal{K}}$. The set of atomic preposition is $\mathcal{AP} = \tilde{\Pi}$ and the labeling function $\mathcal{L} : \Pi \to 2^{\mathcal{AP}}$ is defined as $\pi_k \mapsto \{\pi_k\}$, $k \in \mathcal{K}$. The scenario setting is illustrated in Fig. 1.

We require that the robot *"always visits each region of interest at least once every* 120 *time units"* which is equivalent to the MITL formula $\phi = \bigwedge_{k\in\mathcal{K}} \left(\square\Diamond_I \pi_k\right), \quad I = [0, 120]$. The robot is initially located at $c_1 \in \pi_1$ and, therefore, the infinitely repeating cycle of transitions $\pi_1 \xrightarrow{t_{1,2}(1)} \pi_2 \xrightarrow{t_{2,3}(1)} \pi_3 \xrightarrow{t_{3,1}(1)} \pi_1 \xrightarrow{t_{1,2}(2)} \dots$ with appropriately assigned transition times is an accepting run. Let $t : \mathbb{N}_0 \to \mathbb{Q}^3_{>0}$, $\kappa \mapsto [t_{1,2}(\kappa), t_{2,3}(\kappa), t_{3,1}(\kappa)]^T$ which is defined recursively as follows: $t(0) \coloneqq [0, 0, 0]^T$, then assuming $t(\kappa)$ is defined for some $\kappa \in \mathbb{N}_0$,

$$t(\kappa{+}1) \coloneqq (U_3^T)^\kappa \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U_3^\kappa \hat{t}(\kappa) + (U_3^T)^\kappa \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} U_3^\kappa t(\kappa),$$

where $U_3 \coloneqq \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$ is the upper shift matrix and $\hat{t}(\kappa)$ is the solution of optimization problem (3) under the following constraints:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} U_3^\kappa \hat{t}(\kappa) \leq \begin{bmatrix} 120 \\ 120 \\ 120 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} U_3^\kappa t(\kappa).$$

The motion controller results in collision-free trajectories (Fig. 1), and by performing each transition time in the time derived from the optimization procedure results in a run that satisfies the formula ϕ (see bottom of Fig. 2). Finally, it is worth noting that the transition times converge in just a few steps as illustrated on the top part of Fig. 2 and the overall control effort per suffix execution is reduced (Tbl. I).

TABLE I: Control Effort per Suffix Execution

Cycle	1	2	3	4	5
$\int \|u(x(\tau), \tau)\|^2 \, d\tau$	8.06	7.67	7.67	7.65	7.66

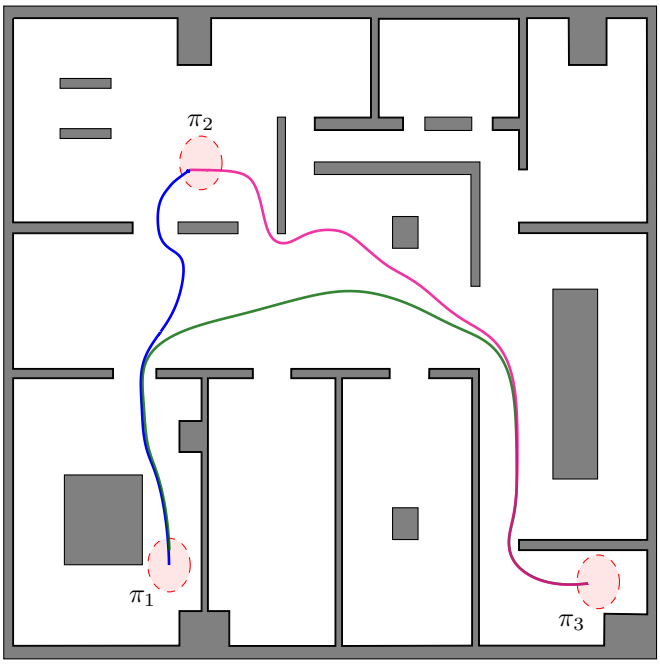

Fig. 1: Workspace overview. The red discs correspond to the three regions of interest. The plotted paths are the resulting trajectories from the first out of the five executions of the suffix.

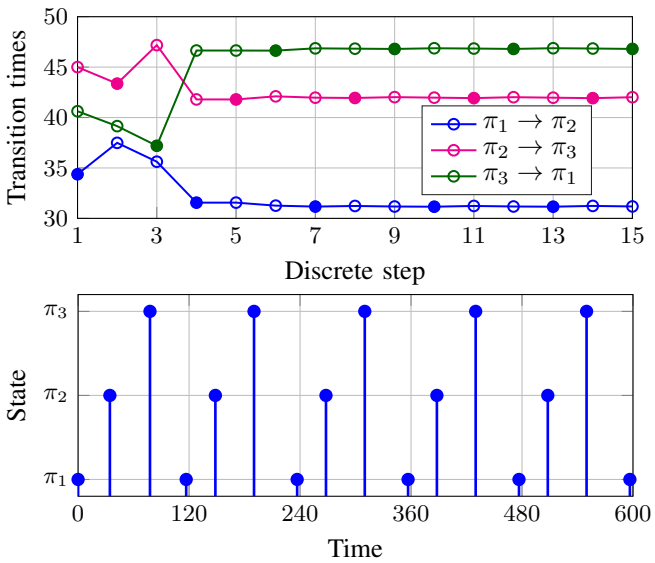

Fig. 2: (top) The transition times calculated before each transition; filled marks correspond to actually used transition times. (bottom) The resulting timed run of the transition system.

VI. Conclusion

In this work, we propose an integrated approach for addressing the motion planning problem for a mobile robot subject to timed temporal specifications operating in an environment with obstacles. We employ a motion controller that achieves safe, timed navigation between regions of interest in the workspace, allowing us to create a finite abstraction of the system. Furthermore, using standard techniques we derive a sequence of states that satisfies the untimed specification and use an iterating optimization-based approach to handle the assignment of transition times, guaranteeing satisfaction of the timed specification and reduction of the control effort.

REFERENCES

[1] A. Bhatia, L. E. Kavraki, and M. Y. Vardi, "Sampling-based motion planning with temporal goals," *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2689–2696, 2010.

[2] C. Belta, V. Isler, and G. J. Pappas, "Discrete abstractions for robot motion planning and control in polygonal environments," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 864–874, 2005.

[3] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.

[4] M. Kloetzer and C. Belta, "Automatic deployment of distributed teams of robots from temporal logic motion specifications," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 48–61, 2010.

[5] 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.

[6] S. G. Loizou and K. J. Kyriakopoulos, "Automatic synthesis of multi-agent motion tasks based on ltl specifications," *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 1, pp. 153–158, 2004.

[7] M. Guo, K. H. Johansson, and D. V. Dimarogonas, "Motion and action planning under ltl specifications using navigation functions and action description language," *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 240–245, 2013.

[8] G. Pola, A. Girard, and P. Tabuada, "Approximately bisimilar symbolic models for nonlinear control systems," *Automatica*, vol. 44, no. 10, pp. 2508–2516, 2008.

[9] D. Boskos and D. V. Dimarogonas, "Decentralized abstractions for feedback interconnected multi-agent systems," *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pp. 282–287, 2015.

[10] P.-J. Meyer, A. Girard, and E. Witrant, "Compositional abstraction and safety synthesis using overlapping symbolic models," *IEEE Transactions on Automatic Control*, vol. 63, no. 6, pp. 1835–1841, 2018.

[11] P. Tabuada, *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media, 2009.

[12] C. K. Verginis and D. V. Dimarogonas, "Timed abstractions for distributed cooperative manipulation," *Autonomous Robots*, vol. 42, no. 4, pp. 781–799, 2018.

[13] C. Baier, J.-P. Katoen, *et al.*, *Principles of model checking*. MIT press Cambridge, 2008.

[14] P. Bouyer, F. Laroussinie, N. Markey, J. Ouaknine, and J. Worrell, "Timed temporal logics," *Models, Algorithms, Logics and Tools*, pp. 211–230, 2017.

[15] M. Faied, A. Mostafa, and A. Girard, "Dynamic optimal control of multiple depot vehicle routing problem with metric temporal logic," *American Control Conference, 2009. ACC'09.*, pp. 3268–3273, 2009.

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

[17] S. Karaman and E. Frazzoli, "Vehicle routing problem with metric temporal logic specifications," *IEEE Conference on Decision and Control (CDC)*, pp. 3953–3958, 2008.

[18] S. Edelkamp, M. Lahijanian, D. Magazzeni, and E. Plaku, "Integrating temporal reasoning and sampling-based motion planning for multigoal problems with dynamics and time windows," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3473–3480, 2018.

[19] Y. Zhou, D. Maity, and J. S. Baras, "Timed automata approach for motion planning using metric interval temporal logic," *European Control Conference (ECC)*, pp. 690–695, 2016.

[20] S. Andersson, A. Nikou, and D. V. Dimarogonas, "Control synthesis for multi-agent systems under metric interval temporal logic specifications," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 2397–2402, 2017.

[21] A. Nikou, J. Tumova, and D. V. Dimarogonas, "Cooperative task planning of multi-agent systems under timed temporal specifications," *IEEE American Control Conference (ACC)*, pp. 7104–7109, July 2016.

[22] A. Nikou, D. Boskos, J. Tumova, and D. V. Dimarogonas, "On the timed temporal logic planning of coupled multi-agent systems," *Automatica*, vol. 97, pp. 339–345, 2018.

[23] C. K. Verginis and D. V. Dimarogonas, "Distributed cooperative manipulation under timed temporal specifications," *American Control Conference (ACC), 2017*, pp. 1358–1363, 2017.

[24] E. A. Gol and C. Belta, "Time-constrained temporal logic control of multi-affine systems," *Nonlinear Analysis: Hybrid Systems*, vol. 10, pp. 21–33, 2013.

[25] 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.

[26] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical computer science*, vol. 126, no. 2, pp. 183–235, 1994.

[27] P. Bouyer, N. Markey, J. Ouaknine, and J. Worrell, "The cost of punctuality," *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science (LICS'07)*, pp. 109–118, 2007.

[28] D. Souza and P. Prabhakar, "On the expressiveness of mtl in the pointwise and continuous semantics," *International Journal on Software Tools for Technology Transfer*, vol. 9, no. 1, pp. 1–4, 2007.

[29] J. Ouaknine and J. Worrell, "On the decidability of metric temporal logic," *Annual IEEE Symposium on Logic in Computer Science (LICS'05)*, pp. 188–197, 2005.

[30] 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.

[31] S. G. Loizou, "The navigation transformation," *IEEE Transactions on Robotics*, vol. PP, no. 99, pp. 1–8, 2017.

[32] P. Vlantis, C. Vrohidis, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Robot navigation in complex workspaces using harmonic maps," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1726–1731.

[33] T. Brihaye, G. Geeraerts, H.-M. Ho, and B. Monmege, "MightyL: A Compositional Translation from MITL to Timed Automata," *29th International Conference on Computer Aided Verification (CAV'17)*, vol. 10426, pp. 421–440, July 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01525524