Feasibility Envelopes for Metric Temporal Logic Specifications

Sadra Sadraddini and Calin Belta

Abstract— **Designing control policies from complex specifications has drawn significant attention in recent years. Metric temporal logic (MTL) is a specification formalism for describing a wide range of temporal properties with specific timing constraints. In this paper, we focus on discrete time linear control systems and specifications given as MTL formulas over linear predicates in the states. We present a method based on polyhedral projection to find the set of all initial states from which all trajectories satisfying MTL formulas can be generated. An illustrative example is included.**

I. Introduction

Formal methods were originally developed to provide languages and algorithms to specify and check the correctness of software and digital circuits. In recent years, formal methods have been increasingly used to design and verify control systems. The specifications are usually given as formulae in temporal logics such as Computation Tree Logic (CTL) and Linear Temporal Logic (LTL) [1], which allow for a rich spectrum of time-abstract requirements such as safety, reachability and sequentiality. Controlling a dynamical system from an LTL specification involves the construction of a finite-state abstraction of the system, followed by an automata-based control strategy [2], [3].

Metric temporal logic (MTL) is an extension of LTL, where the temporal operators are augmented with timing constraints [4], which makes it appealing for applications where time limits are important. As opposed to LTL, which has infinite-time semantics, MTL is interpreted over time-bounded signals. As a result, even though automata-based approaches for MTL control exist [5], they are not appropriate since one timed temporal operators are required to be translated into combination of "next" operators, resulting in very large automata [6]. A common approach to control synthesis from time bounded temporal logic specifications is formulating the control problem as a mixed integer linear programming (MILP) problem [7], [8]. In this context, checking whether a trajectory satisfying the specification can be generated from an initial condition requires solving an MILP problem. In many applications, it is important to characterize the set of all such initial conditions. However, computation of feasibility regions and parametric programming of MILPs is computationally expensive and is not an efficient approach.

In this paper, we focus on discrete-time linear systems and specifications given as MTL formulae over linear predicates in the state of the systems. Our goal is to find all the initial states from which control policies satisfying the MTL specifications are guaranteed to exist. By exploiting the time bounds of MTL, we characterize all the trajectories satisfying an MTL formula by polyhedral sets constructed in higher dimensions. We use the Fourier-Motzkin elimination method to project all the polyhedral sets, subject to the dynamical, state and input constraints, into the space of initial states. We also consider the problem of event responsive specifications, where an event triggers the requirement for the satisfaction of a MTL formula. Such specifications are common in engineering applications such as robotics, where online requests demand certain behaviors from the system. In this context, the set of all initial conditions explained earlier is viewed as the feasibility envelope of the MTL specification. We also explain how to optimally choose the controls from the feasibility region.

Feasibility envelopes are important in reachability and invariance analysis of control systems and have been extensively studied in the literature [9], [10], [11]. The problem of finding the set of initial conditions from which trajectories satisfying an LTL specification are generated is solved simultaneously with the control synthesis procedure in automata-based approaches. Environment responsive control strategies are often studied in the context of GR(1) (reactive LTL) formulas [12], where control strategies are synthesized with the consideration of events persistently occurring in the environment. However, since continuous systems are abstracted into finite state systems, the set of all initial conditions is usually under-approximated. In this paper, our solution to the feasibility envelope is *complete*, in the sense that we find the set of all initial conditions in the original system, hence we do not introduce any conservativeness. Abstracting continuous systems to finite state systems does not introduce conservativeness only if bisimulation quotients are constructed. However, current results on constructing bisimulation quotients rely on finitely many control inputs [13] or sets with no input constraints [2], whereas in this paper we assume that the control inputs are continuous and restricted to polyhedral sets. Furthermore, by using MTL rather than LTL, we consider timing constraints that are important in responding to events in many scenarios with specific deadlines. We also argue that our method is computationally more efficient than recovering the feasibility region of an MILP since many constraints introduced while translating an MTL formula to MILP are redundant.

The remaining of the paper is organized as follows. First, we provide some background on MTL in Sec. II. Next, we formalize the problem studied in this paper in Sec. III. We

The authors are with the Department of Mechanical Engineering, Boston University, Boston, MA 02215 {sadra,cbelta}@bu.edu.

This work was partially supported by the NSF under grants CPS-1446151 and CMMI-1400167.

TABLE I

WORD σ IN EXAMPLE 1

t	0	1	2	3	4	5
p_1	F	T	T	T	T	F
p_2	F	F	F	F	T	F

provide the solution in Sec. IV. An illustrative example is presented in Sec. V.

II. Preliminaries

MTL is defined over a finite set P of time varying atomic propositions. In this paper, we consider all the evolutions in discrete time $t \in \mathbb{N} = \{0, 1, 2, \cdots\}$. Each proposition $p \in P$ at time $t \in \mathbb{N}$ takes a value from the boolean set $\mathbb{B} = \{\mathrm{T}, \mathrm{F}\}$. A timed *word* is defined as $\sigma : \mathbb{N} \to 2^P$, where $\sigma[t] \in 2^P$ is the set of propositions that are true at time t. The syntax of MTL formulas is defined as:

$$\varphi ::= \mathrm{T} \mid p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \mathcal{U}_I \varphi_2,$$

where p is an atomic proposition, $\neg$ and $\wedge$ are boolean negation and conjunction operators, respectively, φ_1, φ_2 are MTL formulas and $\mathcal{U}_I$ is a timed "until" operator, where $I \subset [0, \infty)$ is a time interval. In discrete time setting, the time interval I is in the form of $[a, b]$ where $a, b \in \mathbb{N}$. Additional useful boolean and temporal operators are constructed using the syntax above. The most common are: boolean disjunction: $\varphi_1 \vee \varphi_2 := \neg(\neg\varphi \wedge \neg\varphi_2)$, temporal finally (eventually): $\mathcal{F}_I \varphi := \mathrm{T}\mathcal{U}_I \varphi$ and temporal globally (always): $G_I \varphi := \neg\mathcal{F}_I \neg\varphi$.

We denote the portion of the word σ starting at time t by $\sigma\{t\} := \{\sigma[t], \sigma[t+1], \cdots\}$. Word σ satisfies MTL formula φ, denoted by $\sigma \models \varphi$, if $\sigma\{0\} \models \varphi$. The language of φ is the set of all words satisfying φ. The semantics of MTL is inductively defined as [4]:

$$\begin{array}{lll}
\sigma\{t\} \models p & \Leftrightarrow & p \in \sigma[t], \\
\sigma\{t\} \models \varphi_1 \vee \varphi_2 & \Leftrightarrow & \sigma\{t\} \models \varphi_1 \ \vee \ \sigma\{t\} \models \varphi_2, \\
\sigma\{t\} \models \varphi_1 \wedge \varphi_2 & \Leftrightarrow & \sigma\{t\} \models \varphi_1 \ \wedge \ \sigma\{t\} \models \varphi_2, \\
\sigma\{t\} \models \varphi_1 \ \mathcal{U}_I \ \varphi_2 & \Leftrightarrow & \exists t' \in t + I \ s.t \ \sigma\{t'\} \models \varphi_2 \\
 & & \wedge \ \forall t'' \in [t, t'], \sigma\{t''\} \models \varphi_1, \\
\sigma\{t\} \models \mathcal{F}_I \varphi & \Leftrightarrow & \exists t' \in t + I \ s.t. \ \sigma\{t'\} \models \varphi, \\
\sigma\{t\} \models \mathcal{G}_I \varphi & \Leftrightarrow & \forall t' \in t + I \ s.t. \ \sigma\{t'\} \models \varphi.
\end{array} \tag{1}$$

Since the temporal operators of MTL are bounded, verifying an MTL formula requires the a finite length of the word. The *horizon* of an MTL formula φ, denoted by h^φ, is defined as the last time when the values of propositions are relevant. The horizon is recursively computed as [14]:

$$\begin{array}{l}
h^p = 0, \\
h^{\varphi_1 \wedge \varphi_2} = h^{\varphi_1 \vee \varphi_2} = \max(h^{\varphi_1}, h^{\varphi_2}), \\
h^{\mathcal{F}_{[a,b]}\varphi} = h^{\mathcal{G}_{[a,b]}\varphi} = b + h^\varphi, \\
h^{\varphi_1 \mathcal{U}_{[a,b]} \varphi_2} = b + \max(h^{\varphi_1}, h^{\varphi_2}).
\end{array} \tag{2}$$

Example 1: Let P consist of two propositions p_1 and p_2, where their time varying boolean values are given as Table 1. Consider MTL formulas $\varphi_1 = \mathcal{G}_{[1,3]} p_1 \wedge \mathcal{F}_{[0,5]} p_2$ and $\varphi_2 = \mathcal{G}_{[0,4]} p_1 \vee \mathcal{F}_{[0,3]} p_2$. We have $h^{\varphi_1} = 5$ and $h^{\varphi_2} = 4$. By applying MTL semantics from (1), it is straightforward to verify that $\sigma \models \varphi_1$ but $\sigma \not\models \varphi_2$. ■

III. Problem Formulation and Approach

We consider discrete time linear systems in the form of:

$$x[t+1] = Ax[t] + Bu[t], \tag{3}$$

where $x \in \mathcal{X}$ is the state restricted to $\mathcal{X} \subset \mathbb{R}^n$, $u \in \mathcal{U}$ is the control restricted to $\mathcal{U} \subset \mathbb{R}^m$ and $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ are system matrices. We assume that $\mathcal{X}$ and $\mathcal{U}$ are polyhedral sets.

We consider specifications described using MTL where each of its atomic propositions is over a set of linear constraints on the state in the form of:

$$p := (C_p x \leq c_p), \tag{4}$$

where $C_p \in \mathbb{R}^{d_p \times n}$, $c_p \in \mathbb{R}^d_p$ and d_p is the number of linear constraints in proposition p. The inequality is interpreted element-wise. We may also incorporate atomic propositions over controls by augmenting the state with control inputs. We assume that the specification formula does not contain negation. This assumption is not restrictive and any MTL formula can be written in negation normal form by recursively eliminating the negation operator until all negations precede the propositions [15], where $\neg p$, p in the form of (4), can be rewritten as:

$$\neg p = \bigvee_{i=1}^{d_p} (-C_{p,i}^T x \leq -c_{p,i}),$$

where $C_{p,i}$ is the i'th row of C_p.

In this paper, we wish to find the set of all initial conditions from which trajectories satisfying the MTL specification can be generated. A trajectory of the system is defined as:

$$\xi := x[0], x[1], x[2], \cdots . \tag{5}$$

The word obtained from ξ and the set of propositions P is denoted by σ_ζ. Trajectory ξ satisfies the MTL specification φ if $\sigma_\zeta \models \varphi$.

Problem 1: Given system (3) and an MTL formula φ over atomic propositions in the form of (4), find the largest set of initial conditions, or *feasibility envelope*, $\mathcal{X}_0^\varphi \subseteq \mathcal{X}$ such that:

$$\begin{array}{ll}
x_0 \in \mathcal{X}_0^\varphi \Leftrightarrow & \exists u[0], u[1], u[2], \cdots \ s.t. \ \sigma_\xi \models \varphi, \\
 & x[0] = x_0, x[.] \in \mathcal{X}, u[.] \in \mathcal{U}.
\end{array} \tag{6}$$

Note that we are seeking a complete solution to the problem above, in the sense that if $x_0 \notin \mathcal{X}_0^\varphi$, then there is no sequence of control actions that can generate a trajectory satisfying φ.

We also study event-responsive specifications, which has applications in engineering. An event is viewed as an environment request. For simplicity, we assume that an event happens only once but at an unknown time. We consider specifications of the following form:

$$e \to \varphi_e, \tag{7}$$

where φ_e is a MTL specification which is required to be satisfied if event e occurs. We also set up the problem such

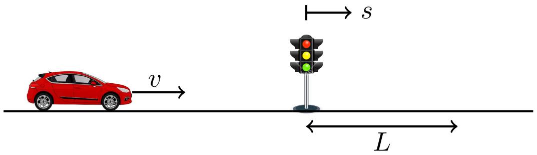

Fig. 1. A car is moving toward a traffic light. If encountered by yellow light, the choice has to be made about stopping before the traffic light or clear the intersection before the traffic light turns red.

that the occurrence of e resets the time to $t = 0$. The problem of interest is finding a set in the state space where if event e happens, there exists a control sequence producing a trajectory satisfying φ_e. This set is equivalent to $X_0^{\varphi_e}$. Since the event e may happen at any time, the state of the system has to be always in the set $X_0^{\varphi_e}$ prior to the event. Therefore, $X_0^{\varphi_e}$ can also be interpreted as the *safety envelope* of the system.

We further explain the event-responsive problem with the following car driving example known as *yellow light dilemma*, which has been extensively studied in the literature [16].

Example 2: When drivers face yellow light before reaching an intersection, the choice of speeding up or down is often troublesome. The specification, in the form of (7), is described using MTL:

$$\text{yellow} \rightarrow \mathcal{F}_{[0,T]}\left(\left((s \leq 0) \wedge (v = 0)\right) \vee (x \geq L)\right),$$

where s and v are the vehicle's position and velocity, L is the length of the intersection, as illustrated in Fig. 1, and T is the duration of yellow light. The specification requires that the car either fully stops behind the traffic light or clears the intersection before T. As mentioned earlier, the time is reset to $t = 0$ when the light turns yellow. We wish to find the feasibility envelope in the position and velocity space of the vehicle such that if the light turns yellow, there exist a sequence of control actions such that the specification is satisfied. We revisit this example formally in Sec. V.

Finding the feasibility envelope is also useful for characterization of optimal trajectories given various cost functions. For example, by finding the feasibility envelope in the yellow light dilemma problem, we are able to characterize strategies for safely driving through intersections in minimum time. Based on the polyhedral sets found to the solution to Problem 1, we also explain how to find optimal controls subject to a cost function.

IV. Solution

In this section, we provide the solution to the Problem 1. First, we show how to convert an MTL specification into a finite set of polyhedra. Next, we explain the projection procedure. We also briefly discuss how to choose optimal controls and highlight the computational limitations of our approach.

A. MTL Polyhedral Representation

Since the MTL specifications are time bounded, satisfaction of an MTL formula depends on a finite length of the trajectory characterized by the horizon of the formula. In other words, given $\{x[0], x[1], \cdots, x[h^\varphi]\}$, the satisfaction of φ is verifiable. We define the vector:

$$\xi^{h^\varphi} := \left(x[0]^T, x[1]^T, \cdots, x[h^\varphi]^T\right)^T, \tag{8}$$

which lies in $n(h^\varphi + 1)$ dimensional space $\prod_{t=0}^{h^\varphi} \mathcal{X}$.

Definition 1: A *P-collection* is a set that can be described by an union of finite number of polyhedral sets.

We define the set $\mathcal{L}(\varphi) \subset \prod_{t=0}^{h^\varphi} \mathcal{X}$ as the set of all ξ^{h^φ} that satisfy φ, which is shown to be a P-collection. $\mathcal{L}(\varphi)$ can also be viewed as the P-collection representing the language of φ. We explain how to characterize $\mathcal{L}(\varphi)$.

First, given a set of propositions P and an MTL formula φ, a boolean logic formula over the values of P over a the horizon of φ is obtained by translating the temporal operators into boolean operators. We denote the boolean logic version of φ by $\tilde{\varphi}$. The translation is recursively executed as:

$$\begin{aligned}
\sigma\{t\} \models \mathcal{G}_{[a,b]}\varphi \quad &\Rightarrow \bigwedge_{\tau=a}^{b} \quad \sigma\{t+\tau\} \models \varphi, \\
\sigma\{t\} \models \mathcal{F}_{[a,b]}\varphi \quad &\Rightarrow \bigvee_{\tau=a}^{b} \quad \sigma\{t+\tau\} \models \varphi, \\
\sigma\{t\} \models \varphi_1 \mathcal{U}_{[a,b)}\varphi_2 \quad &\Rightarrow \bigvee_{\tau=a}^{b} \quad \Big(\sigma\{t+\tau\} \models \varphi_2 \wedge \\
&\qquad\qquad \bigwedge_{\tau'=a}^{\tau} \sigma\{t+\tau'\} \models \varphi_1\Big).
\end{aligned} \tag{9}$$

Example 3 (Example 1 revisited): We apply the process in (9) to specifications φ_1 and φ_2, where we obtain $\tilde{\varphi}_1 = \big(p_1[1] \wedge p_1[2] \wedge p_1[3]\big) \wedge \big(p_2[0] \vee p_2[1] \vee p_2[2] \vee p_2[3] \vee p_2[4] \vee p_2[5]\big)$, and $\tilde{\varphi}_2 = \big(p_1[0] \wedge p_1[2] \wedge p_1[3] \wedge p_1[4]\big) \vee \big(p_2[0] \vee p_2[1] \vee p_2[2] \vee p_2[3]\big)$.

■

Next, we transform the formula $\tilde{\varphi}$ into its *disjunction normal form* (DNF) such that:

$$\tilde{\varphi} = \bigvee_{i=1}^{n_\varphi} \tilde{\varphi}_i, \tag{10}$$

where each φ_i is a conjunctive formula, i.e. consisting of conjunctions between propositions and n_φ is the number of conjunctive formulas. Each proposition represents a polyhedron in $\mathcal{X}$ and the conjunction of several propositions represents the polyhedron resulting from their intersection that lies in $\prod_{t=0}^{h^\varphi} \mathcal{X}$. We denote the polyhedron corresponding to φ_i by $\mathcal{H}_i$, which its H-representation is:

$$\mathcal{H}_i = \left\{H_i \xi^{h^\varphi} \leq h_i\right\}.$$

Finally, $\mathcal{L}(\varphi)$ is the set obtained by:

$$\mathcal{L}(\varphi) = \bigcup_{i=1}^{n_\varphi} \mathcal{H}_i, \tag{11}$$

which it is a P-collection in $\prod_{t=0}^{h^\varphi} \mathcal{X}$.

B. Polyhedral Projection

We recall the definition of the set X_0^φ as:

$$X_0^\varphi = \left\{x_0 \in \mathcal{X} \middle| \exists u[0], u[1], \cdots,\ s.t.\ \sigma_\xi \models \varphi \right\},$$

where $\xi = x[0], x[1], \cdots,\ x[0] = x_0$. Checking $\sigma_\xi \models \varphi$ only requires ξ^{h^φ}. Therefore, the relevant control sequence is $u[0], u[1], \cdots, u[h^\varphi - 1]$. We define the vector:

$$\zeta^{h^\varphi} := \left(u[0]^T, u[1]^T, \cdots, u[h^\varphi - 1]^T\right)^T,$$

which lies in $\prod_{t=0}^{h^\varphi-1}\mathcal{U}$. The finite horizon evolution of the system is given by $\xi^{h^\varphi} = A_\xi x_0 + B_\zeta \zeta^{h^\varphi}$, where

$$A_\xi = \left(\begin{array}{c} I \\ A \\ A^2 \\ \vdots \\ A^{h^\varphi} \end{array}\right), B_\zeta = \left(\begin{array}{cccc} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{h^\varphi-1}B & A^{h^\varphi-2}B & \cdots & B \end{array}\right).$$

Using the terminologies introduced in this section, we rewrite X_0^φ as:

$$X_0^\varphi = \left\{x_0 \in \mathcal{X} \middle| \exists \zeta^{h^\varphi}\ s.t.\ \xi^{h^\varphi} \in \mathcal{L}(\varphi)\right\}.$$

Proposition 1: The set $\mathcal{X}_0^\varphi$ is given by the following union:

$$\mathcal{X}_0^\varphi = \bigcup_{i=1}^{n_\varphi} \left\{x_0 \in \mathcal{X} \middle| \exists \zeta^{h^\varphi}\ s.t.\ \xi^{h^\varphi} \in \mathcal{H}_i\right\}. \qquad (12)$$

Proof: 1) $\xi^{h^\varphi} \in \mathcal{H}_i(\varphi) \Rightarrow xi^{h^\varphi} \in \mathcal{L}(\varphi)$: trivial. 2) $\xi^{h^\varphi} \in \mathcal{L}(\varphi) \Rightarrow \exists i \in \{1, n_\varphi\}\ \xi^{h^\varphi} \in \mathcal{H}_i$: Since ξ^{h^φ} satisfies φ, at least one of the conjunctive formulas in (10) must be satisfied. ■

Remark 1: The proposition above does not necessarily hold in the presence of disturbances in system (3). For systems involving disturbances, Problem 1 is reformulated as finding the set of initial conditions that for all allowable disturbances, a control strategy for satisfying the MTL specification exist. Accounting for all allowable disturbances requires computing the Pontrayagin difference for a P-collection, which is computationally very expensive [17] and is the main bottleneck in extending this work to systems with disturbances.

Next, we find the projection of each polyhedron in (12) into $\mathcal{X}$:

$$\begin{array}{rl} \mathcal{X}_0^\varphi = & \left\{x_0 \in \mathcal{X} \middle| \exists \zeta^{h^\varphi}\ s.t.\ \xi^{h^\varphi} \in \mathcal{H}_i\right\} = \\ & \bigcup_{i=1}^{n_\varphi} proj_{\mathcal{X}} \left(\left\{\begin{array}{l} H_i \xi^{h^\varphi} \leq h_i, \\ \xi^{h^\varphi} = A_\xi x_0 + B_\zeta \zeta^{h^\varphi}, \\ \xi^{h^\varphi} \in \prod_{t=0}^{h^\varphi} \mathcal{X}, \\ \zeta^{h^\varphi} \in \prod_{t=0}^{h^\varphi-1} \mathcal{U}, \end{array}\right.\right), \end{array}$$

where $proj_{\mathcal{X}}$ stands for projection into $\mathcal{X}$. Since $\mathcal{X}$ and $\mathcal{U}$ are polyhedral sets (their H-representations are not shown here), the projection operation can be carried out using *Fourier-Motzkin* elimination method [18]. The variables that are required to be eliminated are the entries in ζ^{h^φ}, which there are total mh^φ number of them. By the taking the union as in (12), we arrive in $\mathcal{X}_0^\varphi$, which it is a P-collection inside $\mathcal{X}$.

C. Optimal Control

In this section, we discuss how to find the optimal control sequence that if applied starting at $x_0 \in \mathcal{X}_0^\varphi$, the generated trajectory satisfies φ. Given a cost function over the finite horizon evolution of the system in the form of:

$$J : \prod_{t=0}^{h^\varphi} \mathcal{X} \times \prod_{t=0}^{h^\varphi-1} \mathcal{U} \to \mathbb{R}.$$

we can find the optimal controls by solving the following set of linear programs:

$$\begin{array}{rlll} \zeta_{opt}^{h^\varphi} = & argmin & \min\limits_{i \in \{1, n_\varphi\}} & J(\xi^{h^\varphi}, \zeta^{h^\varphi}) \\ & & s.t. & \xi^{h^\varphi} \in \mathcal{H}_i, \\ & & & \xi^{h^\varphi} \in \mathcal{X}^{h^\varphi}, \\ & & & \zeta^{h^\varphi} \in \prod_{t=0}^{h^\varphi-1} \mathcal{U}, \end{array}$$

where a total number of n_φ linear programs (LP) is solved. Note that by the virtue of Proposition 1, at least one of the LPs solved is feasible thus existence of a solution is guaranteed. Our method for optimal control is slightly different from the method in [8], where the optimization problem is formulated as an MILP. While MILPs are more efficient by using branch and bound method to search over LPs for finding the optimal solution, our method enumerates all the LP solutions of the conjunctive formulas obtained from the DNF in (10). In some cases, which are fairly common in applications, the time required to do the latter is significantly smaller. In general, finding the optimal controls using the DNF approach is more appropriate when the number of conjunctive forms is smaller than the linear constraints within the conjunctive formulas. This issue is illustrated in the following hypothetical example:

Example 4: Consider the specification $\tilde{\varphi} = \tilde{\varphi}_1 \vee \tilde{\varphi}_2$, where $\tilde{\varphi}_1$ and $\tilde{\varphi}_2$ are conjunctive formulas each consisting of a proposition over 100 linear constraints. The optimal control problem for this specification can be approached by solving 2 LPs. However, using the MILP approach, one has to define 200 integer variables, leading the branch and bound algorithm to search over a tree of depth 200 consisting of 2^{200} LPs. Even in the best case, about 200 distinct LPs are required to be solved.

D. Complexity

There are two possible exponential growths in the complexity of the methods presented: 1) transformation of $\tilde{\varphi}$ into its DNF and 2) the Fourier-Motzkin elimination procedure.

The total number of conjunctive formulas, n_φ, depends on the complexity of the formula φ. In general, n_φ grows exponentially with respect to the number of operators in the formula and also exponentially with respect to the length of the intervals of the temporal operators. There are $|P|(h^\varphi + 1)$ number of propositions in $\tilde{\varphi}$. Therefore, in the worst case, the number of conjunctive formulas is $2^{|P|(h^\varphi+1)}$.

The Fourier-Motzkin elimination procedure introduces additional constraints that, in the worst case, grow double exponentially with respected to number of variables that are eliminated. However, much of the constraints are redundant.

It is shown that the number of non-redundant constraints grows by single exponential [19]. Therefore, in the worst case, the number of constraints of a projected polyhedron is of order $\mathcal{O}((|P|n(h^{\varphi}+1))^{mh^{\varphi}})$. Therefore, the worst case number of constraints required for representing $\mathcal{X}_0^{\varphi}$ is of the following order:

$$\mathcal{O}((|P|n(h^{\varphi}+1))^{mh^{\varphi}})2^{|P|(h^{\varphi}+1)}).$$

V. Illustrative Example

We revisit the example explained in Sec. III. We aim to find the feasibility envelope of a car passing an intersection (as shown in Fig. 1). We consider the following double integrator model for the car:

$$\begin{pmatrix} s[t+1] \\ v[t+1] \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} s[t] \\ v[t] \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u[t], \quad (13)$$

where state is $x = (s, v)^T$. The state space is given by $\mathcal{X} = \{0 \leq v \leq 2\}$ and the acceleration input u is bounded to $\mathcal{U} = \{u \big| -0.3 \leq u \leq 0.2\}$. We recall the MTL specification:

$$\text{yellow} \to \mathcal{F}_{[0,T]}\left(((x \leq 0) \wedge (v = 0)) \vee (x \geq 10)\right). \quad (14)$$

We construct $\mathcal{R}^{\varphi}$ in $2(T+1)$ dimensional space, as explained in Sec. IV, and use Fourier-Motzkin elimination method to find $\mathcal{X}_0^{\varphi}$ in 2-dimensional space. The feasibility envelope for different values of T is shown in Fig. 2. For instance, for $T = 8$ it is observed that the feasibility envelope has two cavities. The physical interpretation of the lower cavity is straightforward but it is more subtle for the upper cavity, which explains that cars require to start to speed down within a certain distance from the intersection. On the other hand, if the light is still not turned yellow, cars can increase speed before reaching the intersection. This practice is not considered and recommended for human drivers [16], but is potentially applicable to autonomous driving.

For comparison, we have simulated two trajectories for the case $T = 8$. The first is a car driving with constant velocity $v = 1.9$ and the second is a car driving within the feasibility envelope but uses a one-step look ahead strategy to maximize its speed. The results are shown in Fig. 3. It is observed that the the state of the first car evolves out of the feasibility envelope at a time point. Therefore, if the light is turned yellow, the car is not able to stop before the traffic light or clear the intersection. However, the second car, although driving faster in average, is guaranteed to be able to properly respond to the yellow light.

VI. Conclusion and Future Work

In this paper, we characterized the feasibility envelopes for linear systems subject to specifications described by metric temporal logic. The key technical contribution of this work is a providing a method to translate the MTL constraints to polyhedral sets constructed in higher dimensions, which are then projected into the space of initial states subject to the system constraints.

Our future works involves extending the current method to MTL specifications with infinite time semantics. We plan to study a fragment of MTL that describes safety specifications. Using set-invariance methods [11], we will provide guarantees on infinite time evolution of the system. Second, we will find control invariant sets in the feasibility envelope such that the system is able to be restricted to the feasibility envelope, prior to an event in environment, for all times.

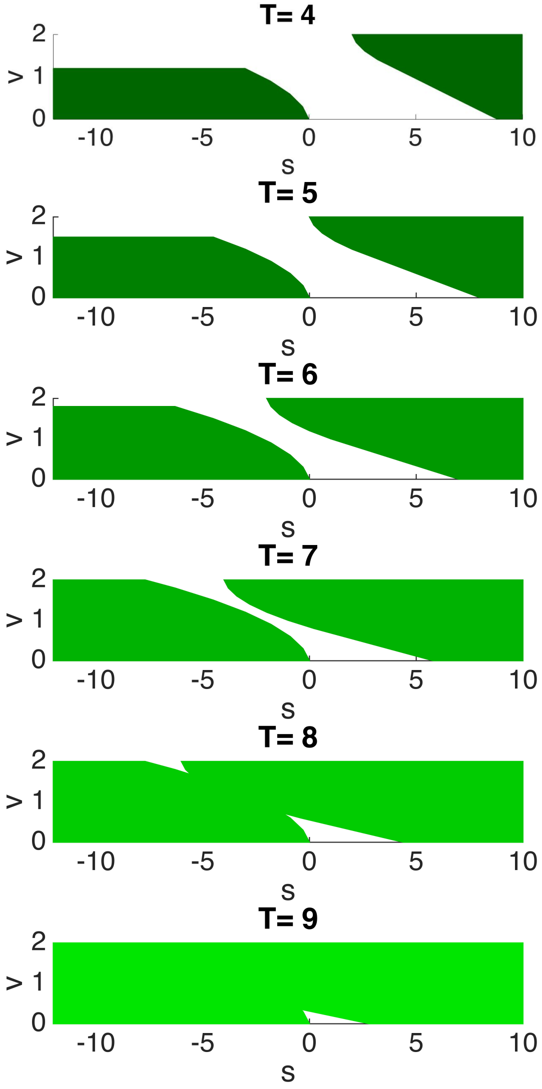

Fig. 2. Case Study: The feasibility envelope of the yellow light dilemma problem is the green shaded area shown for different values of T. The traffic light is located at $s = 0$ and the end of the intersection is at $s = 10$.

References

[1] C. Baier, J.-P. Katoen, and Others, *Principles of model checking*. MIT press Cambridge, 2008, vol. 26202649.

[2] P. Tabuada and G. J. Pappas, "Linear time logic control of discrete-time linear systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 12, pp. 1862–1877, 2006.

[3] 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.

[4] P. Thati and G. Rou, "Monitoring Algorithms for Metric Temporal Logic Specifications," in *Electronic Notes in Theoretical Computer Science*, vol. 113, no. SPEC. ISS., 2005, pp. 145–162.

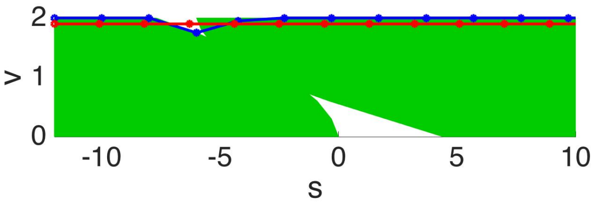

Fig. 3. Case Study: Two sample trajectories: (red) car driving at constant velocity $v = 1.9$ does not consider the feasibility envelope and is not guaranteed to be able to react properly to the yellow light. (car) car driving inside the feasibility envelope with maximum speed. The car slows down before reaching the intersection ($x = -8$) but, in case of not encountering yellow light, starts to speed up at $x = 6$.

[5] A. Dhananjayan and K. T. Seow, “A metric temporal logic specification interface for real-time discrete-event control,” *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 44, no. 9, pp. 1204–1215, 2014.

[6] J. Ouaknine and J. Worrell, “On the decidability of metric temporal logic,” in *Logic in Computer Science, 2005. LICS 2005. Proceedings. 20th Annual IEEE Symposium on*. IEEE, 2005, pp. 188–197.

[7] S. Karaman and E. Frazzoli, “Vehicle routing problem with metric temporal logic specifications,” in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, 2008, pp. 3953–3958.

[8] V. Raman, A. Donz, M. Maasoumy, R. M. Murray, A. Sangiovanni-vincentelli, and S. a. Seshia, “Model Predictive Control with Signal Temporal Logic Specifications,” in *CDC*, no. Cdc, 2014.

[9] M. Oishi, I. Mitchell, A. Bayen, C. Tomlin, and A. Degani, “Hybrid verification of an interface for an automatic landing,” in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 2. IEEE, 2002, pp. 1607–1613.

[10] J. Lygeros, C. Tomlin, and S. Sastry, “Controllers for reachability specifications for hybrid systems,” *Automatica*, vol. 35, no. 3, pp. 349–370, 1999.

[11] F. Blanchini, “Set invariance in control–a survey,” *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[12] 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.

[13] E. Aydin Gol, X. Ding, M. Lazar, and C. Belta, “Finite bisimulations for switched linear systems,” *Automatic Control, IEEE Transactions on*, vol. 59, no. 12, pp. 3122–3134, 2014.

[14] A. Dokhanchi, B. Hoxha, and G. Fainekos, “On-line monitoring for temporal logic robustness,” in *Runtime Verification*. Springer, 2014, pp. 1–20. http://link.springer.com/chapter/10.1007/978-3-319-11164-3{_}19

[15] S. Sadraddini and C. Belta, “Robust Temporal Logic Model Predictive Control,” *53rd Annual Conference on Communication, Control, and Computing (Allerton)*, 2015.

[16] C. Liu, R. Herman, and D. C. Gazis, “A review of the yellow interval dilemma,” *Transportation Research Part A: Policy and Practice*, vol. 30, no. 5, pp. 333–348, 1996.

[17] S. V. Raković, P. Grieder, M. Kvasnica, D. Q. Mayne, and M. Morari, “Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances,” in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 2. IEEE, 2004, pp. 1418–1423.

[18] D. Bertsimas, J. N. Tsitsiklis, and J. Tsitsiklis, *Introduction to Linear Optimization*. Athena Scientific Belmont, MA, 1997, vol. 6.

[19] D. Monniaux, “A quantifier elimination algorithm for linear real arithmetic,” in *Logic for Programming, Artificial Intelligence, and Reasoning*. Springer, 2008, pp. 243–257.