Automatica 56 (2015) 78–85

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Temporal logic model predictive control ☆

Ebru Aydin Gol [a], Mircea Lazar [b], Calin Belta [a]

[a] Division of Systems Engineering, Boston University, Boston, MA, 02215, USA

[b] Department of Electrical Engineering, Eindhoven University of Technology, Den Dolech 2, 5600MB, Eindhoven, The Netherlands

ARTICLE INFO

Article history:
Received 22 December 2013
Received in revised form
28 July 2014
Accepted 14 March 2015
Available online 14 April 2015

Keywords:
Model predictive control
Discrete-time systems
Formal methods
Linear temporal logic

ABSTRACT

This paper proposes an optimal control strategy for a discrete-time linear system constrained to satisfy a temporal logic specification over a set of linear predicates in its state variables. The cost is a quadratic function that penalizes the distance from desired state and control trajectories. The specification is a formula of syntactically co-safe Linear Temporal Logic (scLTL), which can be satisfied in finite time. To incorporate dynamic environments, it is assumed that the reference trajectories are only available over a finite horizon and a model predictive control (MPC) approach is employed. The MPC controller solves a set of convex optimization problems guided by the specification and subject to progress constraints. The constraints ensure that progress is made towards the satisfaction of the formula with guaranteed satisfaction by the closed-loop trajectory. The algorithms proposed in this paper were implemented as a software package that is available for download. Illustrative case studies are included.

1. Introduction

In recent years, there has been an increasing interest in formal synthesis of control strategies for dynamical systems (Bhatia, Kavraki, & Vardi, 2010; Gazit, Fainekos, & Pappas, 2007; Girard, 2010; Aydin Gol, Lazar, & Belta, 2014; Sloth & Wisniewski, 2013; Tabuada & Pappas, 2003; Wongpiromsarn, Topcu, & Murray, 2009; Yordanov, Tumova, Belta, Cerna, & Barnat, 2012). Unlike "classical" control problems, in which the specifications are stability or closeness to a reference point or trajectory possibly coupled with safety, the above works allow for richer specifications that translate to formulas of temporal logics such as Linear Temporal Logic (LTL) (Baier & Katoen, 2008) and fragments of LTL.

We consider synthesis of optimal control strategies from temporal logic specifications. While some results exist for finite systems (Ding, Lazar, & Belta, 2012; Ding, Smith, Belta, & Rus, 2011), this problem is largely open for systems with infinitely many states. We focus on MPC of discrete-time linear systems subject to scLTL formulas over linear predicates in the state variables. The cost is a quadratic function that penalizes the distance between the actual and desired state and control trajectories over a finite time horizon. The goal is to find a control strategy such that the trajectory of the closed-loop system originating from a given initial state satisfies the formula and minimizes the cost. The syntactically co-safe fragment of LTL is rich enough to express a wide spectrum of finite-time properties of dynamical systems, *e.g.*, "Go to A or B and avoid C for all times before reaching T. Do not go to D unless E was visited before".

Our approach consists of two main steps. First, by using the framework developed in Aydin Gol et al. (2014), we perform an iterative partitioning of the state space guided by an automaton enforcing the satisfaction of the scLTL formula. Second, we design an MPC controller over the automaton and the state space of the system. Essentially, we use the automaton to translate the formula into a type of constraint that can be embedded into the MPC problem. The proposed MPC controller produces an optimal control sequence with respect to the available reference trajectory by solving a set of quadratic programs (QPs). The first control is applied and the process is repeated until a final state of the automaton is reached. The constraints of the optimization problem guarantee that the produced trajectory follows an automaton path while making progress towards a final state. The main contribution of this work is the proposed specification-guided MPC framework, in which the satisfaction of the specification by the closed-loop trajectory is guaranteed while the cost over the available finite

☆ This work was partially supported at Boston University by the ONR under grants MURI 014-001-0303-5, MURI N00014-10-10952, MURI N00014-09-1051 and by the NSF under grants CNS-1035588, CMMI-1400167. The material in this paper was partially presented at the Hybrid Systems: Computation and Control (HSCC 2013) Conference, April 8–11, 2013, Philadelphia, USA. This paper was recommended for publication in revised form by Associate Editor Martin Guay under the direction of Editor Frank Allgöwer.

E-mail addresses: ebruaydin@gmail.com (E. Aydin Gol), m.lazar@tue.nl (M. Lazar), cbelta@bu.edu (C. Belta).

http://dx.doi.org/10.1016/j.automatica.2015.03.029
0005-1098/

horizon reference trajectories is minimized at each time step. The framework was implemented as a software package, which is downloadable from hyness.bu.edu/software.

A preliminary version of this work appeared in conference proceedings Aydin Gol and Lazar (2013), where we defined a Lyapunov-type function over the state spaces of the dynamical system and the automaton obtained from the first step, *i.e.,* the automaton refinement step. The function was based on the controllers considered during the refinement step, and was used to enforce the progress towards an accepting state of the automaton in the MPC problem. Here, we expand Aydin Gol and Lazar (2013) by generalizing this idea and introducing a class of Lyapunov-type functions, which we call *potential* functions. The new potential functions do not necessarily depend on the previously designed controllers. Moreover, we relax the progress constraint, which together with the new potential functions allow us to reduce the cost.

2. Notation and preliminaries

For a set $\mathscr{S}$, $\mathrm{Co}(\mathscr{S})$, $\#\mathscr{S}$, and $2^{\mathscr{S}}$ stand for its convex hull, cardinality, and power set, respectively. We use $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$, and $\mathbb{Z}_+$ to denote the sets of real numbers, non-negative reals, integer numbers, and non-negative integers. For $m, n \in \mathbb{Z}_+$, with $m, n \geq 1$, we use $\mathbb{R}^n$ and $\mathbb{R}^{m\times n}$ to denote the set of column vectors and matrices with n and $m \times n$ real entries, respectively. A polyhedron (polyhedral set) in $\mathbb{R}^n$ is the intersection of a finite number of open and/or closed half-spaces. A polytope is a compact polyhedron. We use $\mathscr{V}(\mathscr{P})$ to denote the set of vertices of a polytope $\mathscr{P}$.

A discrete-time linear control system is defined as

$$x_{k+1} = Ax_k + Bu_k, \quad x_k \in \mathbb{X},\ u_k \in \mathbb{U}, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ describe the system dynamics, $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{U} \subset \mathbb{R}^m$ are polyhedral sets, and $x_k \in \mathbb{X}$ and $u_k \in \mathbb{U}$ are the state and the applied control at time $k \in \mathbb{Z}_+$, respectively. An atomic proposition p defined as a linear inequality in $\mathbb{R}^n$ induces a half-space

$$[p] := \{x \in \mathbb{R}^n \mid c^\top x + d \leq 0\}, \quad c_i \in \mathbb{R}^n,\ d \in \mathbb{R}, \tag{2}$$

i.e., $[p] \subset \mathbb{R}^n$ is the set of states that satisfy p.

The control specifications are given as formulas of syntactically co-safe linear temporal logic (scLTL) (Kupferman & Vardi, 2001) over linear predicates. Roughly, an scLTL formula is built up from a set of atomic propositions P, standard Boolean operators: $\neg$ (negation), $\vee$ (disjunction), $\wedge$ (conjunction), and temporal operators: $\bigcirc$ (next), $\mathscr{U}$ (until) and $\lozenge$ (eventually). The semantics of LTL formulas are given over infinite words $\sigma = \sigma_0\sigma_1\ldots$ where $\sigma_i \in 2^P$ for all i. A word σ satisfies an scLTL formula ϕ, if the formula is true at the first position of the word, *i.e.,* σ_0. Intuitively, $\bigcirc\,\phi_1$ is true if ϕ_1 is true at the next position of the word, $\phi_1\,\mathscr{U}\,\phi_2$ is true if ϕ_2 eventually becomes true and ϕ_1 is true until this happens, and $\lozenge\,\phi_1$ is true if ϕ_1 becomes true at some future position in the word.

Given a set of atomic propositions $P = \{p_i\}_{i=0,\ldots,l}$ in the form of linear predicates (see (2)), a trajectory $x_0x_1\ldots$ of system (1) produces a word $P_0P_1\ldots$ where $P_i \subseteq P$ is the set of atomic propositions satisfied by x_i, *i.e.,* $P_i = \{p_j \mid x_i \in [p_j]\}$. scLTL formulas over the set of predicates P can therefore be interpreted over such words. A system trajectory satisfies an scLTL formula over P if the word produced by the trajectory satisfies the corresponding formula.

In Aydin Gol et al. (2014), we considered the problem of controlling discrete-time linear systems from scLTL specifications, and developed a language-guided procedure for the automatic computation of sets of initial states and feedback controllers such that all the resulting trajectories of the closed-loop system satisfy the formula. The procedure involved construction of a finite state automaton (FSA) that accepts all words satisfying the scLTL formula (Kupferman & Vardi, 2001), and taking the dual of the FSA by interchanging its states and transitions. The states of the dual automaton were associated with the regions of the linear system through linear predicates, and the transitions induced region to region controller synthesis problems. The final step was the refinement of the dual automaton until feasible transition controllers were obtained.

Definition 2.1. The dual automaton obtained from the refinement algorithm given in Aydin Gol et al. (2014) is denoted by

$$\mathscr{A}^D = (Q^D, \to^D, 2^P, \tau^D, Q_0^D, F^D), \tag{3}$$

where Q^D is a finite set of states, $\to^D \subseteq Q^D \times Q^D$ is a set of transitions, 2^P is a set of symbols, $\tau^D : Q^D \to 2^P$ is an output function, $Q_0^D \subseteq Q^D$ is a set of initial states and $F^D \subseteq Q^D$ is a set of final states. The region of a state $q \in Q^D$ is denoted by $\mathscr{P}_q \subset \mathbb{X}$.

The transitions of the dual automaton are labeled with a weight function $w : \to^D \to \mathbb{Z}_+$ such that for a transition $(q, q') \in \to^D$ the transition controller synthesized during the refinement step guarantees that all trajectories originating form $\mathscr{P}_q$ reaches $\mathscr{P}_{q'}$ within $w((q, q'))$ steps while remaining in $\mathscr{P}_q$ until they reach $\mathscr{P}_{q'}$. An accepting run $r_{\mathscr{A}^D}$ of a dual automaton is a sequence of states $r_{\mathscr{A}^D} = q_0\ldots q_d$ such that $q_0 \in Q_0^D$, $q_d \in F^D$ and $(q_i, q_{i+1}) \in \to^D$ for all $i = 0, \ldots, d-1$. An accepting run $r_{\mathscr{A}^D}$ produces a word $\sigma = \sigma_0\ldots\sigma_d$ over 2^P such that $\tau(q_i) = \sigma_i$, for all $i = 0, \ldots, d$. The construction of the dual automaton guarantees that $\tau^D(q) = \{p_i \mid x_1 \in [p_i]\} = \{p_j \mid x_2 \in [p_j]\}$ for any $x_1, x_2 \in \mathscr{P}_q$, $q \in Q^D$. Therefore, any system trajectory $x_0\ldots x_d$ that follows a sequence of regions $\mathscr{P}_{q_0}\ldots\mathscr{P}_{q_d}$, *i.e.,* $x_i \in \mathscr{P}_{q_i}$, defined by an accepting automaton run $r_{\mathscr{A}^D} = q_0\ldots q_d$ satisfies the specification. Furthermore, any satisfying trajectory of system (1) follows a sequence of polyhedral sets defined by an accepting run of $\mathscr{A}^D$.

Assumption 2.2. For any $q_0 \in Q^D$ there exists an automaton path $q_0\ldots q_d$, $d \in \mathbb{Z}_+$ such that $w(q_i, q_{i+1}) < \infty$ for all $i = 0, \ldots, d-1$ and $q_d \in F^D$.

3. Problem formulation

Consider a system as defined in (1), and a set of atomic propositions $P = \{p_i\}_{i=0,\ldots,l}$, $l \geq 1$, given as linear inequalities over the system states. Let $x_0^r x_1^r\ldots$ and $u_0^r u_1^r\ldots$ denote a reference trajectory and a reference control sequence, respectively. We assume that, for some N, at time k the reference trajectory of length $N+1$, $x_k^r\ldots x_{k+N}^r$, and the reference control sequence of length N, $u_k^r\ldots u_{k+N-1}^r$, are known. At time $k \in \mathbb{Z}_+$, the cost of a finite trajectory $x_k\ldots x_{k+N}$ originating at x_k and generated by the control sequence $\mathbf{u}_k = u_k\ldots u_{k+N-1}$ is defined with respect to the available reference trajectory and control sequence as follows:

$$\begin{aligned} C(x_k, \mathbf{u}_k) = {} & (x_{k+N} - x_{k+N}^r)^\top L_N (x_{k+N} - x_{k+N}^r) \\ & + \sum_{i=0}^{N-1} \big\{ (x_{k+i} - x_{k+i}^r)^\top L (x_{k+i} - x_{k+i}^r) \\ & + (u_{k+i} - u_{k+i}^r)^\top R (u_{k+i} - u_{k+i}^r) \big\}. \end{aligned} \tag{4}$$

$L, L_N \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are positive definite matrices.

Problem 3.1. Given an scLTL formula Φ over a set of linear predicates P, a dynamical system as defined in (1), and an initial state $x_0 \in \mathbb{X}$, find a feedback control strategy such that the closed-loop trajectory originating at x_0 satisfies Φ while minimizing the cost (4).

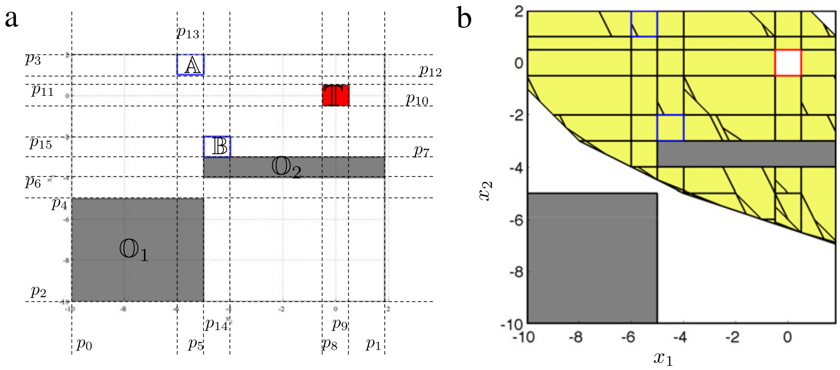


Fig. 1. (a) The regions and the corresponding linear predicates for the specification from Example 3.2. The predicates are shown in the half planes where they are satisfied; (b) The set of satisfying initial states (shown in yellow). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We propose a two-step solution to Problem 3.1. In the first step, we use the procedure presented in Aydin Gol et al. (2014) to construct a refined dual automaton (see Definition (3)). In the second step, we design an MPC controller that minimizes the cost over the available reference trajectory, while ensuring that the resulting trajectory satisfies the specification. The following example illustrates the first step of the proposed solution.

Example 3.2. We consider the double integrator dynamics with sampling time of 1 s, which are described by (1) with $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$. The control constraint set is $\mathbb{U} = \{u \mid -2 \leq u \leq 2\}$ and the initial state is $x_0 = \begin{bmatrix} 1.4 \\ -2.8 \end{bmatrix}$. The specification is to visit region $\mathbb{A}$ or region $\mathbb{B}$, and then the target region $\mathbb{T}$, while always avoiding obstacles $\mathbb{O}_1$ and $\mathbb{O}_2$, and staying inside a safe region $\mathbb{X} = \{x \mid -10 \leq x_1 \leq 1.85, -10 \leq x_2 \leq 2\}$. These polyhedral regions, together with the linear predicates used in their definitions, are shown in Fig. 1. The specification can be written as the following scLTL formula:

$\Phi_1 = ((p_0 \wedge p_1 \wedge p_2 \wedge p_3 \wedge \neg(p_4 \wedge p_5) \wedge \neg(\neg p_5 \wedge \neg p_6 \wedge \neg p_7))\ \mathscr{U}\ (p_8 \wedge p_9 \wedge p_{10} \wedge p_{11})) \wedge (\neg(p_8 \wedge p_9 \wedge p_{10} \wedge p_{11})\ \mathscr{U}\ ((p_5 \wedge p_{12} \wedge p_{13}) \vee (\neg p_5 \wedge p_7 \wedge p_{14} \wedge p_{15})))$.

The refined dual automaton obtained from the language-guided procedure has 132 states and 354 transitions with finite weights. The set of satisfying initial states $\mathbb{X}_0^{\Phi_1}$ is shown in Fig. 1(b). The computation took 83 s on an iMac with an Intel Core i5 processor at 2.8 GHz and 8 GB memory.

4. Potential function

In control theory, control Lyapunov functions (CLFs) are used to enforce closed-loop stability of an equilibrium point. In this paper, we define a Lyapunov-type function to enforce the satisfaction of the accepting condition of an automaton.

Definition 4.1. A function $V : \bigcup_{q \in Q^D} \{\{q\} \times \mathscr{P}_q\} \to \mathbb{Z}_+$ is called a *potential function* for a system (1) and a dual automaton (3) if it satisfies:

(i) $V(q, x) = 0$ for all $q \in F^D$.
(ii) For each $(q, x) \in \bigcup_{q \in Q^D} \{\{q\} \times \mathscr{P}_q\}$, it holds that if $V(q, x) \neq 0$ and $V(q, x) \neq \infty$, then there exists a control $u \in \mathbb{U}$ such that $x' = Ax + Bu$, $x' \in \mathscr{P}_{q'}$, $(q, q') \in \to^D$, and $V(q', x') < V(q, x)$.

To define a class of functions that satisfies properties (i) and (ii) of Definition 4.1, we introduce a *successor* function that induces a partial order over the states of the automaton.

Definition 4.2. A function $S : Q^D \to Q^D$ is called a *successor function* for a dual automaton (3) if it satisfies:

(i) $(q, S(q)) \in \to^D$ and $w((q, S(q))) \neq \infty$ for all $q \in Q^D \setminus F^D$.
(ii) $S(q) = q$ if and only if $q \in F^D$.
(iii) For each $q \in Q^D$, there exists a finite $d \in \mathbb{Z}_+$ such that $S^d(q) \in F^D$, where $S^d(q) = S(S^{d-1}(q))$ and $S^0(q) = q$.

A successor function $S(\cdot)$ induces a partial order $\preceq_S$ over Q^D such that $S^d(q) \preceq_S q$ for all $d \in \mathbb{Z}_+$. Moreover final states of $\mathscr{A}^D$ are fixed points of the function $S(\cdot)$.

We define an *automaton potential function* $V_{aut,S} : Q^D \to \mathbb{Z}_+$ recursively for a given successor function $S(\cdot)$ as follows:

$$V_{aut,S}(q) = \begin{cases} 0 & \text{if } q \in Q^F, \\ w((q, S(q))) + V_{aut,S}(S(q)) & \text{otherwise.} \end{cases} \tag{5}$$

Note that property 4.2-(i) and Assumption 2.2 guarantee that $V_{aut,S}(q) < \infty$ for all $q \in Q^D$. Moreover $V_{aut,S}(q') \leq V_{aut,S}(q)$ if $q' \preceq_S q$.

Definition 4.3. A function

$$V_{con} : \bigcup_{(q,q') \in \to^D, q \neq q'} \{\{(q, q')\} \times \mathscr{P}_q\} \longrightarrow \mathbb{Z}_+ \setminus \{0\}$$

is called a *control potential function* for a system (1) and a dual automaton (3) if it satisfies:

(i) $V_{con}((q, q'), x) \leq w((q, q'))$.
(ii) If $V_{con}((q, q'), x) \neq \infty$, then there exists a control $u \in \mathbb{U}$ such that either $Ax + Bu \in \mathscr{P}_{q'}$ or $Ax + Bu \in \mathscr{P}_q$ and $V_{con}((q, q'), Ax + Bu) < V_{con}((q, q'), x)$.

We define a set $\mathscr{R}^{k,qq'}_{V_{con}} \subseteq \mathscr{P}_q$, $k \in \mathbb{Z}_+$, for a given control potential function $V_{con}(\cdot, \cdot)$ and a transition $(q, q') \in \to^D$ as

$$\mathscr{R}^{k,qq'}_{V_{con}} = \{x \in \mathscr{P}_q \mid V_{con}((q, q'), x) \leq k\}. \tag{6}$$

In the rest of the paper, we assume that the set (6) is described by unions of polytopes, which will be instrumental for the MPC controller design.

Finally, we define the *potential* at (q, x) for given successor function $S(\cdot)$, control potential function $V_{con}(\cdot, \cdot)$, and automaton potential function $V_{aut,S}(\cdot)$ as

$$V_S(q, x) = \begin{cases} 0 & \text{if } q \in F^D, \\ V_{con}((q, S(q)), x) + V_{aut,S}(S(q)) & \text{otherwise.} \end{cases} \tag{7}$$

Informally, the candidate potential function (7) at (q, x), $q \in Q^D$, $x \in \mathscr{P}_q$ is defined as an upper bound for the time required to reach $\mathscr{P}_{S^d(q)}$ from x by applying the corresponding polytope-to-polytope feedback controllers along an automaton path defined by the successor function $qS(q) \ldots S^d(q)$, where $d \in \mathbb{Z}_+$ and $S^d(q)$ is a final state of the automaton.

Proposition 4.4. *The function defined in (7) is a potential function according to Definition 4.1.*

Proof. The property (i) of Definition 4.1 is satisfied trivially by the function $V_S(\cdot,\cdot)$ (7). To prove that the function satisfies 4.1-(ii), we consider two cases: $V_{con}((q, S(q)), x) = 1$ and $V_{con}((q, S(q)), x) > 1$.

Note that $V_{con}((q, S(q)), x) = 1$ and property 4.3-(ii) imply that there exists $u \in \mathbb{U}$ such that $Ax + Bu \in \mathscr{P}_{S(q)}$. As such, in the case $V_{con}((q, S(q)), x) = 1$, the claim holds as $V_S(S(q), x') \le V_{aut,S}(S(q))$ for all $x' \in \mathscr{P}_{S(q)}$. In the case when $V_{con}((q, S(q)), x) > 1$, from 4.2-(i) and 4.3-(i), it holds that $V_{con}((q, S(q)), x) \neq \infty$. By property 4.3-(ii), there exists $u \in \mathbb{U}$ such that one of the following holds:

(a) $Ax + Bu \in \mathscr{P}_{q'}$,
(b) $Ax + Bu \in \mathscr{P}_q$ and $V_{con}((q, q'), Ax + Bu) < V_{con}((q, q'), x)$. The claim is true for (a) by the same argument given for the $V_{con}((q, S(q)), x) = 1$ case. For (b), the claim holds trivially by the definition of the function $V_S(\cdot,\cdot)$ (7). □

5. MPC strategy

In this section, we present an MPC scheme to solve Problem 3.1 for a given dual automaton $\mathscr{A}^D = (Q^D, \to^D, 2^P, \tau^D, Q_0^D, F^D)$ and a transition weight function $w : \to^D \longrightarrow \mathbb{Z}_+$. We formulate an MPC optimization problem over $\bigcup_{q\in Q^D} \{\{q\} \times \mathscr{P}_q\}$ to be solved "online" at each time step.

Definition 5.1. An automaton-enabled finite trajectory $\mathbf{T} = (q_0, x_0) \dots (q_N, x_N)$ is a sequence of automaton (3) and system (1) state pairs such that

(i) for each $k = 0, \dots, N-1$ there exists $u_k \in \mathbb{U}$ such that $x_{k+1} = Ax_k + Bu_k$,
(ii) $x_k \in \mathscr{P}_{q_k}$, for all $k = 0, \dots, N$,
(iii) $(q_k, q_{k+1}) \in \to^D$, for all $k = 0, \dots, N-1$.

The definition of an automaton-enabled trajectory implies that the projection $\gamma_{\mathscr{A}}(\mathbf{T}) = q_0 \dots q_N$ of the trajectory onto the automaton states is an automaton path and the projection $\gamma_X(\mathbf{T}) = x_0 \dots x_N$ onto the state space of system (1) is a trajectory of system (1) that follows the sequence of polyhedra defined by the automaton path.

The construction of the dual automaton $\mathscr{A}^D$ from Section 2 and Definition 5.1 guarantees that for any satisfying trajectory $\mathbf{x} = x_0 \dots x_d$, $d \in \mathbb{Z}_+$ of system (1) originating at x_0, there exists an automaton-enabled trajectory $\mathbf{T}$ such that $\gamma_X(\mathbf{T}) = \mathbf{x}$ and $\gamma_{\mathscr{A}}(\mathbf{T})$ is an accepting run of $\mathscr{A}^D$. Therefore, in the MPC controller design, we restrict our attention to the control sequences that generate automaton-enabled trajectories. We use $\mathbf{U}_N(q, x)$ to denote the set of all control sequences of length N that produce automaton-enabled trajectories starting from (q, x) as characterized in Definition 5.1. By following the standard MPC notation, we use $\mathbf{T}_k = (q_{0|k}, x_{0|k}) \dots (q_{N|k}, x_{N|k})$ to denote a *predicted* automaton-enabled trajectory originating at (q_k, x_k), *i.e.,* $q_{0|k} = q_k$, $x_{0|k} = x_k$, at time $k \in \mathbb{Z}_+$.

At each time-step $k \in \mathbb{Z}_+$, we solve an optimization problem over $\mathbf{U}_N(q_k, x_k)$ and find the optimal automaton-enabled trajectory $\mathbf{T}^{*k} = (q^*_{0|k}, x^*_{0|k}) \dots (q^*_{N|k}, x^*_{N|k})$ generated by the optimal control sequence $\mathbf{u}^*_k \in \mathbf{U}_N(q_k, x_k)$, until a final automaton state is reached, *i.e.,* $q_k \in F^D$. As the first control of the optimal control sequence is applied, we have the following relation:

$$x_{k+1} = x^*_{1|k}, \qquad q_{k+1} = q^*_{1|k}, \quad k = 0, 1, 2, \dots. \tag{8}$$

To guarantee the satisfaction of the specification, we enforce a constraint on the predicted trajectory by using the potential functions presented in Section 4. Suppose that $V_{con}(\cdot,\cdot)$ is a control potential function (Definition 4.3), $\underline{S}(\cdot)$ and $\overline{S}(\cdot)$ are successor functions (Definition 4.2) such that the corresponding automaton potential functions $V_{aut,\underline{S}}(\cdot)$ and $V_{aut,\overline{S}}(\cdot)$ as defined in (5) satisfy:

$$V_{aut,\underline{S}}(q) \le V_{aut,\overline{S}}(q), \quad \text{for all } q \in Q^D. \tag{9}$$

Furthermore, let $V_{\underline{S}}(\cdot,\cdot)$ be the potential function defined by $\underline{S}(\cdot)$ and $V_{con}(\cdot,\cdot)$ as given in (7). Given these functions, the MPC optimization problem is formulated as follows:

$$\min_{\mathbf{u}_k \in \mathbf{U}_N(q_k, x_k)} C(x_k, \mathbf{u}_k), \tag{10}$$

subject to $V_{\underline{S}}(q_{N|k}, x_{N|k}) < \upsilon_k$,

where $\upsilon_k \in \mathbb{Z}_+$, $\upsilon_0 = V_{aut,\overline{S}}(q_0)$ and for $k \ge 1$, υ_k is defined by

$$\upsilon_k = \min\{\upsilon_{k-1} - 1, V_{aut,\overline{S}}(q^*_{N|k-1})\}. \tag{11}$$

For $k \ge 1$, the optimal predicted trajectory obtained at the previous time step and υ_{k-1} are used to enforce the satisfaction of the specification. Specifically, the predicted trajectory at time k must end in a state, for which there exists a control sequence guaranteeing that the trajectory originating from that state reaches a final automaton state within υ_k steps by following the sequence of polyhedra defined by the successor function $\underline{S}(\cdot)$. It is important to note that the bound υ_k is computed according to $\overline{S}(\cdot)$, and by (7) and (9), $V_{aut,\overline{S}}(q^*_{N|k-1})$ is an upper bound on $V_{\underline{S}}(q^*_{N|k-1}, x^*_{N|k-1})$. As the definition of the bound υ_k implies that $\upsilon_{k+1} < \upsilon_k$, the resulting trajectory eventually reaches a final automaton state.

Next, we show that the optimal solution of the MPC problem given in (10) can be found by solving a finite number of convex optimization problems. First, consider the set of automaton paths of length N that originate from q_k, *i.e.,*

$$\mathbf{P}^N_{q_k} = \{q_{0|k} q_{1|k} \dots q_{N|k} \mid q_{0|k} := q_k, (q_{i|k} q_{i+1|k}) \in \to^D, \\ i = 0, \dots, N-1\}. \tag{12}$$

Since Q^D and $\to^D$ are finite sets, $\mathbf{P}^N_{q_k}$ is a finite set. The definition of an automaton-enabled trajectory $\mathbf{T}_k$ of horizon N (Definition 5.1) implies that $\gamma_{\mathscr{A}}(\mathbf{T}_k) \in \mathbf{P}^N_{q_k}$ for any trajectory that can be produced by a control sequence from the set $\mathbf{U}_N(q_k, x_k)$. Given a finite automaton path $\mathbf{q}_k \in \mathbf{P}^N_{q_k}$, let $\mathbf{U}^{\mathbf{q}_k}_N(q_k, x_k)$ denote the set of all control sequences that produce an automaton-enabled trajectory $\mathbf{T}_k$ with $\gamma_{\mathscr{A}}(\mathbf{T}_k) = \mathbf{q}_k$. Essentially, $\mathbf{U}^{\mathbf{q}_k}_N(q_k, x_k)$ is the set of all control sequences that produce trajectories of system (1) that originate at x_k and follow the sequence of polyhedra defined by $\mathbf{q}_k$. Then, it is straightforward to see that $\mathbf{U}_N(q_k, x_k) = \bigcup_{\mathbf{q}_k \in \mathbf{P}^N_{q_k}} \mathbf{U}^{\mathbf{q}_k}_N(q_k, x_k)$. Consider a path $\mathbf{q}_k = q_{0|k} \dots q_{N|k} \in \mathbf{P}^N_{q_k}$ and the following optimization problem in the variables $\mathbf{u}_k = u_{0|k} \dots u_{N-1|k}$:

$$\min_{\mathbf{u}_k} C(x_k, \mathbf{u}_k),$$

subject to

$$x_{i|k} \in \mathscr{P}_{q_{i|k}}, \quad i = 1, \dots, N, \tag{13a}$$

$$u_{i|k} \in \mathbb{U}, \quad i = 0, \dots, N-1, \tag{13b}$$

$$V_{\underline{S}}(q_{N|k}, x_{N|k}) < \upsilon_k, \tag{13c}$$

where υ_k is as defined in (11). The set of control sequences that satisfy constraints (13a) and (13b) is $\mathbf{U}^{\mathbf{q}_k}_N(q_k, x_k)$. Therefore, the optimal solution of the MPC problem given in (10) can be found by solving an optimization problem as given in (13) for each $\mathbf{q}_k \in \mathbf{P}^N_{q_k}$.

By the definition of the potential function given in (7), the progress constraint given in (13c) at time $k \ge 0$ becomes:

$$V_{con}((q_{N|k}, \underline{S}(q_{N|k})), x_{N|k}) < \upsilon_k - V_{aut,\underline{S}}(\underline{S}(q_{N|k})). \tag{14}$$

Let $\bar{k} := \upsilon_k - 1 - V_{aut,\underline{S}}(\underline{S}(q_{N|k}))$. If $\bar{k} \ge w(q_{N|k}, \underline{S}(q_{N|k}))$, then the inequality given in (14) is trivially satisfied for all $x_{N|k} \in \mathscr{P}_{q_{N|k}}$. If, however $\bar{k} < w(q_{N|k}, \underline{S}(q_{N|k}))$, then the inequality is satisfied only if

$$x_{N|k} \in \mathscr{R}_{V_{con}}^{k, q_{N|k} \underline{S}(q_{N|k})}. \tag{15}$$

As such, if the set $\mathscr{R}_{V_{con}}^{k, q_{N|k} \underline{S}(q_{N|k})}$ (see (6)) is a polytope or union of polytopes, then the optimal solution of the optimization problem (13) can be found by solving a convex optimization problem for each of these polytopes.

To guarantee that the resulting closed-loop trajectory of system (1) reaches a region $\mathscr{P}_{q_f}$, where $q_f \in F^D$, at each time-step k the prediction horizon, denoted as I_k, is determined with respect to the predicted trajectory obtained at the previous step. Specifically, the length of the observed reference trajectory, N, is used as the initial prediction horizon I_0 at time-step $k = 0$. Then, for time-step $k \geq 1$, if the predicted trajectory obtained at the previous step visits a final state at position j for the first time, $j - 1$ is used as the prediction horizon I_k. Otherwise, the same prediction horizon as in the previous time-step, I_{k-1}, is used. The following function is used to determine the prediction horizon for a given trajectory $\mathbf{T}_k = (q_{0|k}, x_{0|k}) \dots (q_{I_k|k}, x_{I_k|k})$:

$$I(\mathbf{T}_k) = \begin{cases} I_k & \text{if } 0 < V_{\underline{S}}(q_{i|k}, x_{i|k}), \quad \forall i = 0, \dots, I_k \\ j - 1 & \text{if } 0 < V_{\underline{S}}(q_{i|k}, x_{i|k}), \quad \forall i = 0, \dots, j-1, \\ & V_{\underline{S}}(q_{j|k}, x_{j|k}) = 0. \end{cases} \tag{16}$$

Algorithm 1 Automaton-guided MPC

Input: $\mathcal{A}^D = (Q^D, \rightarrow^D, \Gamma^D, \tau^D, Q_0^D, F^D)$, transition weight function $w : \rightarrow^D \rightarrow \mathbb{Z}_+$, an initial condition $x_0 \in \mathcal{P}_{q_0}$ for some $q_0 \in Q_0^D$, MPC horizon N, and functions $V_{con}(\cdot, \cdot)$, $\underline{S}(\cdot)$ and $\bar{S}(\cdot)$.

1: Set $k := 0$, $I_k := N$, $v_0 := V_{aut,\bar{S}}(q_0)$. (Initialization)
2: **while** $q_k \notin F^D$ **do**
3: $\quad$ $OptCost = \infty$, $\mathbf{u}_k^* = \emptyset$, $\mathbf{T}_k^* = \emptyset$.
4: $\quad$ Compute $\mathbf{P}_{q_k}^{I_k}$.
5: $\quad$ **for all** $\mathbf{q} = q_{0|k} \dots q_{I_k|k} \in \mathbf{P}_{q_k}^{I_k}$ **do**
6: $\quad\quad$ **if** $V_{aut,\underline{S}}(\underline{S}(q_{I_k|k})) < v_k - 1$ **then**
7: $\quad\quad\quad$ $c = \min_{\mathbf{u}_k = u_{0|k} \dots u_{I_k - 1|k}} C(x_k, \mathbf{u}_k)$ subject to (13a), (13b) and (13c)
8: $\quad\quad\quad$ **if** $c < OptCost$ **then**
9: $\quad\quad\quad\quad$ $OptCost := c$, set $\mathbf{u}_k^*$ and $\mathbf{T}_k^*$ with respect to the solution of the optimization problem (line 7).
10: $\quad\quad\quad$ **end if**
11: $\quad\quad$ **end if**
12: $\quad$ **end for**
13: $\quad$ Apply $\mathbf{u}_{0|k}^*$, set $q_{k+1} := q_{1|k}^*$, $x_{k+1} := x_{1|k}^*$.
14: $\quad$ $I_{k+1} := I(\mathbf{T}_k^*)$.
15: $\quad$ $v_{k+1} := V_{aut,\bar{S}}(q_{I_{k+1}|k}^*)$.
16: $\quad$ **if** $I_{k+1} == I_k$ **then**
17: $\quad\quad$ $v_{k+1} := \min\{v_k - 1, v_{k+1}\}$.
18: $\quad$ **end if**
19: $\quad$ $k := k + 1$.
20: **end while**

The proposed MPC controller is summarized in Algorithm 1, where a set of optimization problems is solved at each time step until a final automaton state is reached (line 2). At each time step, the linear quadratic optimization problem given in line 7 is solved for each automaton path $\mathbf{q} \in \mathbf{P}_{q_k}^{I_k}$ (12) which satisfies the condition given in line 6. Note that if an automaton path does not satisfy the condition given in line 6, the problem given in (13) becomes infeasible. When the loop over $\mathbf{P}_{q_k}^{I_k}$ (line 5) is terminated, the first element of the optimal control sequence $\mathbf{u}_k^*$ is applied and the state (q_{k+1}, x_{k+1}) is computed. Notice that at each time step, either the prediction horizon or the bound used in the terminal constraint (13c) is reduced (lines 14–18).

Assumption 5.2. The length of any satisfying trajectory of system (1) originating at x_0 is lower bounded by N.

Assumption 5.2 is made to simplify the presentation of the following results.

Lemma 5.3. *Suppose that Assumptions 2.2 and 5.2 hold, and there exists $q_0 \in Q^D$ such that $x_0 \in \mathscr{P}_{q_0}$. Then, the optimization problem given in line 7 of Algorithm 1 is feasible for some $\mathbf{q}_0 \in \mathbf{P}_{q_0}^N$ at the initial condition (q_0, x_0).*

Proof. Assumption 2.2 and Definition 4.2 imply that $v_0 = V_{aut,\bar{S}}(q_0) < \infty$. Since $V_{\underline{S}}(q_0, x_0) \leq V_{aut,\underline{S}}(q_0)$ and $V_{aut,\underline{S}}(q_0) \leq V_{aut,\bar{S}}(q_0)$ for all $q_0 \in Q^D$, it holds that $V_{\underline{S}}(q_0, x_0) \leq v_0$.

By Proposition 4.4 and Assumption 5.2, there exist a control sequence $\mathbf{u} = u_0 \dots u_{N-1}$ and an automaton path $\mathbf{q} = q_0 \dots q_N$ such that the value of the potential function $V_{\underline{S}}(\cdot, \cdot)$ strictly decreases along the trajectory $\mathbf{T} = (q_0, x_0) \dots (q_N, x_N)$ generated by $\mathbf{u}$ from the initial condition (q_0, x_0), and hence, $V_{\underline{S}}(q_N, x_N) < V_{\underline{S}}(q_0, x_0) \leq v_0$. As such, the optimization problem is feasible for $\mathbf{q}$. $\square$

Theorem 5.4. *Suppose that Assumptions 2.2 and 5.2 hold, and there exists $q_0 \in Q^D$ such that $x_0 \in \mathscr{P}_{q_0}$. Then:*

(i) *If the optimization problem given in line 7 of Algorithm 1 is feasible for some $\mathbf{q}_k \in \mathbf{P}_{q_k}^{I_k}$ at time k for state (q_k, x_k), then there exists $\mathbf{q}_{k+1} \in \mathbf{P}_{q_{k+1}}^{I_{k+1}}$ such that the problem is feasible for $\mathbf{q}_{k+1}$ and state (q_{k+1}, x_{k+1}).*
(ii) *The trajectory of system (1) produced by the closed-loop system satisfies the specification.*

Proof. (i) Let $\mathbf{T}_k^* = (q_{0|k}^*, x_{0|k}^*) \dots (q_{I_k|k}^*, x_{I_k|k}^*)$ be the trajectory generated by the optimal control sequence $\mathbf{u}^{*_k} = u_{0|k}^* \dots u_{I_k|k}^*$ at step k. From (16), we have that $I_{k+1} \leq I_k$. By Proposition 4.4, there exist a control $u \in \mathbb{U}$ and a state $q' \in Q^D$ such that $x' = Ax_{I_{k+1}|k} + Bu \in \mathscr{P}_{q'}$, $(q_{I_{k+1}|k}, q') \in \rightarrow^D$ and

$$V_{\underline{S}}(q', x') < V_{\underline{S}}(q_{I_{k+1}|k}^*, x_{I_{k+1}|k}^*). \tag{17}$$

By (7) and (9), it holds that

$$V_{\underline{S}}(q_{I_{k+1}|k}^*, x_{I_{k+1}|k}^*) \leq V_{aut,\underline{S}}(q_{I_{k+1}|k}^*) \leq V_{aut,\bar{S}}(q_{I_{k+1}|k}^*). \tag{18}$$

In the case when $I_k = I_{k+1}$, by constraint (13c) we have that $V_{\underline{S}}(q_{I_{k+1}|k}^*, x_{I_{k+1}|k}^*) < v_k$, and as such, by (17) and (18) it holds that

$$V_{\underline{S}}(q', x') < v_k - 1. \tag{19}$$

Eqs. (17)–(19) imply that $V_{\underline{S}}(q', x') < v_{k+1}$. As such, the control sequence $u_{1|k}^* \dots u_{I_k|k}^*, u'$ is a feasible solution of the optimization problem at step $k + 1$ for $q_{1|k}^* \dots q_{I_{k+1}|k}^* q'$.

(ii) We first show that the produced trajectory reaches a final automaton state in finite time. By Lemma 5.3, the optimization problem is feasible at time $k = 0$ for some $\mathbf{q}_0 \in \mathbf{P}_{q_0}$. From Theorem 5.4-(i) it follows that an optimal predicted trajectory $\mathbf{T}_k^*$ exists at each time step until a final state is reached. Let $v^* = \max_{q \in Q^d} V_{aut,\bar{S}}(q)$. By Assumption 2.2 and Definition 4.2, $v^* < \infty$. From lines 1 and 15 of Algorithm 1, it holds that $v_k \leq v^*$. From line 17 of Algorithm 1, we have that $v_{k+1} < v_k$ until a final automaton state appears in a trajectory. The strict decrease and the progress constraint (13c) imply that there exists $k' \leq v_0$ such that the trajectory $\mathbf{T}_{k'}^*$ visits a final automaton state. The optimization horizon at step $k' + 1$ satisfies $I_{k'+1} < N$, and $v_{k'+1} = V_{aut,\bar{S}}(q_{I_{k'+1}|k'}^*) \leq v^*$. By applying the same argument iteratively, we conclude that there exists a time $k'' < Nv^*$ such that $I_{k''} = 1$ and the predicted trajectory

$\mathbf{T}^*_{k''}$ ends in a final automaton state. Therefore, the trajectory produced by the closed-loop system reaches a final automaton state within Nv^* steps. As shown above, the proposed MPC controller produces a finite trajectory $(q_0, x_0)\dots(q_l, x_l)$, $l \le Nv^*$. Next, we show that the projected system trajectory $x_0 \dots x_l$ satisfies Φ. It is assumed that $x_0 \in \mathscr{P}_{q_0}$ and $q_0 \in Q_0^D$. By the definition of $\mathbf{P}_q^{l_k}$, $(q_i, q_{i+1}) \in \to^D$ for all $i = 0, \dots, l-1$ and by the termination condition $q_l \in F^D$. Consequently, $q_0 \dots q_l$ is an accepting automaton run. The constraints of the optimization problem given in line 7 ensure that $x_k \in \mathscr{P}_{q_k}$ for all $i = 0, \dots, l$. Hence, the system trajectory $x_0 \dots x_l$ satisfies the specification. □

Remark 5.5. The presented temporal logic model predictive control approach can be extended to piece-wise affine systems described by

$$x_{k+1} = A_i x_k + B_i u_k + c_i, \quad x_k \in \mathbb{X}_i,\ u_k \in \mathbb{U},\ i = 1, \dots, l, \tag{20}$$

where l is the number of modes (different dynamics), $\mathbb{X}_i$, $i = 1, \dots, l$ provide a polyhedral partition of $\mathbb{X} = \bigcup_{i=1,\dots,l} \mathbb{X}_i \subset \mathbb{R}^n$, such that $\mathbb{X}_i \cap \mathbb{X}_j = \emptyset$ for all $i \neq j$, $\mathbb{U}$ is a polyhedral set in $\mathbb{R}^m$, and A_i, B_i, and c_i are matrices of appropriate sizes. The extension requires a refinement of the initial dual automaton with respect to the particular structure of the state space $\mathbb{X} = \bigcup_{i=1,\dots,l} \mathbb{X}_i$ such that for each state q of the refined automaton there exist $\alpha(q) \in \{1, \dots, l\}$ and $\mathscr{P}_q \subset \mathbb{X}_{\alpha(q)}$. Consequently, during the refinement step, the system dynamics $(A_{\alpha(q)}, B_{\alpha(q)}, c_{\alpha(q)})$ can be used to solve $\mathscr{P}_q - to - \mathscr{P}_{q'}$ control synthesis problems for transitions leaving the state q.

The MPC problem given in (10) still reduces to solving a finite number of QPs, that only depends on the size of the set $\mathbf{P}^N_{q_k}$ due to the fact that only one mode is active for any $\mathscr{P}_q$ in a path. Therefore, once a refined dual automaton is obtained as explained above, Algorithm 1 can be used to control a PWA system by adapting the dynamics in the optimization problem given in (13) as follows:

$$x_{j+1|k} = A_{\alpha(q_{j|k})} x_{j|k} + B_{\alpha(q_{j|k})} u_{j|k} + c_{\alpha(q_{j|k})}, \quad j = 1, \dots, N.$$

6. Candidate potential functions and complexity

In this section, we present candidate potential functions that can be used to implement the proposed MPC controller, and then we analyze the associated complexity.

6.1. Successor function

A successor function $S(\cdot)$ according to Definition 4.2 can easily be constructed by traversing the graph of the automaton starting from the final states. Let Q_T denote the states that are traversed. Initially, set $S(q_f) := q_f$, $q_f \in Q^F$ and $Q_T := Q^F$. Then iteratively choose a state q from the set $Q^D \setminus Q_T$ such that there exist a state $q' \in Q_T$ and $w((q, q')) < \infty$, and set $S(q) := q'$ and $Q_T := Q_T \cup \{q\}$. By Assumption 2.2, this process terminates and Q_T covers Q^D. Note that, while constructing a successor function $S_{SP}(\cdot)$ as outlined above, if q' and the candidate state q are chosen such that

$$(q, q') = \arg \min_{(q,q') \in \{Q^D \setminus Q_T\} \times Q_T \cap \to^D} w((q, q')) + V_{aut,S_{SP}}(q'), \tag{21}$$

then $V_{aut,S_{SP}}(q)$ is the cost of the shortest path from q to a final automaton state. It is clear that for the successor function $S_{SP}(\cdot)$ and any other successor function $S(\cdot)$ it holds that

$$V_{aut,S_{SP}}(q) \le V_{aut,S}(q), \quad \text{for all } q \in Q^D. \tag{22}$$

Hence it was demonstrated how $\underline{S}(\cdot)$ and $\overline{S}(\cdot)$ can be constructed such that (9) holds.

6.2. Control potential function

We present two control potential functions: one based on one step controllable sets and the other one on the feedback controllers that are used to compute the transition weight function $w(\cdot)$.

6.2.1. Potential function based on one step controllable sets

Consider $V_{con,CS} : \bigcup_{(q,q') \in \to^D} \left\{\{(q, q')\} \times \mathscr{P}_q\right\} \longrightarrow \mathbb{Z}_+$:

$$V_{con,CS}((q, q'), x) = \arg\min\{k \in \{1, \dots, w((q, q'))\} \mid x \in \mathscr{R}^k_{qq'}\}, \tag{23}$$

where $\mathscr{R}^k_{qq'}$ denotes the set of states in $\mathscr{P}_q$ that can reach $\mathscr{P}'_q$ in k steps, *i.e.*,

$$\mathscr{R}^k_{qq'} = \begin{cases} \{x \in \mathscr{P}_q \mid \exists u \in \mathbb{U} : Ax + Bu \in \mathscr{P}_{q'}\} & \text{if } k = 1, \\ \{x \in \mathscr{P}_q \mid \exists u \in \mathbb{U} : Ax + Bu \in \mathscr{R}^{k-1}_{qq'}\} & \text{if } k > 1. \end{cases} \tag{24}$$

From (6) and (23), it follows that

$$\mathscr{R}^{k,qq'}_{V_{con,CS}} = \bigcup_{i=1,\dots,k} \mathscr{R}^i_{qq'}. \tag{25}$$

6.2.2. Potential function based on feedback controllers

The following properties are used to define a control potential function with respect to a feedback control law that solves $\mathscr{P}_q - to - \mathscr{P}_{q'}$ control problems. *It is important to note that both $\mathscr{P}_q - to - \mathscr{P}_{q'}$ control methods from* Aydin Gol et al. (2014) *satisfy these properties.* The detailed description of the controllers for solving $\mathscr{P}_q - to - \mathscr{P}_{q'}$ control problems can be found in Section 5 of Aydin Gol et al. (2014).

Property 6.1. *Let $g : \mathscr{P}_q \to \mathbb{U}$ be a feedback control law that solves the $\mathscr{P}_q - to - \mathscr{P}_{q'}$ control problem. Let $k^* = w((q, q'))$. Let $\{x^i_j\}_{j=0,\dots,k^*}$ be the trajectory generated by the feedback control $g(\cdot)$ from the vertex $v^i \in \mathscr{V}(\mathscr{P}_q)$, i.e., $x^i_0 := v^i$. The feedback control $g(\cdot)$ satisfies the following properties:*

(i) *For each vertex $v^i \in \mathscr{V}(\mathscr{P}_q)$:*

$$x^i_j \in \mathscr{P}_q, j = 0, \dots, k^* - 1, \quad \text{and} \quad x^i_{k^*} \in \mathscr{P}_{q'}.$$

(ii) *If $x \in Co(\{x^i_k\}_{i=1,\dots,\#\mathscr{V}(\mathscr{P}_q)})$, $k < k^*$, it holds that*

$$Ax + Bg(x) \in Co(\{x^i_{k+j}\}_{i=1,\dots,\#\mathscr{V}(\mathscr{P}_q)}),$$
for some $1 \le j \le k^* - k$.

Suppose that Property 6.1 holds, and consider $V_{con,FC} : \bigcup_{(q,q') \in \to^D} \left\{\{(q, q')\} \times \mathscr{P}_q\right\} \longrightarrow \mathbb{Z}_+$:

$$V_{con,FC}((q, q'), x) = k^* - \arg\max\{j \in \{0, \dots, k^*\} \mid x \in Co(\{x^i_j\}_{i=1,\dots,\#\mathscr{V}(\mathscr{P}_q)})\}. \tag{26}$$

From (6) and (26), it follows that

$$\mathscr{R}^{k,qq'}_{V_{con,FC}} = \bigcup_{j=1,\dots,k} Co(\{x^i_{w((q,q'))-j}\}_{i=1,\dots,\#\mathscr{V}(\mathscr{P}_{q_{N|k}})}). \tag{27}$$

Note that both $V_{con,CS}(\cdot,\cdot)$ (23) and $V_{con,FC}(\cdot,\cdot)$ (26) are control potential functions according to Definition 4.3. The proofs follow from the definitions of the functions, and hence are omitted here.

It is important to note that $V_{con,CS}((q, q'), x)$ (23) is the number of steps required to reach $\mathscr{P}_{q'}$ from $x \in \mathscr{P}_q$ by system (1). Therefore, for any other control potential function $V_{con}(\cdot,\cdot)$ it holds that

$$V_{con,CS}((q, q'), x) \le V_{con}((q, q'), x), \quad \forall (q, q') \in \to^D,\ x \in \mathscr{P}_q. \tag{28}$$

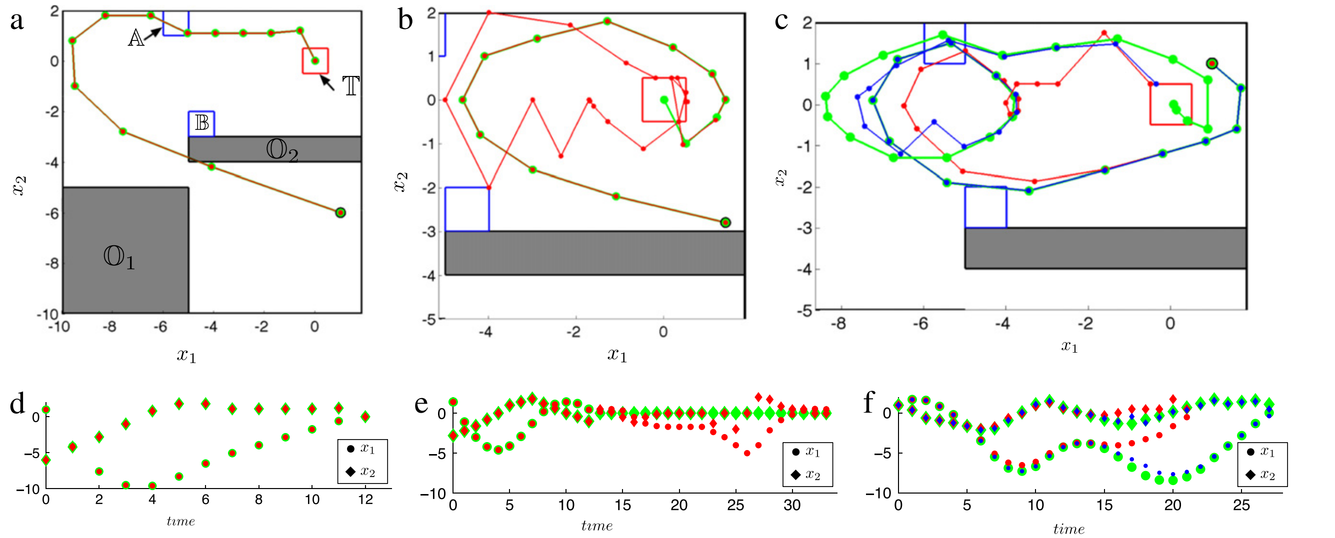


Fig. 2. The reference trajectory (green, satisfying in (a) and (c), and violating in (b)) and the trajectory of the controlled system (red and blue). The initial states are marked by black circles in (a–c). The projections of the trajectories over the first and second dimensions with respect to time are shown in (d–f). (a, d) $N = 2$, total cost = 0. (b, e) $N = 4$, total cost = 58.53. (c, f) $N = 2$, total cost = 61.78 (red) and $N = 5$, total cost = 3.44 (blue). The color codes in (d–f) and (a–c) are the same. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Moreover from (28), it follows that for all $(q, q') \in \rightarrow^D$ with $w((q, q')) \neq \infty$,

$$\mathscr{R}^{k,qq'}_{V_{con}} \subseteq \mathscr{R}^{k,qq'}_{V_{con,CS}}, \quad k = 1, \ldots, w((q, q')). \tag{29}$$

Remark 6.2. The methods proposed for solving Problem 3.1 can be seen as a generalization of the solution presented in Aydin Gol and Lazar (2013), where a potential function $V_{S_{SP}}(\cdot, \cdot)$ was defined from $S_{SP}(\cdot)$ and $V_{con,FC}(\cdot, \cdot)$. The function $V_{S_{SP}}(\cdot, \cdot)$ was used to compute potentials of $(q_{N|k}, x_{N|k})$ and $(q^*_{N|k-1}, x^*_{N|k-1})$, which was used in the progress constraint, *i.e.*, $v_k = V_{S_{SP}}(q^*_{N|k-1}, x^*_{N|k-1})$. Therefore, when $V_{con,CS}(\cdot, \cdot)$, $S_{SP}(\cdot)$ and any successor function $S(\cdot)$ are used for $V_{con}(\cdot, \cdot)$, $\underline{S}(\cdot)$, and $\overline{S}(\cdot)$, respectively, the MPC problem given in (13) is a relaxation of the MPC problem presented in Aydin Gol and Lazar (2013).

6.3. Complexity

The complexity of the presented MPC controller is characterized by the number of quadratic programs solved at each iteration. The optimal solution of the MPC problem given in (10) is found by solving the optimization problem given in (13) for each $\mathbf{q}_k \in \mathbf{P}^N_{q_k}$. The number of paths of length d originating at a node of a graph is upper bounded by b^d, where b is the branching factor of the graph, *i.e.*, the maximum number of outgoing edges from a node. Therefore, the size of $\mathbf{P}^N_{q_k}$ is upper bounded by b^N. The number of quadratic optimization problems solved to find the optimal solution of (13) depends on the representation of the terminal constraint set $\mathscr{R}^{\cdot,\cdot\cdot}_{\cdot}$ from (15), and therefore the choice of the control potential function.

Let $w^* = \max_{(q,q') \in \rightarrow^D}\{w((q, q')) < \infty\}$. It is clear that $\mathscr{R}^{k,qq'}_{V_{con,CS}}$ (25) and $\mathscr{R}^{k,qq'}_{V_{con,FC}}$ (27) are represented as unions of at most $w^* - 1$ polytopes. Therefore, for the control potential functions $V_{con,CS}(\cdot, \cdot)$ and $V_{con,FC}(\cdot, \cdot)$, the optimal solution of the MPC problem (10) can be found by solving at most $b^N w^*$ quadratic programs.

If $\mathscr{P}_q$, $\mathscr{P}_{q'}$ and $\mathbb{U}$ are polytopes, $\mathscr{R}^k_{qq'}$ (24) is also a polytope and can be computed via orthogonal projection. On the other hand, the computation of the sets $Co(\{x^i_{w((q,q'))-j}\}_{i=1,\ldots,\#\mathscr{V}(\mathscr{P}_{q_{N|k}})})$ from (27) requires to store the corresponding feedback control law $g(\cdot)$, to compute the vertex trajectories and to perform a convex hull operation (see (27)). Note that for each transition $(q, q') \in \rightarrow^D$ with $w((q, q')) \neq \infty$, the polyhedral sets $\{\mathscr{R}^k_{qq'}\}_{k=1,\ldots,w((q,q'))}$ and $\{Co(\{x^i_k\}_{i=1,\ldots,\#\mathscr{V}(\mathscr{P}_q)})\}_{k=1,\ldots,w((q,q'))}$ can be precomputed and stored for faster computations of the corresponding terminal constraint sets.

7. Case study

Consider the double integrator dynamics and the specification given in Example 3.2. The cost function is defined as in (4) with $L_N = L = 0.5I_2$ and $R = 0.2$. The successor function $S_{SP}(\cdot)$ and the control potential function $V_{con,CS}(\cdot, \cdot)$ which are defined in Section 6 are used in Algorithm 1 for $\underline{S}(\cdot)$ and $V_{con}(\cdot, \cdot)$. The successor function $\overline{S}(\cdot)$ is constructed by traversing the graph of the automaton $\mathscr{A}^D$ as explained in Section 6.1 with a slight modification of the rule given in (21), *i.e.*, the states q, q', with $w((q, q')) < \infty$ that maximize the right hand side of (21) are chosen.

We use both satisfying and violating reference trajectories. The reference trajectories $\mathbf{x}^{r_i}$, $i = 1, 2, 3$ are generated by the reference control sequences $\mathbf{u}^{r_i}$, $i = 1, 2, 3$ from the initial conditions $x^{r_1}_0 = \begin{bmatrix} 1 \\ -6 \end{bmatrix}$, $x^{r_2}_0 = \begin{bmatrix} 1.4 \\ -2.8 \end{bmatrix}$ and $x^{r_3}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, where

$$\mathbf{u}^{r_1} = 1.8, 1.4, 1.8, 1.8, 1, 0, -0.7, 0, 0, 0, 0.1, -1.2, 0.$$

$$\mathbf{u}^{r_2} = 0.6, 0.6, 0.8, 0.8, 1, 0.4, 0.4, -0.6, -0.6, -0.6, -0.4, -0.6, 1.$$

$$\begin{aligned}\mathbf{u}^{r_3} = \; & -0.6, -1, -0.3, -0.3, -0.4, -0.5, 0.2, 1, 1, 1, 0.4, -0.8, \\ & -0.5, -0.5, -0.5, -0.5, 0, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, \\ & -0.5, 0.2, 0.2, -0.5, -0.5, -1.2, 0.2, 0.2667, 0.1333.\end{aligned}$$

The reference trajectories $\mathbf{x}^{r_1}$ and $\mathbf{x}^{r_3}$ satisfy the specification, while $\mathbf{x}^{r_2}$ violates it. The reference trajectories and the corresponding simulated trajectories generated by Algorithm 1 are shown in Fig. 2. $\mathbf{x}^{r_1}$ and $\mathbf{u}^{r_1}$ were used to generate Fig. 2(a–b) from Aydin Gol and Lazar (2013). As shown in Fig. 2(a), the controlled trajectory $\mathbf{x}^1$ follows $\mathbf{x}^{r_1}$ until the specification is satisfied, which was not possible in Aydin Gol and Lazar (2013). As shown in Fig. 2(b), the controlled trajectory $\mathbf{x}^2$ follows the reference trajectory $\mathbf{x}^{r_2}$ only for the first 12 steps, then the distance between them increases as the controlled trajectory visits region B to satisfy the specification. Even though reference trajectory $\mathbf{x}^{r_3}$ satisfies the specification, as shown in Fig. 2(c) the controlled trajectories for optimization

Table 1
Average and maximum (in parentheses) computation times (s).

	$N = 2$	$N = 3$	$N = 4$	$N = 5$
$\mathbf{x}^{r_1}$	0.009 (0.018)	0.01 (0.02)	0.01 (0.02)	0.01 (0.02)
$\mathbf{x}^{r_2}$	0.08 (0.47)	0.41 (1.67)	3.77 (10.4)	28.8 (80.1)
$\mathbf{x}^{r_3}$	0.04 (0.29)	0.12 (0.62)	0.63 (3.25)	5.5 (30.5)

horizons $N = 2$ and $N = 5$ cannot exactly follow $\mathbf{x}^{r_3}$ due to the progress constraint.

As discussed in Section 6.3, the number of QPs solved to generate the closed-loop trajectories increases exponentially with the length of the optimization horizon. The average and the maximum amount of time spent at each iteration of Algorithm 1 is shown in Table 1.

8. Conclusions

We proposed a framework for designing model predictive controllers for linear (and piece-wise affine) discrete-time systems subject to specifications expressed in a fragment of LTL. Conditions for guaranteeing recursive feasibility and formula satisfaction were derived based on so-called potential functions. Implementation of the MPC algorithm required solving a finite number of QP problems.

References

Baier, C., & Katoen, J. (2008). *Principles of model checking*. The MIT Press.

Bhatia, A., Kavraki, L. E., & Vardi, M. Y. (2010). Motion planning with hybrid dynamics and temporal goals. In *IEEE conf. on decision and control. Atlanta, GA* (pp. 1108–1115).

Ding, X. C., Lazar, M., & Belta, C. (2012). Receding horizon temporal logic control for finite deterministic systems. In *American control conference. Montreal, Canada* (pp. 715 -720).

Ding, X. C., Smith, S. L., Belta, C., & Rus, D. (2011). Mdp optimal control under temporal logic constraints. In *IEEE conf. on decision and control. Orlando, FL* (pp. 532–538).

Gazit, H. K., Fainekos, G., & Pappas, G. J. (2007). Where's Waldo? Sensor-based temporal logic motion planning. In *IEEE conference on robotics and automation. Rome, Italy* (pp. 3116–3121).

Girard, A. (2010). Synthesis using approximately bisimilar abstractions: state-feedback controllers for safety specifications. In *Hybrid systems: computation and control* (pp. 111–120). ACM.

Aydin Gol, E., & Lazar, M. (2013). Temporal logic model predictive control for discrete-time systems. In *Hybrid systems: computation and control* (pp. 343–352). ACM.

Aydin Gol, E., Lazar, M., & Belta, C. (2014). Language-guided controller synthesis for linear systems. *IEEE Transactions on Automatic Control*, *59*(5), 1163–1176.

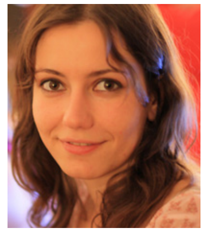
Kupferman, O., & Vardi, M. Y. (2001). Model checking of safety properties. *Formal Methods in System Design*, *19*, 291–314.

Sloth, C., & Wisniewski, R. (2013). Complete abstractions of dynamical systems by timed automata. *Nonlinear Analysis. Hybrid Systems*, *7*(1), 80–100.

Tabuada, P., & Pappas, G. (2003). Model checking LTL over controllable linear systems is decidable. In O. Maler and A. Pnueli (Eds.), (Vol. 2623). Springer-Verlag.

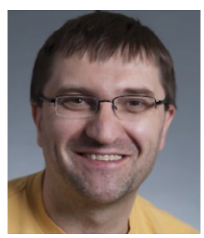
Wongpiromsarn, T., Topcu, U., & Murray, R. M. (2009). Receding horizon temporal logic planning for dynamical systems. In *IEEE conf. on decision and control. Shanghai, China* (pp. 5997–6004).

Yordanov, B., Tumova, J., Belta, C., Cerna, I., & Barnat, J. (2012). Temporal logic control of discrete-time piecewise affine systems. *IEEE Transactions on Automatic Control*, *57*(6), 1491–1504.



Ebru Aydin Gol received her B.Sc. degree in Computer Engineering from Orta Dogu Teknik Universitesi, Ankara, Turkey, in 2008, M.Sc. degree in Computer Science from Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, in 2010 and Ph.D. degree in Systems Engineering from Boston University, Boston, MA, USA in 2014. She has been a Site Reliability Engineer at Google since 2014. Her research interests include verification and control of dynamical systems, optimal control, and synthetic biology.

Mircea Lazar (born in Iasi, Romania, 1978) received his M.Sc. and Ph.D. in Control Engineering from the Technical University "Gh. Asachi" of Iasi, Romania (2002) and the Eindhoven University of Technology, The Netherlands (2006), respectively. For the Ph.D. thesis "Model predictive control of hybrid systems: Stability and robustness" he received the EECI (European Embedded Control Institute) Ph.D. award. Since 2006 he has been an Assistant Professor in the Control Systems group of the Electrical Engineering Faculty at the Eindhoven University of Technology. He is currently serving as chair of the IEEE CSS Technical Committee on Hybrid Systems and as a member of the IFAC Technical Committee 2.3 Non-linear control systems. His research interests lie in stability theory, Lyapunov methods, hybrid systems and model predictive control.



Calin Belta is a Professor in the Department of Mechanical Engineering, Department of Electrical and Computer Engineering, and the Division of Systems Engineering at Boston University, where he is also affiliated with the Center for Information and Systems Engineering (CISE) and the Bioinformatics Program. His research focuses on dynamics and control theory, with particular emphasis on hybrid and cyber–physical systems, formal synthesis and verification, and applications in robotics and systems biology. He is a Senior Member of the IEEE and an Associate Editor for the SIAM Journal on Control and Optimization (SICON) and the IEEE Transactions on Automatic Control. He received the Air Force Office of Scientific Research Young Investigator Award and the National Science Foundation CAREER Award.