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 $\diamondsuit$ (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 $\diamondsuit\ \phi_1$ 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:

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

$L, L_N \in \mathbb{R}^{n\times n}$ 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).