1. Introduction

In recent years, there has been an increasing interest in formal synthesis of control strategies for dynamical systems (Bhatia, Kavraki, & Vardy, 2010; Gazit, Fainekos, & Pappas, 2007; Girard, 2010; Aydin Gol, Lazar, & Bela, 2014; Sloth & Wisniewski, 2013; Tabuada & Pappas, 2003; Wongpiromsarn, Topcu, & Murray, 2009; Yordanov, Tunova, Bela, 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, & Bela, 2012; Ding, Smith, Bela, & 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 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
The procedure involved construction of a finite state automaton 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

$$\omega^D = (Q^D, \to^D, 2^D, \tau^D, Q_0^D, F^D),$$

where $Q^D$ is a finite set of states, $\to^D \subseteq Q^D \times Q^D$ is a set of transitions, $2^D$ is a set of symbols, $\tau^D : Q^D \to 2^D$ 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 $R_q \subseteq X$.

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 from $Q_0$ reach $Q_q$ within $\omega^D$ steps while remaining in $R_q$ until they reach $R_{q'}$. An accepting run $r_{\omega^D}$ of a dual automaton is a sequence of states $r_{\omega^D} = q_0, q_1, \ldots, q_d$ 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 $r_{\omega^D}$ defines a word $\sigma = q_0 \ldots q_d$ over $Q^D$.

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 from $Q_0$ reach $Q_q$ within $\omega^D$ steps while remaining in $R_q$ until they reach $R_{q'}$. An accepting run $r_{\omega^D}$ of a dual automaton is a sequence of states $r_{\omega^D} = q_0, q_1, \ldots, q_d$ 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 $r_{\omega^D}$ defines a word $\sigma = q_0 \ldots q_d$ over $Q^D$.

**Assumption 2.2.** For any $q_0 \in Q^D$ there exists an automaton path $q_0, q_1, \ldots, q_d \in \mathbb{Z}_+$ 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 $P = \{p_i\}_{i=0,\ldots,N}$, $i \geq 1$, given as linear inequalities over the system states. Let $x_k^1, \ldots, x_k^N$ denote a reference trajectory and a reference control sequence, respectively. We assume that, for some $N$, at time $k$ the reference trajectory of length $N + 1$, $x_k^1, \ldots, x_k^N$, and the reference control sequence of length $N$, $u_k^1, \ldots, u_k^{N-1}$, are known. At time $k \in \mathbb{X}$, the cost of a finite trajectory $x_k, x_{k+1}, \ldots, x_{k+N}$ originating at $x_k$ and generated by the control sequence $u_k, \ldots, u_{k+1}$ is defined with respect to the available reference trajectory and control sequence as follows:

$$C(x_k, u_k) = (x_{k+N} - x_{k+N})^T L N (x_{k+N} - x_{k+N})$$

$$+ \sum_{i=0}^{N-1} \left( (x_{k+i} - x_{k+i})^T L (x_{k+i} - x_{k+i}) + (u_{k+i} - u_{k+i})^T R (u_{k+i} - u_{k+i}) \right).$$

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

**Problem 3.1.** Given an scLTL formula $\Phi$ 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 $\Phi$ while minimizing the cost (4).

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 from $Q_0$ reach $Q_q$ within $\omega^D$ steps while remaining in $R_q$ until they reach $R_{q'}$. An accepting run $r_{\omega^D}$ of a dual automaton is a sequence of states $r_{\omega^D} = q_0, q_1, \ldots, q_d$ 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 $r_{\omega^D}$ defines a word $\sigma = q_0 \ldots q_d$ over $Q^D$. Therefore, any system trajectory $x_0, x_1, \ldots, x_d$ that follows a sequence of regions $Q_0, Q_1, \ldots, Q_d$, defined by an accepting automaton run $r_{\omega^D} = q_0, \ldots, q_d$ satisfies the specification. Furthermore, any satisfying trajectory of system (1) follows a sequence of polyhedral sets defined by an accepting run of $\omega^D$.
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 $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. The control constraint set is $U = \{u \mid -2 \leq u \leq 2\}$ and the initial state is $x_0 = \begin{bmatrix} 1.4 \\ -2.8 \end{bmatrix}$. The specification is to visit region $A$ or region $B$, and then the target region $T$, while always avoiding obstacles $O_1$ and $O_2$, and staying inside a safe region $X = \{x \mid -10 \leq x_1 \leq 1.85, -10 \leq x_2 \leq 2\}$. These polygonal 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:

$$
\Phi_1 = ((p_0 \lor p_1 \lor p_2 \lor p_3 \lor \neg(p_4 \land p_5) \lor \neg(p_5 \land p_6 \land \neg(p_7)) \lor (p_8 \land p_9 \land p_{10} \land p_{11})) \lor ((p_5 \land p_{12} \land p_{13}) \lor (p_5 \land p_{14} \land p_{15})))
$$

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 $\mathcal{X}_0^\Phi$ 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 $V : \bigcup_{q \in Q^D} \{q \times \mathcal{P}_q\} \to \mathbb{Z}_+$ 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 $(q, x) \in \bigcup_{q \in Q^D} \{q \times \mathcal{P}_q\}$, it holds that if $V(q, x) \neq 0$ and $V(q, x) \neq \infty$, then there exists a control $u \in U$ such that $x' = Ax + Bu$, $x' \in \mathcal{P}_q(q, q') \to 0^D$, 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) $(q, S(q)) \in 0^D$ 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 $S^0(q) = q$, and $S^d(q) = q$.

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 $\alpha^D$ are fixed points of the function $S(\cdot)$.

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

$$
V_{out, S}(q) = \begin{cases} 0 & \text{if } q \in F^D; \\ w((q, S(q))) + V_{out, S}(S(q)) & \text{otherwise}. 
\end{cases}
$$

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

**Definition 4.3.** A function $V_{con} : \bigcup_{(q, q') \in 0^D \times \mathcal{P}_q} \{(q, q') \times \mathcal{P}_q\} \to \mathbb{Z}_+ \setminus \{0\}$ is called a control potential function for a system (1) and a dual automaton (3) if it satisfies:

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

We define a set $\mathcal{A}_{con}^{k,q'} \subseteq \mathcal{P}_q$, $k \in \mathbb{Z}_+$, for a given control potential function $V_{con}(\cdot, \cdot)$ and a transition $(q, q') \in 0^D$ as

$$
\mathcal{A}_{con}^{k,q'} = \{x \in \mathcal{P}_q \mid V_{con}((q, q'), x) \leq k\}.
$$

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 a given successor function $S(\cdot)$, control potential function $V_{con}(\cdot, \cdot)$, and automaton potential function $V_{out, S}(\cdot)$ as

$$
V(q, x) = \begin{cases} 0 & \text{if } q \in F^D; \\ V_{con}((q, S(q)), x) + V_{out, S}(S(q)) & \text{otherwise}. 
\end{cases}
$$

Informally, the candidate potential function (7) at $(q, x)$, $q \in Q^D$, $x \in \mathcal{P}_q$ is defined as an upper bound for the time required to reach $\mathcal{P}_{S(q)}$ from $x$ by applying the corresponding polytope-topolytope feedback controllers along an automaton path defined by the successor function $S(q) \ldots S^d(q)$, 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 $V_N(\cdot, \cdot)$. To prove that the function satisfies 4.1-(ii), we consider two cases: $V_{\text{con}}(q, S(q)), x) = 1$ and $V_{\text{con}}(q, S(q)), x) > 1$.

Note that $V_{\text{con}}(q, S(q)), x) = 1$ and property 4.3-(ii) imply that there exists $u \in U$ such that $Ax + Bu \in \mathcal{P}_2(q)$. As such, in the case $V_{\text{con}}(q, S(q)), x) = 1$, the claim holds as $V_N(q, S(q)) \leq V_{\text{con}}(q, S(q))$ for all $x \in \mathcal{P}_2(q)$. In the case when $V_{\text{con}}(q, S(q)), x) > 1$, from 4.2-(i) and 4.3-(i), it holds that $V_{\text{con}}(q, S(q)), x) \neq \infty$. By property 4.3-(ii), there exists $u \in U$ such that one of the following holds:

(a) $Ax + Bu \in \mathcal{P}_2$,
(b) $Ax + Bu \in \mathcal{P}_2$ and $V_{\text{con}}(q, S(q)), Ax + Bu) < V_{\text{con}}(q, S(q)), x)$.  

The claim is true for (a) by the same argument given for the $V_{\text{con}}(q, S(q)), \cdot) = 1$ case. For (b), the claim holds trivially by the definition of the function $V_N(\cdot, \cdot)$ (7).

5. MPC strategy

In this section, we present a MPC scheme to solve Problem 3.1 for a given dual automaton $A^D = (Q^D, \rightarrow^D, \bar{x}^D, \bar{u}^D, \mathcal{P}^D, \mathcal{F}^D)$ and a transition weight function $w : \rightarrow^D \rightarrow \mathbb{Z}_+$. We formulate an MPC optimization problem over $\bigcup_{q \in Q^D} \{q \times \mathcal{P}_2\}$ to be solved “online” at each time step.

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

(i) for each $k = 0, \ldots, N - 1$ there exists $u_k \in U$ such that $x_{k+1} = Ax_k + Bu_k$,
(ii) $x_k \in \mathcal{P}_2$, for all $k = 0, \ldots, N$,
(iii) $(q_k, q_{k+1}) \rightarrow^{D}$, for all $k = 0, \ldots, N - 1$.

The definition of an automaton-enabled trajectory implies that the projection $\gamma^{D}(T) = q_0 \ldots q_N$ of the trajectory onto the automaton states is an automaton path and the projection $\gamma(x)(T) = x_0 \ldots x_N$ 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 $A^D$ from Section 2 and Definition 5.1 guarantees that for any satisfying trajectory $x = x_0 \ldots x_d, \quad d \in \mathbb{Z}_+$, of system (1) originating at $x_0$, there exists an automaton-enabled trajectory $T$ such that $\gamma^{D}(T) = x$ and $\gamma(x)(T)$ is an accepting run of $A^D$. Therefore, in the MPC controller design, we restrict our attention to the control sequences that generate automaton-enabled trajectories. We use $U_N(q, x)$ 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 $T_k = (q_0, x_0) \ldots (q_N, x_N)$ to denote a predicted automaton-enabled trajectory originating at $(q_0, x_0)$, i.e., $q_0 = q_0, x_0 = x_0$, at time $k \in \mathbb{Z}_+$. At each time-step $k \in \mathbb{Z}_+$, we solve an optimization problem over $U_N(q_0, x_0)$ and find the optimal automaton-enabled trajectory $T_k^* = (q_0, x_0) \ldots (q_N, x_N)$ generated by the optimal control sequence $u_k \in U_N(q_0, x_0)$, until a final automaton state is reached, i.e., $q_N \in \mathcal{P}_2$. As the first control of the optimal control sequence is applied, we have the following relation:

$x_{k+1} = x_{k+1}^* \quad k = 0, 1, 2, \ldots, \quad (8)$

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 $V_{\text{con}}(\cdot, \cdot)$ is a control potential function (Definition 4.3), $\mathcal{S}(\cdot)$ and $\mathcal{S}(\cdot)$ are successor functions (Definition 4.2) such that the corresponding automaton potential functions $V_{\text{aut}}(\cdot)$ and $V_{\text{aut}}(\cdot)$ as defined in (5) satisfy:

$V_{\text{aut}}(q) \leq V_{\text{aut}}(q), \quad \forall q \in Q^D.$

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

$\min_{u_k \in U(q, x)} C(q_k, u_k), \quad \text{subject to } V_k(q_{Nk}, x_{Nk}) < v_k,$

where $v_k \in \mathbb{Z}_+$, $v_0 = V_{\text{aut}}(q_0)$ and for $k \geq 1, v_k$ is defined by

$v_k = \min(v_{k-1} - 1, V_{\text{aut}}(q_{Nk-1})).$

For $k \geq 1$, the optimal predicted trajectory obtained at the previous time step and $v_{k-1}$ 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 $v_k$ steps by following the sequence of polyhedra defined by the successor function $\mathcal{S}(\cdot)$. It is important to note that the bound $v_k$ is computed according to $\mathcal{S}(\cdot)$, and by (7) and (9), $V_{\text{aut}}(q_{Nk-1})$ is an upper bound on $V_k(q_{Nk-1}, x_{Nk-1})$. As the definition of the bound $v_k$ implies that $v_{k-1} < v_k$, 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 $q_0$, i.e.,

$P_{N_0} = \{q_0\ldots q_{Nk} \mid q_0 := q_0, (q_0, q_{Nk}) \in A^D, \quad i = 0, \ldots, N - 1\}.$

Since $Q^D$ and $\rightarrow^D$ are finite sets, $P_{N_0}$ is a finite set. The definition of an automaton-enabled trajectory $T_k$ of horizon $N$ (Definition 5.1) implies that $\gamma^{D}(T_k) \in P_{N_0}$ for any trajectory that can be produced by a control sequence from the set $U_0(q_0, x_0)$. Given a finite automaton path $q_k \in P_{N_0}$, let $U_k(q_0, x_0)$ denote the set of all control sequences that produce an automaton-enabled trajectory $T_k$ with $\gamma^{D}(T_k) = q_k$. Essentially, $U_k(q_0, x_0)$ is the set of all control sequences that produce trajectories of system (1) that originate at $q_0$ and follow the sequence of polyhedra defined by $q_k$. Then, it is straightforward to see that $U_k(q_0, x_0) = \bigcup_{q_k \in P_{N_0}} U_k(q_0, x_0)$. Consider a path $q_k = q_0 \ldots q_{Nk} \in P_{N_0}$, and the following optimization problem in the variables $u_k = u_0, \ldots, u_{Nk-1}$:

$\min C(q_k, u_k), \quad$ subject to

$x_{ik} \in \mathcal{P}_2$, \quad i = 1, \ldots, N, \quad (13a)$

$u_{ik} \in U, \quad i = 0, \ldots, N - 1, \quad (13b)$

$V_k(q_{Nk}, x_{Nk}) < v_k,$

where $v_k$ is as defined in (11). The set of control sequences that satisfy constraints (13a) and (13b) is $U_k(q_0, x_0)$, 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 $q_k \in P_{N_0}$.

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

$V_{\text{con}}((q_{Nk}, (q_{Nk})), x_{Nk}) < v_k - V_{\text{aut}}(q_{Nk}), \quad (14)$

Let $\tilde{k} := v_k - 1 - V_{\text{aut}}(q_{Nk})$. If $\tilde{k} \geq w(q_{Nk}, (q_{Nk})), \quad$ then the inequality given in (14) is trivially satisfied for all $x_{Nk} \in \mathcal{P}_2(q_{Nk})$. If,
however $k < w(q_{Nk}, s(q_{Nk}))$, then the inequality is satisfied if
\[ x_{Nk} \in \mathcal{A}_{\text{con}}^{k(q_{0}S(q_{Nk}))}. \]  
(15)

As such, if the set $\mathcal{A}_{\text{con}}^{k}(q_{0}S(q_{Nk}))$ (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 $\mathcal{R}$, where $q_{t} \in P^{D}$, 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, $k - 1$, is used. The following function is used to determine the prediction horizon for a given trajectory $T_{k} = (q_{0:k}, x_{0:k}) \ldots (q_{Nk}, x_{Nk})$:

\[ I(T_{k}) = \begin{cases} I_{k} & \text{if } 0 < V_{k}(q_{0:k}, x_{0:k}), \quad \forall i = 0, \ldots, I_{k} \\ V_{0} & \text{if } 0 \leq V_{k}(q_{0:k}, x_{0:k}), \quad \forall i = 0, \ldots, j - 1, \\ 0 & \text{if } j = 0, \\ \end{cases} 
(16)

Algorithm 1: Automaton-guided MPC

Input: $A^{D} = (Q^{D}, \rightarrow^{D}, T^{D}, \tau^{D}, q_{0}^{D}, F^{D})$, transition weight function $w : \rightarrow^{D} \rightarrow \mathbb{R}$, an initial condition $x_{0} \in \mathcal{X}_{\text{ref}}$ for some $q_{0} \in Q^{D}$, MPC horizon $N$, and functions $V_{\text{con}}(\cdot, \cdot), S(\cdot)$ and $\bar{S}(\cdot)$.  
1. Set $k = 0$, $I_{k} = N$, $V_{0} := V_{\text{aut}}(q_{0})$. (Initialization)
2. while $q_{k} \notin F^{D}$ do
3. OptCost = $\infty$, $u^{*} = \emptyset$, $T_{k}^{*} = \emptyset$.
4. Compute $P_{k}^{0}$.
5. for all $q = q_{0:k} \ldots q_{Nk} \in P_{k}^{0}$ do
6. if $V_{\text{aut}}(q_{(q_{0:k}))} < v_{k} - 1$ then
7. $c = \min_{u_{k} \in u_{k}} \mathbb{R}^{-} u_{k-1} g_{k}(X_{k}, U_{k})$ subject to
8. \[ (13a), (13b) \text{ and } (13c) \]
9. if $c < \text{OptCost}$ then
10. OptCost := $c$, set $u_{k}^{*}$ and $T_{k}^{*}$ with respect to the solution of the optimization problem (line 7).
11. end if
12. end for
13. Apply $u_{k-1}^{*}$, Set $q_{k+1} := q_{0:k}^{*}, x_{k+1} := x_{1:k}^{*}$
14. $I_{k+1} := I(T_{k}^{*})$
15. $u_{k+1}^{*} := V_{\text{aut}}(q_{k+1}^{*})$
16. if $I_{k+1} = I_{k}$ then
17. $v_{k+1} := \min_{v_{k} \in \mathbb{R}^{-}} (v_{k} - 1, v_{k+1})$
18. end if
19. $k := k + 1$.
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 $q \in P_{k}^{0}$ (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 $P_{k}^{0}$ (line 5) is terminated, the first element of the optimal control sequence $u_{k}^{*}$ 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 $x_{0} \in \mathcal{X}_{\text{ref}}$. Then, the optimization problem given in line 7 of Algorithm 1 is feasible for some $q_{0} \in P_{k}^{0}$ at the initial condition $(q_{0}, x_{0})$.

**Proof.** Assumption 2.2 and Definition 4.2 imply that $v_{0} = V_{\text{aut}}(q_{0}) < \infty$. Since $V_{2}(q_{0}, x_{0}) \leq V_{\text{aut}}(q_{0})$ and $V_{\text{aut}}(q_{0}) \leq V_{\text{aut}}(q_{0})$ for all $q_{0} \in Q^{D}$, it holds that $V_{2}(q_{0}, x_{0}) \leq v_{0}$.

By Proposition 4.4 and Assumption 5.2, there exist a control sequence $u = u_{0} \ldots u_{N-1}$ and an automaton path $q = q_{0} \ldots q_{N}$ such that the value of the potential function $V_{2}(\cdot, \cdot)$ strictly decreases along the trajectory $T = (q_{0}, x_{0}) \ldots (q_{N}, x_{N})$ generated by $u$ from the initial condition $(q_{0}, x_{0})$, and hence, $V_{2}(q_{0}, x_{0}) < V_{2}(q_{N}, x_{N})$. As such, the optimization problem is feasible for $q_{0}$. □

**Theorem 5.4.** Suppose that Assumptions 2.2 and 5.2 hold, and there exists $q_{0} \in Q^{D}$ such that $x_{0} \in \mathcal{X}_{\text{ref}}$. Then:

(i) If the optimization problem given in line 7 of Algorithm 1 is feasible for some $q_{0} \in P_{k}^{0}$ at time $k$ for state $(q_{k}, x_{k})$, then there exists $q_{k+1} \in P_{k+1}$ such that the problem is feasible for $q_{k+1}$ 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 $T_{k}^{*} = (q_{0:k}, x_{0:k}) \ldots (q_{Nk}, x_{Nk})$ be the trajectory generated by the optimal control sequence $u_{k}^{*} = u_{0:k}^{*} \ldots u_{N-1}^{*}$ at step $k$. From (16), we have that $I_{k+1} \leq I_{k}$. By Proposition 4.4, there exist a control $u \in U$ and a state $q' \in Q^{D}$ such that $x' = A_{0} x_{k+1} + B_{u} (q_{k+1}, q') \in \mathcal{D}$, and $V_{2}(x', q') < V_{2}(q_{0:k+1}, x_{0:k+1})$. By (7) and (9), it holds that

\[ V_{2}(q_{0:k+1}, x_{0:k+1}) \leq V_{\text{aut}}(q_{0:k+1}) \leq V_{\text{aut}}(q_{k+1}). \]  
(18)

In the case when $k_{k} = I_{k+1}$, by constraint (13c) we have that $V_{2}(q_{0:k+1}, x_{0:k+1}) < v_{k+1}$, and as such, by (17) and (18) it holds that $V_{2}(q', x') < v_{k+1} - 1$. (19)

Eqs. (17)–(19) imply that $V_{2}(q', x') < v_{k+1}$. As such, the control sequence $u_{0}^{*} \ldots u_{N-1}^{*}$ is a feasible solution of the optimization problem at step $k + 1$ for $q_{0:k}^{*} \ldots q_{Nk}^{*}$. (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 $q_{0} \in P_{k}^{0}$. From Theorem 5.4-(i) it follows that an optimal predicted trajectory $T_{k}^{*}$ exists at each time step until a final state is reached. Let $v' = \max_{q_{0} \in P_{k}^{0}} V_{\text{aut}}(q_{0})$. 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 $v_{k+1} \leq v_{k}$ until a final automaton state appears in a trajectory. The strict decrease and the progress constraint (13c) imply that there exists $k \leq v_{k}$ such that the trajectory $T_{k}^{*}$ visits a final automaton state. The optimization horizon at step $k + 1$ satisfies $I_{k+1} < N$, and $v_{k+1} = V_{\text{aut}}(q_{k+1}^{*}) < v'$. By applying the same argument iteratively, we conclude that there exists a time $k' < N v' \text{ such that } I_{k'} = 1$ and the predicted trajectory
$T^v_\phi$ ends in a final automaton state. Therefore, the trajectory projected by the closed-loop system reaches a final automaton state within $N v^* \phi$ steps. As shown above, the proposed MPC controller produces a finite trajectory $(q_0, x_0, \ldots, (q_i, x_i), i \leq N v^* \phi$. Next, we show that the projected system trajectory $x_0, \ldots, x_i$ satisfies $\Phi$. It is assumed that $x_0 \in P^0_{\Phi_0}$ and $q_0 \in Q^0$. By the definition of $P^0_{\Phi_0}$, $(q_i, q_{i+1}) \rightarrow D$ for all $i = 0, \ldots, l - 1$ and by the termination condition $q_l \in D^\Phi$. Consequently, $q_0, \ldots, q_l$ is an accepting automaton run. The constraints of the optimization problem given in line 7 ensure that $x_i \in P^0_{\Phi_0}$ for all $i = 0, \ldots, l$. Hence, the system trajectory $x_0, \ldots, x_l$ satisfies the specification. \Box

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

$$x_{j+1} = A x_j + B u_j + c, \quad x_j \in X_i, \quad u_j \in U, \quad i, j = 1, \ldots, l,$$

where $l$ is the number of modes (different dynamics), $X_i$ is a polyhedral partition of $X = \bigcup_{i=1}^{l} X_i \subset \mathbb{R}^n$, such that $X_i \cap X_j = \emptyset$ for all $i \neq j$. $U$ 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 $Q$ and the candidate structure of the state space $X = \bigcup_{i=1}^{l} X_i$ such that for each state $x \in X_i$, the there exist $a(x) \in \{1, \ldots, l\}$ and $Q^0_i \subset X_i$. Consequently, during the refinement step, the system dynamics $(A_{x_i}(x), B_{x_i}(x), c_{x_i}(x))$ can be used to solve $P^0_{\Phi_0}$ to $P^0_{\Phi_0}$ 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 $P^0_{\Phi_0}$ due to the fact that only one mode is active for $P^0_{\Phi_0}$ 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:

$$x_{j+1,k} = A_{u(q,k)} x_{j,k} + B_{u(q,k)} u_{j,k} + c_{u(q,k)}, \quad j = 1, \ldots, N.$$

### 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 $Q_{\Phi}$ denote the states that are traversed. Initially, set $S(q_0) := q_0, q_r \in Q^F$ and $Q_x := Q^F$. Then iteratively choose a state $q'$ from the set $Q^D \setminus Q_x$ such that there exist a state $q' \in Q_{\Phi}$ and $w((q, q')) < \infty$, and set $S(q) := q'$ and $Q_x := Q_x \cup \{q\}$. By Assumption 2.2, this process terminates and $Q_x$ covers $Q^D$. Note that, while constructing a successor function $S_{\Phi}(\cdot)$ as outlined above, if $q'$ and the candidate state $q$ are chosen such that

$$(q, q') = \arg \min_{(q, q') \in Q \times Q} w((q, q')) + V_{\text{aut}, S_{\Phi}}(q'),$$

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

$$V_{\text{aut}, S_{\Phi}}(q) \leq V_{\text{aut}, S}(q), \quad \text{for all } q \in Q^D.$$  

Hence it was demonstrated how $S(\cdot)$ and $\tilde{S}(\cdot)$ 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 $V_{\text{con}, \text{CS}} : \bigcup_{(q, q')} \rightarrow \{ (q, q') | P^0_q \rightarrow \mathbb{Z}_+ :$

$$V_{\text{con}, \text{CS}}((q, q'), x) = \arg \min_{k \in \{1, \ldots, d((q, q')) \}} \{ x \in \mathbb{R}^q \}.$$

where $\mathbb{R}^q$ denotes the set of states in $Q_q$ that can reach $Q_q$ in $k$ steps, i.e.,

$$\beta_{q,k} = \left\{ \{ x \in Q_q \mid \exists u \in U : A x + B u \in P^0_q \} \right\}.$$  

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

$$\beta_{q,k} = \bigcup_{i=1, \ldots, k} \beta_{q,k}.$$  

**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 $P_q \rightarrow \mathbb{Z}_+$ control problems. It is important to note that both $P_q \rightarrow \mathbb{Z}_+$ control methods from Aydin Gol et al. (2014) satisfy these properties. The detailed description of the controllers for solving $P_q \rightarrow \mathbb{Z}_+$ control problems can be found in Section 5 of Aydin Gol et al. (2014).

**Property 6.1.** Let $g : Q_q \rightarrow \{1, \ldots, k^*\}$ be a feedback control law that solves the $P_q \rightarrow \mathbb{Z}_+$ control problem. Let $k^* = w((q, q'))$. Let $x_i \in \mathbb{R}^q$ be the trajectory generated by the feedback control $g(\cdot)$ from the vertex $v_i \in \mathbb{R}^q$, i.e., $x_i := v_i$. The feedback control $g(\cdot)$ satisfies the following properties:

(i) For each vertex $v_i \in \mathbb{R}^q$:

$$x_i \in Q_q, j = 0, \ldots, k^* - 1, \quad \text{and} \quad x_{k^*} \in Q_q.$$

(ii) If $x \in \text{Co}(\{x_j\}_{j=1, \ldots, k^*})$, then $k^*$, it holds that

$$A x + B g(x) \in \text{Co}(\{x_{j+1}\}_{j=1, \ldots, k^*}), \quad \text{for some } 1 \leq j < k^*.$$  

Suppose that Property 6.1 holds, and consider $V_{\text{con}, \text{FC}} : \bigcup_{(q, q')} \rightarrow \{ (q, q') | P^0_q \rightarrow \mathbb{Z}_+ :$

$$V_{\text{con}, \text{FC}}((q, q'), x) = \arg \min_{k \in \{1, \ldots, d((q, q')) \}} \{ x \in \mathbb{R}^q \}.$$  

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

$$\beta_{q,k} = \bigcup_{j=1, \ldots, k} \text{Co}(\{x_{j+1}\}_{j=1, \ldots, k}).$$

Note that both $V_{\text{con}, \text{CS}}(\cdot, \cdot)$ (23) and $V_{\text{con}, \text{FC}}(\cdot, \cdot)$ (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 $V_{\text{con}, \text{CS}}((q, q'), x)$ (23) is the number of steps required to reach $Q_q$ from $x \in Q_q$ by system (1). Therefore, for any other control potential function $V_{\text{con}, \cdot}(\cdot, \cdot)$ it holds that

$$V_{\text{con}, \text{CS}}((q, q'), x) \leq V_{\text{con}}((q, q'), x), \quad \forall (q, q') \rightarrow D, x \in Q_q.$$  

(28)
Moreover from (28), it follows that for all \((q, q') \in \mathcal{D} \) with \( w((q, q')) \neq \infty \),

\[
\mathcal{A}_{\text{Vcon}}^{k, qq} \subseteq \mathcal{A}_{\text{Vcon,CS}}^{k, qq}, \quad k = 1, \ldots, w((q, q')).
\]  

(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 \( V_{\text{SP}}(\cdot, \cdot) \) was defined from \( S_{\text{SP}}(\cdot, \cdot) \) and \( V_{\text{con,EC}}(\cdot, \cdot) \). The function \( V_{\text{con}}(\cdot, \cdot) \) 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., \( v_k = V_{\text{con}}(q_{N,k-1}, x_{N,k-1}) \). Therefore, when \( V_{\text{con,CS}}(\cdot, \cdot), S_{\text{SP}}(\cdot, \cdot) \) and any successor function \( S(\cdot) \) are used for \( V_{\text{con}}(\cdot, \cdot), \overline{S}(\cdot) \), 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 \( q_k \in \mathcal{P}_k^0 \). 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 \( \mathcal{P}_k^0 \) is upper bounded by \( b^k \). The number of quadratic optimization programs solved to find the optimal solution of (13) depends on the representation of the terminal constraint set \( \mathcal{X}^{\infty} \) from (15), and therefore the choice of the control potential function.

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

If \( \mathcal{P}_q, \mathcal{P}_q', \mathcal{P}_k \) and \( \mathcal{U} \) are polytopes, \( \mathcal{A}_{\text{PQ}}(24) \) is also a polytope and can be computed via orthogonal projection. On the other hand, the computation of the sets \( \mathcal{C}(x_{\text{Vcon,CS}}(i, \ldots, x_{\text{PQ}}(\cdot, \cdot))) \) 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 \((q, q') \in \mathcal{D} \) with \( w((q, q')) \neq \infty \), the polyhedral sets \( \mathcal{A}_{\text{PQ}}(i, \ldots, x_{\text{PQ}}(\cdot, \cdot)) \) and \( \mathcal{C}(x_{\text{Vcon,CS}}(i, \ldots, x_{\text{PQ}}(\cdot, \cdot))) \) 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 \( L_k = L = 0.5 I_2 \) and \( R = 0.2 \). The successor function \( S_{\text{SP}}(\cdot, \cdot) \) and the control potential function \( V_{\text{con,CS}}(\cdot, \cdot) \) which are defined in Section 6 are used in Algorithm 1 for \( \overline{S}(\cdot) \) and \( V_{\text{con,EC}}(\cdot, \cdot) \). The successor function \( \overline{S}(\cdot) \) is constructed by traversing the graph of the automaton \( \mathcal{D} \) 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 \( x^i, i = 1, 2, 3 \) are generated by the reference control sequences \( u^i, i = 1, 2, 3 \) from the initial conditions \( x_0^i = \begin{bmatrix} 1 \\ -6 \end{bmatrix}, x_0^2 = \begin{bmatrix} 1.4 \\ -2.8 \end{bmatrix} \), and \( x_0^3 = \begin{bmatrix} 1 \end{bmatrix} \), where

\[
\begin{align*}
u^1 &= 1.8, 1.4, 1.8, 1, 0, -0.7, 0, 0, 0.1, -1.2, 0, \\
u^2 &= 0.6, 0.6, 0.8, 0.8, 0, 0.4, -0.6, -0.6, -0.6, -0.6, -0.6, -0.6, 1, \\
u^3 &= -0.6, -1, -0.3, -0.3, -0.4, -0.5, 0.2, 1, 1, 1, 0.4, -0.8, \\
&-0.5, -0.5, -0.5, 0.5, 0.5, 0.5, 0.5, 0.5, -0.5, 0.2, 0.2, -0.5, -0.5, -1.2, 0, 0.2, 0.2667, 0.1333.
\end{align*}
\]

The reference trajectories \( x^i \) and \( x^3 \) satisfy the specification, while \( x^3 \) violates it. The reference trajectories and the corresponding simulated trajectories generated by Algorithm 1 are shown in Fig. 2. \( x^i \) and \( u^i \) were used to generate Fig. 2(a–b) from Aydin Gol and Lazar (2013). As shown in Fig. 2(a), the controlled trajectory \( x^1 \) follows \( x^1 \) until the specification is satisfied, which was not possible in Aydin Gol and Lazar (2013). As shown in Fig. 2(b), the controlled trajectory \( x^2 \) follows the reference trajectory \( x^2 \) 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 \( x^3 \) satisfies the specification, as shown in Fig. 2(c) the controlled trajectories for optimization.
horizons \( N = 2 \) and \( N = 5 \) cannot exactly follow \( x^3 \) 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 satisfiability were derived based on so-called potential functions. Implementation of the MPC algorithm required solving a finite number of QP problems.

### References


