Safety Control with Preview Automaton

Zexiang Liu and Necmiye Ozay

Abstract—This paper considers the problem of safety controller synthesis for systems equipped with sensor modalities that can provide preview information. We consider switched systems where switching mode is an external signal for which preview information is available. In particular, it is assumed that the sensors can notify the controller about an upcoming mode switch before the switch occurs. We propose preview automation, a mathematical construct that captures both the preview information and the possible constraints on switching signals. Then, we study safety control synthesis problem with preview information. An algorithm that computes the maximal invariant set in a given mode-dependent safe set is developed. These ideas are demonstrated on two case studies from autonomous driving domain.

I. INTRODUCTION

Modern autonomous systems, like self-driving cars, unmanned aerial vehicles, or robots, are equipped with sensors like cameras, radars, or GPS that can provide information about what lies ahead. Incorporating such preview information in control and decision-making is an appealing idea to improve system performance. For instance, a considerable amount of work has been done for designing closed-form optimal control strategies with limited preview on future reference signal [2]–[5] with applications in vehicle control [6], [7]. Preview information or forecasts of external factors can also be easily incorporated in a model predictive control framework [8], [9].

However, similar ideas are less explored in the context of improving system’s safety assurances. For the aforementioned methods on optimal control with preview information, extra constraints need to be introduced into the optimal control problem to have safety guarantees, for which the closed-form optimal solution cannot be derived easily in general. In model predictive control framework, the online optimization is only tractable over a finite receding horizon, and thus it is difficult to have assurance on safety for which the constraints need to be satisfied over an infinite horizon. The goal of this paper is to develop a framework to enable incorporation of preview information in correct-by-construction control.

Specifically, we consider discrete-time switched systems where the mode signal is controlled by external factors. We assume the system is equipped with sensors that provide preview information on the mode signal (i.e., some future values of the mode signal can be sensed/predicted at runtime). To capture how the mode signal evolves and how it is sensed/predicted at run-time, we introduce preview automaton. Then, we focus on safety specifications defined in terms of a safe set within each mode, and develop an algorithm that computes the maximal invariant set inside these safe sets while incorporating the preview information. A simple example where such information can be relevant is depicted in Fig. 1 where an autonomous vehicle can use its forward looking sensors or GPS and map information to predict when the road grade will change. The proposed framework provides a means to leverage such information to compute provably-safe controllers that are less conservative compared to their preview agnostic counterparts.

Our work is related to [10]–[12] where synthesis from linear temporal logic or general omega-regular specifications are considered for discrete-state systems. In [10], it is assumed that a fixed horizon lookahead is available; whereas, in our work, the preview or lookahead time is non-deterministic, and preview automaton can be composed with both discrete-state and continuous-state systems. While we restrict our attention to safety control synthesis, extensions to other logic specifications are also possible. Another main difference with [10] is that we use the preview automaton also to capture constraints on mode switching. The idea of using automata or temporal logics to capture assumptions on mode switching is used in [13] and [14]. In particular, the structure of the controlled invariant sets we compute is similar to the invariant sets (for systems without control) in [13]. However, neither [13] nor [14] takes into account preview information.
The remainder of this paper is organized as follows. After briefly introducing the basic notations next, in Section II we describe the problem setup, define the preview automaton and formally state the safety control problem. An algorithm to solve the safety control problem with preview is proposed and analyzed in Section III. In Section IV we demonstrate the proposed algorithm by two case studies on vehicle cruise control and another on lane keeping before we conclude the paper in Section V.

Notation: We use the convention that the set $\mathbb{N}$ of natural numbers contains 0. Denote the expanded natural numbers as $\mathbb{N} = \mathbb{N} \cup \{\infty\}$. Denote the power set of set $X$ as $2^X$.

II. Problem Setup

In this paper, we consider switched systems $\Sigma$ of the form:

$$x(t + 1) \in f_{\sigma(t)}(x(t), u(t)),$$

where $\sigma(t) \in \{1, \ldots, s\}$ is the mode of the system, $x(t) \in X$ is the state and $u(t) \in U$ is the control input. We assume that the switching is uncontrolled (i.e., the mode $\sigma(t)$ is determined by the external environment) however $\sigma(t)$ is known when choosing $u(t)$ at time $t$. By defining each $f_i : X \times U \to 2^X$ to be set-valued, we capture potential disturbances and uncertainties in the system dynamics that are not directly measured at run-time but that affect the system’s evolution.

We are particularly interested in scenarios where some preview information about the mode signal is available at run-time. That is, the system has the ability to lookahead and get notified of the value of mode signal before the mode signal switches value. More specifically, we assume that for each pair of modes $(i, j)$, if the switching from $i$ to $j$ takes place next, a sensor can detect this switching for $\tau_{ij}$ time steps ahead of the switching time, where $\tau_{ij} \geq 0$ is called the preview time and belongs to a time interval $T_{ij}$. Mathematically, if $\sigma(t + \tau_{ij} - 1) = i$ and $\sigma(t + \tau_{ij}) = j$, then the value $\sigma(t + \tau_{ij})$ is available before choosing $u(t)$ at time $t$, for some $\tau_{ij} \in T_{ij}$.

In many applications, switching is not arbitrarily fast. That is, there is a minimal holding time (or, dwell time) between two consecutive switches. For each mode $i$ of the switched system, we associate a least holding time $H_i \geq 1$ such that if the system switches to mode $i$ at time $t$, the environment cannot switch to another mode at any time between $t$ and $t + H_i - 1$. Note that $H_i = 1$ for all $i$ is the trivial case where the system does not have any constraints on the least holding time. Moreover, there could be constraints on what modes can switch to what other modes.

Following example illustrates some of the concepts above.

Example 1. In Figure 1, a vehicle runs on a highway where the road grade can switch between ranges $r_1 = [-30.5, -29.5]$ and $r_2 = [-0.5, 0.5]$ and between $r_2$ and $r_3 = [29.5, 30.5]$ (no direct switching between $r_1$ and $r_3$). We use a switched system $\Sigma$ of 3 modes to model the three ranges $r_1$, $r_2$ and $r_3$. Thanks to the perception system on the vehicle, the switching from mode $i$ to mode $j$ can be detected $\tau_{ij} \in T_{ij}$ steps ahead, where $T_{ij}$ is a known interval of feasible preview times for $(i, j) \in \{(1, 2), (2, 1), (2, 3), (3, 2)\}$. Also, the least time steps for the vehicle in range $r_i$ is $H_i$ for $i = 1, 2, 3$.

For simplicity, in the rest of the paper, we will assume that the least holding time is greater than or equal to the least feasible preview time among all modes that the system can switch to from mode $i$, i.e., $H_i \geq \min(\cup T_{ij})$ for any mode $i$. This assumption is justified in many applications where switching is “slow” compared to the worst-case sensor range. For instance, the road curvature or road grade does not change too frequently.

The main contribution of this paper is two folds:

- to provide a new modeling mechanism for switched systems that can capture both the constraints on the switching and the preview information,
- to develop algorithms that can compute controllers to guarantee safety with preview information in a way that is less conservative compared to their preview agnostic counterparts.

A. Preview Automaton

Provided the prior knowledge on the preview time interval $T_{ij}$ and the least holding time $H_i$, we model the allowable switching sequences of a switched system with preview with a mathematical construct we call preview automaton.

Definition 1. (Preview Automaton) A preview automaton $G$ corresponding to a switched system $\Sigma$ with $s$ modes is a tuple $G = (Q, E, T, H)$, where

- $Q = \{1, 2, \ldots, s\}$ is a set of nodes (discrete states), where node $q \in Q$ corresponds to the mode $q$ in $\Sigma$;
- $E \subseteq Q \times Q$ is a set of transitions;
- $T : E \to \{(t_1, t_2) : 0 \leq t_1 \leq t_2 : t_1, t_2 \in \mathbb{N}, t_2 \in \mathbb{N}\}$ labels each transition with the time interval of possible preview times corresponding to that transition;
- $H : Q \to (\mathbb{N} \setminus \{0\}) \cup \{\infty\}$ labels each node $q \in Q$ with the least holding time corresponding to that node.

We make a few remarks. First, we do not allow any self-loops, i.e., $(q, q) \notin E$ for all $q \in Q$. Second, the preview times $T(q_1, q_2)$ for any $(q_1, q_2) \in E$ is in one of the three forms: a singleton set $\{t_1\}$ (interval $[t_1, t_1]$), or a finite interval $[t_1, t_2]$ with $t_1 < t_2$, or an infinite interval $[t_1, \infty)$. Finally, if there is a state $q$ with no outgoing edges, that is $\{(q, p) \in E : p \in Q\} = \emptyset$, we set $H(q) = \infty$ to indicate that once the system visits $q$, it remains in $q$ indefinitely, so the deadlock are not allowed. We call such a state a sink state. The set of sink states and the set of non-sink states in $Q$ are denoted by $Q_{\text{sinks}}$ and $Q_{\text{ns}}$.

In Definition 1 the nodes of the preview automaton are chosen to be the modes of the switched system for simplicity. It is easy to extend the definition to allow multiple nodes in the preview automaton to correspond to the same mode. Alternatively, redefining the switched system by replicating certain modes and keeping the current definition can serve the same purpose.
Example 2. The preview automaton for the switched system in Example 1 has nodes $Q = \{1, 2, 3\}$ with transitions shown in Figure 2. The least holding mapping $H(q) = H_q$ for all $q \in Q$ and $T(q_1, q_2) = T_{q_1q_2}$ for $(q_1, q_2) \in \{(1, 2), (2, 1), (2, 3), (3, 2)\}$.

Any transition in the preview automaton is associated with an input in the form of the preview of the switching mode. We assume that there is at most one preview between any two consecutive switches. During the execution of the preview automaton, if a preview takes place at time $t$, there is a corresponding preview input sequence of the preview automaton, including the timestamp $\tau(t)$ of the occurrence of the preview, the destination state $d \in Q$ of the next transition and the remaining time steps (the preview time) $\tau$ from the current time $t$ up to the next transition. If no preview takes place before the next switching time, the preview input corresponding to that switch is trivially $(t, 0, q)$, where $t$ and $q$ are the time instant and destination of the next transition. Note that $t + \tau$ is the time that the system transits from the last mode to the mode $d$.

Definition 2. Given preview automaton $G = \{Q, E, T, H\}$ and initial state $q_0 \in Q$, a sequence of tuples $\{(t_k, \tau_k, d_k)\}_{k=1}^N (N < \infty$ when the system remains in $d_N$ after $t \geq t_N + \tau_N)$ is a valid preview input sequence of $G$ if for all $1 \leq k < N$, the sequence satisfies (with $t_0 = 0$, $\tau_0 = 0$, $d_0 = q_0$) that

1. $\tau_{k-1} \geq 0$ and $t_{k-1} + \tau_{k-1} \leq t_k$ and
2. $(d_{k-1}, d_k) \in E$ and $\tau_k \in T(d_{k-1}, d_k)$ and
3. $(t_k + \tau_k - 1) - (t_{k-1} + \tau_{k-1}) \geq H(d_{k-1})$.

In above definition, conditions (1), (2) and (3) guarantee that only one preview input is received between two consecutive switches, the mode switch constraints and preview time constraints are met, and the holding time constraint is met, respectively. Once a valid input sequence is given, we can uniquely identify the transitions of the preview automaton over time, that is, the execution of the preview automaton with respect to that input sequence. In the rest of the paper, we only consider valid preview input sequences and drop the word valid when it is clear from the context.

Definition 3. Given preview automaton $G = \{Q, E, T, H\}$ and a preview input sequence $\{(t_k, \tau_k, d_k)\}_{k=1}^N$, the execution of $G$ with respect to the preview input sequence is a sequence of tuples $\{(t_k, q_k)\}_{k=0}^N$ where

1. $t_0 = 0, t_1 = t_1 + \tau_1 - 1$ and $t_k = [t_{k-1} + \tau_{k-1} + \tau_k - 1]$ for all $k \geq 0$.
2. $q_k = d_k$ for all $k \geq 1$.

Note that two different valid preview input sequence may have the same execution. According to Definition 2 and 3, the set of possible executions of one preview automaton is determined by the set of valid preview input sequences of the preview automaton.

Once we have the preview automaton $G$ corresponding to a switched system $\Sigma$, we have a model of the allowable switching sequences for $\Sigma$, given by the executions of $G$. Therefore we can define the runs of a switched system with respect to a preview automaton.

Definition 4. A sequence $\{(q(t), x(t))\}_{t=0}^\infty$ is a run of the switched system $\Sigma$ of $s$ modes with preview automaton $G$ under the control inputs $\{u(t)\}_{t=0}^\infty$ if (1) $\{q(t)\}_{t=0}^\infty$ is an execution of $G$ for some preview input sequence and (2) $x(t+1) \in f(q(t), x(t), u(t))$ for $t \geq 0$.

B. Problem Statement

Though the preview automaton can be useful in the existence of more general specification, we focus only on safety specifications in this work. Suppose that each mode $k$ is associated with a safe set $S_k \subseteq X$, that is, the set of states where we require the switched system to stay within when the system’s active mode is $k$. Denote the collection of safe sets $\{S_i\}_{i \in Q}$ for each mode as a safety specification for the switched system $\Sigma$. Then, given a safety specification $\{S_i\}_{i \in Q}$, a run $\{(q(t), x(t))\}_{t=0}^\infty$ is safe if $(q(t), x(t)) \in \bigcup_{i \in Q} (i, S_i)$ for all $t \geq 0$. Otherwise, this run is unsafe.

A controller is usually assumed to know the partial run of the system up to the current time before making a control decision at each time instant. In our case, since the system can look ahead and see the next transition, reflected by the preview input signal, the controller for a switched system equipped with a preview automaton is assumed to have access to the preview inputs of the preview automaton up to the current time.

Definition 5. Denote $\{(p(t), x(t))\}_{t=0}^\infty$ and $\{(t_k, \tau_k, d_k)\}_{k=0}^\infty$ as the partial run and preview inputs of the switched system up to time $t^*$ ($k^*$ refers to the latest preview up to $t^*$, i.e., $k^* = \max k$ s.t. $t_k \leq t^*$). A controller $\mathcal{U}$ of the switched system $\Sigma$ with preview automaton $G$ is a function that maps the partial run $\{(p(t), x(t))\}_{t=0}^\infty$ and the preview inputs $\{(t_k, \tau_k, d_k)\}_{k=0}^\infty$ to a control input $u(t^*)$ of the switched system for any $t^* \geq 0$.

Definition 6. Given a switched system $\Sigma$ and a safety specification $\{S_i\}_{i \in Q}$, a subset $W_i$ of the state space of $\Sigma$ is a single winning set with respect to mode $i$ if there exists a controller $\mathcal{U}$ such that any run of the closed-loop switched system with initial state in $\{i\} \times W_i$ is safe. A winning set $W_i$ is the maximal winning set with respect to the mode $i$ if for any $x \notin W_i$, for any controller $\mathcal{U}$, there exists an unsafe run with initial state $(i, x)$. A winning set with respect to the switched system $\Sigma$ is the collection $\{W_i\}_{i=1}^N$ of single winning sets for all modes, which is called winning set for short.
We note that, by definition, arbitrary unions of winning sets with respect to one mode is still a winning set, and therefore the maximal winning set is unique under mild conditions [15] and contains all the winning sets with respect to that mode. Now, we are ready to state the problem of interest.

**Problem 1.** Given a switched system $\Sigma$ with corresponding preview automaton and safety specification $\{S_i\}_i \subseteq Q$, find the maximal winning set $\{W_i\}_i \subseteq Q$.

Before an algorithm that computes the maximal winning set is introduced, we first study the following toy example to demonstrate the usefulness of preview information.

**Example 3.** A switched transition system with two modes is shown in Figure 3. The state space and input space of the switched system are $\{s_1, s_2, s_3\}$ and $\{u_1, u_2\}$ respectively. The safety specification is $S_1 = S_2 = \{s_1, s_2\}$. To satisfy this safety specification, the system state has to be $s_1$ when $f_1$ is active and $s_2$ when $f_2$ is active. Thus by inspection, when there is no preview, the winning sets are empty and when there is a preview for at least one-step ahead before each transition, there is a non-empty winning set $W_1 = \{s_1\}$ and $W_2 = \{s_2\}$.

Example 3 suggests that a winning set is not the same as a controlled invariant set of the switched system. When the preview information is ignored or unavailable, they are the same and therefore Problem 1 can be solved by computing the controlled invariant sets within the safe sets. However, if the preview is available, the controlled invariant sets can be conservative since their computation does not take advantage of the online preview information. Therefore, in Example 3 the maximal controlled invariant set is empty, but the winning set is non-empty.

### III. MAXIMAL WINNING SET COMPUTATION WITH PREVIEW INFORMATION

In this section, we propose an algorithm to solve Problem 1. Recall that in Definition 1 the feasible preview time interval given by $T$ can be unbounded from the right, which is difficult to deal with in general because it essentially corresponds to a potentially unbounded clock. However, the following theorem reveals an important property of the preview automaton, which allows us to replace the preview time interval with its lower bound in the computation of winning sets.

**Theorem 1.** Let $G = \{Q, E, T, H\}$ and $\hat{G} = \{\hat{Q}, \hat{E}, \hat{T}, \hat{H}\}$ be two preview automata of the switched system $\Sigma$ with $s$ modes, where $\hat{T}(q_1, q_2) = \min(T(q_1, q_2))$ for any $(q_1, q_2) \in E$. Then given a safety specification $\{S_i\}_i \subseteq Q$, $\{W_i\}_i \subseteq Q$ is a winning set with respect to $G$ if and only if $\{W_i\}_i \subseteq \hat{Q}$ is a winning set with respect to $\hat{G}$.

**Proof.** Note that $G$ and $\hat{G}$ are the same except the feasible preview time interval $T$. For each transition $(p_1, p_2) \in E$, $\hat{T}(q_1, q_2)$ is equal to the lower bound of $T(q_1, q_2)$.

To show the “only if” direction, suppose that $W_1$ is a winning set with respect to $G$ for mode $i$. Then by Definition 1 there exists a controller $U$ such that any run of the closed-loop system with initial state in $\{i\} \times W_i$ with respect to any valid preview input sequence of $G$ is safe. By Definition 2 and 3 any preview input sequence and the corresponding execution of $\hat{G}$ are also valid preview inputs and execution of $G$. Therefore, using the same controller $U$, any run of the closed-loop system with initial state in $\{i\} \times W_i$ with respect to any valid preview input sequence of $G$ is safe. Hence we conclude that each $W_i$, for $i \in Q$, is a winning set with respect to $\hat{G}$ for mode $i$.

To show the “if” direction, suppose that $W_1$ is a winning set with respect to $G$ for mode $i$, and $U$ is a controller such that any run of the closed-loop system with initial state in $\{i\} \times W_i$ with respect to any valid preview input sequence $\hat{G}$ is safe. Note that any execution of $G$ is also an execution of $\hat{G}$, but the corresponding preview input sequences of $G$ and $\hat{G}$ can be different. Suppose that $\{(i_k, q_k, \tau_k)\}_{k=1}^N$ and $\{(i'_k, q'_k, \tau'_k)\}_{k=1}^N$ are two preview input sequences of $G$ and $\hat{G}$ corresponding to the same execution $\pi = \{(i_k, q_k)\}_{k=0}^N$. Then by Definition 2 for all $k \geq 1$, $\tau'_k' = \hat{T}(q'_k, q'_k+1) = \min(T(q'_k, q'_k+1))$ and $\tau_k = T(q_k, q_k+1)$ and $\tau'_k' = \tau_k = \min(T(q_k, q_k+1))$. Hence $\tau'_k' \leq \tau_k$ and $\tau'_k' \geq \tau_k$ for all $k \geq 1$, which implies that for any transition in $\pi$, a controller of $G$ always knows the next mode from the preview input of $G$ earlier than a controller of $\hat{G}$.

Since the preview input of the next transition is earlier in $G$ than in $\hat{G}$, given an execution $\pi = \{(i_k, q_k)\}_{k=0}^N$ for any $k \geq 1$, $\forall k$ can always infer the $k^{th}$ preview input of $G$ from the $k^{th}$ input of $G$ before time $t'_k'$. We force controller $U$ to generate control inputs for the switched system $\Sigma$ with preview automaton $G$ based on the inferred inputs of $\hat{G}$. Then any run of $\Sigma$ when closing the loop with the customized $\Sigma$ and $\hat{G}$ is a run of the closed-loop system with respect to $\Sigma$, $\forall k$ and $\hat{G}$, which is safe if the initial state is in $\{i\} \times W_i$. Hence $W_i$ is a winning set of $G$ for all $i \in Q$.

Thanks to Theorem 1 in terms of maximal winning set computation, it is enough to consider the preview automaton whose preview time interval is a singleton set for all transitions without introducing any conservatism. This

---

2Given the $k^{th}$ preview input $(i_k, q_k, \tau_k, q_{k+1})$ of $G$, the $k^{th}$ preview input of $\hat{G}$ is $(i_k + \tau_k - \tau_k', q'_k, q_{k+1})$ with $\tau_k' = \hat{T}(q_k, q_{k+1})$. 
property stated in Theorem 1 can reduce computation cost and simplify the algorithms. Therefore, whenever there is a preview automaton $G$, we first convert $G$ into the form of $\hat{G}$ in Theorem 1. Algorithm 1 is designed to compute the maximal winning set for the preview automaton in the form of $\hat{G}$, whose result is equal to the maximal winning set of $G$. We note that $\hat{G}$ can be expanded to a non-deterministic finite transition system with $\sum_i (H_i - (\min_j T_{ij}) + \sum_j T_{ij})$ states. Taking a product of this finite transition system with the switched system, the problem can be reduced to an invariance computation (with measurable and unmeasurable non-determinism) on the product system. However, the algorithms we propose avoid product construction and directly define fixed-point operations on the switched system’s state space.

In Algorithm 1, lines 4-6 compute the maximal winning set for each sink state in $G$. Lines 7-12 compute the winning sets of the non-sink states iteratively, with updates given by Algorithm 2. The main operators used in these algorithms are as follows. First, given a mode $i$ of the switched system with state space $X$ and action space $U$, and a subset $V$ of $X$, the one-step controlled predecessor of $V$ with respect to the dynamics $f_i$ is defined as

$$Pre^i(V) = \{x \in X : \exists u \in U, f_i(x, u) \subseteq V\},$$

that is the set of states that can be guaranteed to reach the set $V$ in one time step by some control inputs in $U$.

Second, given a safe set $S_i \subseteq X$, the one-step constrained controlled predecessors $PreInt(\cdot)$ of an arbitrary set $V$ with respect to the dynamics $f_i$ as

$$PreInt^i(V, S_i) = Pre^i(V) \cap S_i,$$

Now define $V_0 = S_i$ and update $V_k$ recursively for $k \geq 0$ by

$$V_{k+1} = PreInt^i(V_k, S_i).$$

Note that $\{V_k\}_{k=0}^\infty$ is monotonically non-increasing sequence of sets and the fixed point (reached when $V_{k+1} = V_k$) is the maximal controlled invariant set within the safe set $S_i$ with respect to the dynamics $f_i$, denoted as $Inv^i(S_i)$. Finally, given the preview automaton $G = \{Q, E, T, H\}$, the successors of some node $i \in Q$ is defined as $Post^G(i) = \{j : (i, j) \in E\}$.

Some properties of these operators and the $InvPre$ operator defined by Algorithm 2 are analyzed next. These properties are used later to prove the correctness of the main algorithm. In what follows we use $\{\tilde{W}_i\}_{i \in Q} \subseteq \{W_i\}_{i \in Q}$ to denote the element-wise set inclusion $\tilde{W}_i \subseteq W_i$ for all $i \in Q$. When we talk about maximality, maximality is in (element-wise) set inclusion sense.

**Lemma 1.** Consider two collections of subsets $\tilde{W} = \{\tilde{W}_i\}_{i \in Q}$ and $W = \{W_i\}_{i \in Q}$ of $X$. If $\tilde{W} \subseteq W \subseteq S$, then $\tilde{W}$ and $W$ satisfy

$$InvPre^h(G, \tilde{W}, S) \subseteq InvPre^h(G, W, S) \subseteq S_i$$

for any non-sink state $i \in Q_{ns}$.

**Algorithm 1 Winning Set for Problem 1**

```plaintext
1: function $ConInv(S, S = \{S_i\}_{i \in Q}, G)$
2:   initialize $\{W_i\}_{i \in Q}$ with $W_i = S_i, \forall i \in Q$.
3:   initialize $\{\tilde{W}_i\}_{i \in Q}$ with $\tilde{W}_i = \emptyset$.
4:   for $i \in Q$ such that $H(i) = \infty$ (sink states)
5:     $W_i \leftarrow Inv^i(S_i)$
6:   end for
7: while $\exists i \in Q$ such that $W_i \neq \tilde{W}_i$
8:   $\tilde{W}_i \leftarrow W_i, \forall i \in Q$
9:   for $i \in Q$ such that $H(i) < \infty$
10:     $W_i \leftarrow InvPre^i(G, \{W_i\}_{j \in Post(i)}, S)$
11:   end for
12: end while
13: return $\{W_i\}_{i \in Q}$
14: end function
```

**Algorithm 2 InvPre operator for Algorithm 1**

```plaintext
1: function $InvPre^i(G, W, S)$
2:   for $j$ in $Post^G(i)$
3:     $C_{0j} = W_j$ and $T_{ij} = T(i, j)$
4:   for $l = 1, 2, 3, ..., T_{ij}$
5:     $C_{l,j} = PreInt^i(C_{l-1,j}, S_i)$
6:   end for
7: end for
8: $T_{\text{min}} = \min_{j \in Post^G(i)} T(i, j)$
9: $C_{\text{max}} = Inv^i(\bigcap_{j \in Post^G(i)} CT_{ij,j})$
10: $H_i = H(i)$
11: for $k = T_{\text{min}} + 1, \cdots, H_i$
12:    $C_k = PreInt^i(C_{k-1,j}, S_i)$
13:    if $J_k = \{j \in Post^G(i) : T_{ij} \geq k\} \neq \emptyset$ then
14:      $C_k = C_k \cap (\bigcap_{j \in J_k} CT_{ij,j})$
15:    end if
16: end for
17: return $C_{H_i}$
18: end function
```

**Lemma 2.** $W = \{W_i\}_{i \in Q}$ is the maximal winning set with respect to the safe set $S = \{S_i\}_{i \in Q}$ if and only if $\{W_i\}_{i \in Q_{ns}}$ is the maximal solutions of the following equations:

$$W_i = InvPre^i(G, W, S), \forall i \in Q_{ns}.$$  \hspace{1cm} (6)

where the components of the winning set $W$ for sink states are chosen according to $W_j = Inv^j(S_j)$ for all $j \in Q$.

The proofs of Lemma 1 and 2 are given in the appendix. Let us illustrate how Algorithm 1 works, using the switched system shown in Fig. 4 with preview automaton in Fig. 4 before proving that the proposed algorithm indeed computes the maximal winning set.

**Example 4.** Since there are no sink nodes in the preview automaton in Fig. 4, lines 4-6 in Algorithm 1 are skipped. We use pair $(k,l)$ to indicate the $k$th iteration of the whole loop and $l$th iteration of the for loop in line 7 and 9 in Algorithm 1 and use $W^{k,l}$ to refer to the value of $W_i$ after the $(k,l)$
iteration. Note that at iteration \((k,l)\), only \(W_l\) is being updated and the other \(W_i\) remains unchanged for \(i \neq l\).

Initially \(W_{1,0} = W_{2,0} = \{s_1, s_2\}\). In the iteration \((0,1)\), \(W_{1,1} = InvPre^f_i(G, \{W_{1,0}, W_{2,0}\}, S) = \{s_1\}\) and \(W_{2,1} = W_{2,0}\).

In the iteration \((0,2)\), \(W_{0,2} = InvPre^f_i(G, \{W_{1,1}, W_{2,1}\}, S) = \{s_2\}\) and \(W_{1,2} = W_{1,1}\). In the following iterations \((1,1), (1,2), W_{1,1}\) and \(W_{2,2}\) are unchanged. Therefore, the termination condition in line 7 is satisfied and the output of Algorithm 1 of this example is \(W_1 = \{s_1\}\) and \(W_2 = \{s_2\}\). It is easy to verify that \(W_i = \{s_i\}\) and \(W_2 = \{s_2\}\) form the maximal winning set for this problem.

The main completeness result is provided next.

**Theorem 2.** If Algorithm 1 terminates, the tuple of sets \(\{W_i\}_{i=1}^n\) it returns is the maximal winning set within the safe set \(S = \{S_i\} \in Q\) of the switched system \(\Sigma\) with the preview automaton \(G\).

**Proof.** Suppose that \(\{W_i\}_{i=Q}\) is the maximal winning set we are looking for. Let us partition the discrete state space \(Q\) into the set of sink states \(Q_s = \{q \in Q : H(q) = \infty\}\) and the set of non-sink states \(Q_{\text{ns}} = Q \setminus Q_s = \{q \in Q : H(q) < \infty\}\).

If \(q\) is a sink state, once the system enters the mode \(q\), the system remains in mode \(q\) without any future switching. Therefore, the maximal winning set \(W_q\) is trivially the maximal controlled invariant set within the safe set \(S_q\) with respect to the dynamics of mode \(q\), that is \(W_q = Inv^f_i(S_q)\). In line 4-6 of Algorithm 1 we compute the maximal winning sets for all the sink states.

We have solved \(W_q\) for sink state \(i \in Q_s\). Let us consider the maximal winning sets for non-sink states. We want to show that \(W\) being updated based on lines 7-12 of Algorithm 1 converges to \(W^*\). Without loss of generality, assume that \(Q_{\text{ns}} = \{1, 2, ..., n_{\text{ns}}\}\) and let the “for” loop in line 9 iterate over the indices \(1, 2, ..., n_{\text{ns}}\) in the natural order.

We use \(W_{k,l}\) to indicate the updated value of \(W\) after the \(k\)th iteration of the “while” loop (line 7) and the \(p\)th iteration of the “for” loop (line 9). Then the initial value of \(W\) is \(W_{0,0} = \{W_{i,0}\}_{i=Q}\) where \(W_{i,0} = S\) for all \(i \in Q_{\text{ns}}\) and \(W_{i,0} = \emptyset\) for all \(i \in Q_s\). According to line 9-10, for all \(k \geq 0\) and \(0 \leq l \leq n_{\text{ns}} - 1\), \(W_{k,l+1} = InvPre^f_i(G, W_{k,l}, S)\) for \(i = l + 1\) and \(W_{j,l+1} = W_{j,l}\) for all \(j \neq l + 1\) and \(W_{k,\infty} = W_{k,n_{\text{ns}}}\). Now we want to prove that if \(W^* \subseteq W_{k,0}\) for some \(k \geq 0\), then \(W^* \subseteq W_{k,l} \subseteq S\) for any \(l \in \{1, 2, ..., n_{\text{ns}}\}\) by induction. (Base case 1) Since we have \(W^* \subseteq W_{k,l} \subseteq S\), by Lemma 1 and 2, we have

\[W_{i} = InvPre^f_i(G, W^*, S) \subseteq InvPre^f_i(G, W_{k,0}, S) = W_{k,0} \subseteq S_{i}\]

for \(i = 1\). Since \(W^* \subseteq W_{k,0}\) and \(W_{j,1} = W_{j,0}\) for all \(j \neq 1\), we have \(W^* \subseteq W_{k,1} \subseteq S\). (Induction hypothesis 1) Suppose that \(W^* \subseteq W_{k,l} \subseteq S\) for some \(0 \leq l \leq n_{\text{ns}} - 1\). Again, by Lemma 1 and 2, \(W^* \subseteq W_{k,l+1} \subseteq S\). Finally by induction, if \(W^* \subseteq W_{k,l} \subseteq S\), \(W^* \subseteq W_{k,l} \subseteq S\) for all \(l \in \{1, 2, ..., n_{\text{ns}}\}\).

Then next we want to prove by induction that \(W^* \subseteq W_{k,l} \subseteq S\) for all \(k \geq 0\). (Base case 2) \(W^* \subseteq W_{0,0} \subseteq S\) by construction. (Induction hypothesis 2) Suppose \(W^* \subseteq W_{k,0} \subseteq S\) for some \(k \geq 0\). Then, we have proven that \(W^* \subseteq W_{k,n_{\text{ns}}} = W_{k,1,0} \subseteq S\). Therefore by induction, \(W^* \subseteq W_{k,0} \subseteq S\) for any \(k \geq 0\).

The two induction arguments above prove that \(W^* \subseteq W_{k,l} \subseteq S\) for any \(k \geq 0\) and \(0 \leq l \leq n_{\text{ns}}\).

Now let us show that \(W_{0,0}, W_{1,0}, W_{2,0}, ..., W_{k,0}, W_{k,1}, ..., W_{k,\infty}\) is a non-expanding sequence. Since \(W_{k,\infty} = W_{k,1,0}\) for all \(k \geq 0\), it suffices to show \(W_{k,l+1} \subseteq W_{k,l}\) for any \(k \geq 0\) and \(0 \leq l \leq n_{\text{ns}} - 1\).

(Base case 3) Note that \(InvPre^f_i(G, V, S) \subseteq S_i\) for arbitrary \(V \subseteq X\) and \(i \in Q\). Thus by definition \(W_{1,0} = InvPre^f_i(G, W_{0,0}, S) \subseteq S_i = W_{0,0}\) for \(i = 1\). Note that \(W_{1,0} = W_{j,0}\) for all \(j \neq 1\). Thus \(W_{0,1} \subseteq W_{0,0}\). Now consider \(W_{2,0}\). Note that \(W_{2,0} = InvPre^f_i(G, W_{0,0}, S) \subseteq S_i = W_{0,1}\). Thus \(W_{2,0} \subseteq W_{0,1}\). Similarly, we have \(W_{0,\infty} \subseteq \cdot \cdot \cdot \subseteq W_{0,1} \subseteq W_{0,0}\).

(Induction hypothesis 3) Suppose \(W_{k,\infty} \subseteq \cdot \cdot \cdot \subseteq W_{k,1} \subseteq W_{k,0}\) for some \(k > 0\). To show that \(W_{k+1,l+1} \subseteq W_{k+1,l}\) for all \(l\), we need another induction argument. (Base case 4) We know \(W_{k+1,0} = W_{k+1,1,0}\), and \(W_{k+1,1,0} = W_{j,0}\) for \(j \neq 1\).

For \(i = 1\), \(W_{k+1,1} = W_{k+1,1,0} = W_{k+1,0} = InvPre^f_i(G, W_{k,0}, S)\). By induction hypothesis 3, \(W_{k+1,1} \subseteq W_{k,0}\) and thus by Lemma 1 and 2

\[W_{k+1,1} = InvPre^f_i(G, W_{k+1,0}, S) \subseteq InvPre^f_i(G, W_{k,0}, S) = W_{k+1,0} \]

for \(i = 1\), and therefore \(W_{k+1,1} \subseteq W_{k+1,0}\).

(Induction hypothesis 4) Suppose that \(W_{k+1,l+1} \subseteq \cdot \cdot \cdot \subseteq W_{k+1,0}\) for all \(j \neq l + 1\). Also, for \(i = l + 1\), \(W_{l+1,l+1} = W_{l+1,1} = InvPre^f_i(G, W_{k,l}, S)\). By the induction hypothesis 3 and 4, \(W_{l+1,1} \subseteq W_{k,l}\) and thus by Lemma 1 and 2, again, for \(i = l + 1\),

\[W_{l+1,l+1} = InvPre^f_i(G, W_{l+1,1}, S) \subseteq InvPre^f_i(G, W_{l+1,0}, S) = W_{l+1,1} \]

and therefore \(W_{l+1,l+1} \subseteq W_{l+1,l}\). Then by induction 4, we have \(W_{k+1,\infty} \subseteq \cdot \cdot \cdot \subseteq W_{k+1,1} \subseteq W_{k+1,0}\).

Therefore by the induction 3, we show that \(W_{0,0}, W_{0,1}, W_{0,2}, ..., W_{0,\infty}\) is a monotonic non-expanding sequence within \(S\), which implies that the limit of this sequence \(W^*\) (the output of Algorithm 1) exists and is contained by \(S\) thus safe. By line 7-12 of Algorithm 1 \(W^* = 0\) is a solution of equations in (6). Also, since for any \(k\) and \(l\), \(W^* \subseteq W_{k,l}\) and \(W^*\) is the maximal solution of equations in (6), we have \(W^* = 0\).
Note that the above proof also guarantees termination if the switched system under consideration has finitely many states. For switched systems with continuous state spaces, the non-expanding property of the computed sets guarantees convergence but termination in finite number of steps is not guaranteed, in general. For linear switched systems, termination can still be guaranteed using algorithms from [16], [17] by slightly sacrificing maximality (see also [18]).

Once the maximal winning set (or a winning set) \( W^* = \{ W^*_i \}_{i \in \mathcal{Q}} \) is obtained, a controller can be extracted roughly as follows: for a sink node \( i \in \mathcal{Q}_s \), the allowable control inputs for each state in the controlled invariant set \( W^*_i \) can be obtained by applying the \( Pre \) operator to \( W^*_i \). For a non-sink node \( j \in \mathcal{Q}_n \), we need a “invariance” controller to make sure the system state remain in \( W^*_j \) before a preview happens, and a “reachability” controller for each transition \( (j, k) \in E \) and each possible preview time \( \tau_{jk} \in T(j, k) \) such that from the time point a preview is received by the controller, system state can guarantee to reach \( W^*_k \) in \( \tau_{jk} \) steps, where the allowable control input for each step can be obtained by applying the \( PreInv \) recursively for \( \tau_{jk} \) times. For the “invariance” controller, we also need to make sure that the system state reaches certain parts of the maximal winning set based on the holding time (time steps elapsed since last transition) such that once a preview occurs, the system state is within the domain of the corresponding “reachability” controller. The process of computing the “reachability” controllers actually corresponds to line 2-7 in Algorithm 2 and the process of computing the “invariance” controller corresponds to line 9 and 12-14, if the preview automaton has a singleton preview time interval. The process can be generalized to general preview time intervals from the Algorithm 2 based on the description above.

IV. CASE STUDIES

In the following case studies, we apply the proposed algorithms to switched affine systems, where the state space and safe set are polytopes. In this case \( Pre \) and \( PreInv \) operators reduces to polytopic operations, which we implement using the MPT3 toolbox [19].

A. Vehicle Cruise Control

Our first example is a cruise control problem for the scenario shown in Example 1. The longitudinal dynamics of a vehicle with road grade is given by

\[
\dot{v} = -\frac{f_0}{m} - \frac{f_1}{m} v + \frac{F_w}{m} - g \sin \theta
\]  

(7)

where \( v \) is the longitudinal speed, \( m \) is the vehicle mass, \( f_0 \) and \( f_1 \) are the coefficients related to frictions, \( F_w \) is the wheel force, \( g \) is the gravitational acceleration and \( \theta \) is the road grade. We choose \( F_w \) as the control input and \( \Theta \) as a disturbance. We discretize \( 7 \) with time step \( \Delta t = 0.1s \). The discrete-time dynamics with disturbance ranges \( r_1, r_2 \) and \( r_3 \) consist of the modes 1, 2, 3 in the switched system defined in Example 1.

The safety specification is to keep the longitudinal speed within \( X = [31.95, 32] m/s \). The speed range is intentionally picked small enough so that the change the road grade induces on the dynamics makes the specification hard to be satisfied. The parameters are chosen as \( m = 1650kg, f_0 = 0.1N, f_1 = 5N/s/m, g = 10m/s^2 \). The control input range is \( F_w = [-0.65mg, 0.66mg] \) for the preview automaton shown in Fig. 2 the holding time for each mode is 2 and the preview time for each transition is 1.

To make a comparison, we compute the maximal controlled invariant set for the dynamics discretized from 7 with disturbance in \([-30.5^\circ, 30.5^\circ]\) (convex hull of \( r_1, r_2, r_3 \)). If such an invariant set exists, it is a feasible winning set for our problem. However, the resulting controlled invariant set is empty, which suggests that the problem is infeasible if disturbance can vary arbitrarily in \([-30.5^\circ, 30.5^\circ]\). In contrast, the winning set obtained from Algorithm 1 is \( \{ W_i \}_{i=1} \) with \( W_1 = W_2 = W_3 = X \). Therefore the preview automaton is crucial in this case study for the existence of a safety controller.

B. Vehicle Lane Keeping Control

In the second example, we apply the proposed method to synthesize a lane-keeping controller, which controls the steering to limit the lateral displacement of vehicle within the lane boundaries.

The lateral dynamics we use are from a linearized bicycle model [18]. The four states of the model consist of the lateral displacement \( y \), lateral velocity \( v \), yaw angle \( \Delta \Psi \) and yaw rate \( r \). The vehicle is controlled by the steering input \( \delta_t \) in range \([-\pi/2, \pi/2]\). We assume that the longitudinal velocity \( u \) of the vehicle is constant and equal to 30m/s. The disturbance \( r_d \) is a function of the road curvature, which is what we assume to have preview information on at run-time.

The maximal recommended range of \( r_d \) on Michigan highways [20] with respect to \( u = 30m/s \) is about \([-0.06, 0.06]\). We divide \([-0.06, 0.06]\) evenly into 5 intervals \( d_1 = [-0.06, -0.036], d_2 = [-0.036, -0.012], \ldots, d_5 = [0.036, 0.06] \) and construct a switched system with 5 modes, where each mode \( i \in Q = \{1, 2, 3, 4, 5\} \) corresponds to a lateral dynamics with \( r_d \) bounded in \( d_i \), denoted by \( f_i \). The corresponding preview automaton is shown in Fig. 5 where transitions are only between any two modes with adjacent \( r_d \) intervals. For simplicity, the preview time interval \( T(i, j) = \tau_c \) for all \( (i, j) \in E \), and the least holding time \( H(i) = \tau_d \) for all \( i \in Q \) for some constants \( \tau_c \) and \( \tau_d \).

The safe set is given by the constraints \( |y| \leq 0.9, |v| \leq 1.2, |\Delta \Psi| \leq 0.05, |r| \leq 0.3 \) for all modes.

TABLE I: Computation costs for different \((\tau_c, \tau_d)\)

| \((\tau_c, \tau_d)\) | #iterations | time (min) |
|---------------------|-------------|------------|
| (1.2)               | 4           | 18.9       |
| (2.2)               | 4           | 18.0       |
| (1.1)               | 5           | 20.3       |
| (5.5)               | 3           | 16.8       |

We apply Algorithm 1 to compute the maximal winning sets for various \( \tau_c \) and \( \tau_d \). The values of \((\tau_c, \tau_d)\) with corresponding numbers of iterations at termination and running time are listed in Table 1. Denote the maximal winning set
Fig. 5: Preview automaton for the lane-keeping case study

As a comparison, we compute the maximal controlled invariant set based approaches. In this paper, we introduced preview automaton and provided an algorithm for safety control synthesis in the existence of preview information. The proposed algorithm is shown to compute the maximal winning set upon termination. These ideas are demonstrated with two examples from the autonomous driving domain. As shown in these examples, incorporation of preview information in control synthesis leads to less conservative safety guarantees compared to standard controlled invariant set based approaches. In this paper, we will investigate the use of preview automaton for synthesizing controllers from more general specifications. We also have some ongoing work investigating the connections of preview automaton with discrete-time I/O hybrid automaton with clock variables representing preview and holding times.

Fig. 6: Projections of \( W_{inv}, W_2(1,2), W_2(2,2) \) onto two subspaces. The red, blue and green regions are the projection of \( W_{inv} \), the difference of projections of \( W_2(1,2) \) and \( W_{inv} \) and the difference of projections of \( W_2(2,2) \) and \( W_2(1,2) \).

With respect to mode 2 for each pair \((\tau_c, \tau_d)\) in Table I as \( W_2(\tau_c, \tau_d) \).

As a comparison, we compute the maximal controlled invariant set for the lateral dynamics with \( \tau_d \) to 2 while the least holding time \( \tau_d \) is subject of our future work.

Fig. 7: Projections of \( W_{inv}, W_2(1,1), W_2(1,5) \) onto two subspaces. The red, blue and green regions are the projection of \( W_{inv} \), the difference of projections of \( W_2(1,1) \) and \( W_{inv} \) and the difference of projections of \( W_2(1,5) \) and \( W_2(1,1) \).

The change of the least holding time \( \tau_d \) than the change of the preview time \( \tau_c \).

Finally, \( W_{inv}, W_2(1,1), W_2(5,5) \) are compared in Fig. 7 where we increase \( \tau_c \) and \( \tau_d \) simultaneously. \( W_2(5,5) \) is numerically equal to \( W_2(1,5) \), and thus its projections are the same as the projections of \( W_2(1,5) \) shown in Fig. 7. In fact, the winning sets with respect to modes 2, 3, 4 for \( \tau_c = 1, \tau_d = 5 \) and \( \tau_c = 5, \tau_d = 5 \) are numerically equal; the winning set with respect to mode 1 and 5 slightly grows when \((\tau_c, \tau_d)\) changes from \((1,5)\) to \((5,5)\), but the growth is too small to be visualized. The observation in Fig. 6 and 7 reveals one theoretical conjecture: If the preview time and the least holding time are large enough, a longer preview time and/or a longer holding time will not increase the size of the maximal winning set. That is, the size of the maximal winning set converges as the preview time and the least holding time increase. To verify this conjecture is part of our future work.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced preview automaton and provided an algorithm for safety control synthesis in the existence of preview information. The proposed algorithm is shown to compute the maximal winning set upon termination. These ideas are demonstrated with two examples from the autonomous driving domain. As shown in these examples, incorporation of preview information in control synthesis leads to less conservative safety guarantees compared to standard controlled invariant set based approaches. In the future, we will investigate the use of preview automaton for synthesizing controllers from more general specifications. We also have some ongoing work investigating the connections of preview automaton with discrete-time I/O hybrid automaton with clock variables representing preview and holding times.

REFERENCES

[1] Z. Liu and N. Ozay, "Safety control with preview automaton," in Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control. ACM, 2019, pp. 280–281.

[2] T. B. Sheridan, “Three models of preview control,” IEEE Transactions on Human Factors in Electronics, no. 2, pp. 91–102, 1966.

[3] M. Tomizuka and D. Whitney, “Optimal discrete finite preview problems (why and how is future information important?),” Journal of Dynamic Systems, Measurement, and Control, vol. 97, no. 4, pp. 319–325, 1975.
APPENDIX

Proof of Lemma 7 By definition of Pre, for any sets $W_1 \subseteq W_2 \subseteq S$, $Pre_{i}^{f}(W_1) \subseteq Pre_{i}^{f}(W_2)$ for any $i \in Q$. Therefore $Pre_{i}^{f}(W_1, S_i) \subseteq Pre_{i}^{f}(W_2, S_i) \subseteq S_i$ and furthermore $Inv_{i}^{f}(W_1) \subseteq Inv_{i}^{f}(W_2) \subseteq S_i$ for any $i \in Q$. According to line 2 of Algorithm 1 the input $W$ of InvPre is always a subset of $S$. Note that all the intermediate variables $C_{i,j}$, $C_{m,n}$, $C_{k}$ are recursively computed by PreInt and Inv. Based on the monotonicity of PreInt and Inv we can check step by step that the values of the intermediate variables $C_{i,j}$, $C_{m,n}$, $C_{k}$ of InvPre with respect to input $W_1$ are contained by the values of those variables with respect to input $W_2$. Therefore $InvPre_{i}^{f}(G, W_1, S) \subseteq InvPre_{i}^{f}(G, W_2, S) \subseteq S_i$.

Proof of Lemma 2 First note that $InvPre_{i}^{f}(G, W, S)$ only depends on $G$, $\{W_j\}_{j \in PostG(i)}$ and $S$, though we use the whole $W$ instead of $\{W_j\}_{j \in PostG(i)}$ as input of Algorithm 2 for short. In this proof, we change the notation to $InvPre_{i}^{f}(G, \{W_j\}_{j \in PostG(i)}$, $S)$ to make this point clear.

The key observation in this proof is: if the system switches from node $i$ to node $j$ at some time $t$ with state $x(t)$, for the purpose of synthesizing future control strategies, it is equivalent to the case that the system initially starts from the state $(j, x(t))$, and therefore there exists a controller to guarantee safety for the rest of the run if and only if $x(t) \in W^*_j$.

Denote $\{W^*_j\}_{i \in Q}$ as the maximal winning set. By Proposition 2 $W^*_j = Inv_{i}^{f}(S_i)$ for all sink states $j \in Q_s$. Let $k \in Q_{ns}$. Now suppose that we know the maximal winning set $W^*_j$ with respect to all $k \in Q_{ns}$ except $i$. We want to show that $W^*_j$ can actually be computed by $InvPre_{i}^{f}(G, \{W^*_j\}_{j \in PostG(i)}$, $S)$. Note that $i \notin Post(i)$ since we do not allow self-loops in the preview automaton.

Denote the minimum preview time among all feasible transitions as $T_{min} = \min_{i \in PostG(i)} T(i, j)$. Let us first consider the case where no preview happens during the time interval $[0, t - 1]$ with $t \geq H(i) - T_{min}$. In this case, the least holding time constraint will be satisfied whatever the next transition is. Let us consider the maximal set of states $C_{i, j}$ such that if $x(t)$ is within $C_{i, j}$, there exists a controller that makes the closed-loop system satisfy the safety spec.

Suppose that one preview happens at $t$ with destination state $j \in PostG(i)$ and remaining time $\tau_j = T(i, j)$. To guarantee the safety spec being satisfied in the future, for any $t \leq t' < t + \tau_j$, $x(t')$ has to be within $S_i$, and $x(t + \tau_j)$ has to be within $W^*_j$ at time $t + \tau_j$. The maximal subset of $S_i$ that is able to reach $W^*_j$ in one step is given by $PreInv_{i}^{f}(W^*_j, S)$. By applying $PreInv_{i}^{f}$ many times, we obtain the maximal subset of states in $S_i$ that can stay within $S_i$ until reaching $W^*_j$ at $t + \tau_j$, which is $C_{i,j}$ in line 2 of Algorithm 2.

The intersection $\bigcap_{i \in PostG(i)} C_{i,j}$ is the maximal set of values of $x(t)$ that guarantees safety spec under the condition that a preview takes place at $t$. If there is no preview at $t, t + 1, \ldots$, the system needs to stay within $\bigcap_{i \in PostG(i)} C_{i,j}$ for safety at time $t + 1, t + 2, \ldots$. Therefore $C_{i,j} = Inv_{i}^{f}(\bigcap_{i \in PostG(i)} C_{i,j})$ (line 9).

Now consider the case where no preview takes place between $[0, t - 1]$ where $t = H(i) - T_{min} - 1$. Denote $C_{i,j}^{max+1}$ as the maximal subset of $S_i$ such that if $x(t) \in C_{i,j}^{max+1}$ there exists a controller to guarantee safety spec; otherwise there is no such a controller. If there is no preview at time $t$, all we want for safety is $x(t + 1) \in C_{i,j}$. Then $PreInv_{i}^{f}(C_{i,j})$ is the maximal set of values of $x(t)$ such that there exists a controller that makes sure $x(t + 1) \in C_{i,j}$. Otherwise, if there is a preview at $t$ with destination state $j$, all we want is $x(t) \in C_{i,j}$ by the previous discussion. The set of all feasible destination states that can have preview at $t$ is $J_{i,j}^{max+1} = \left\{ j \in PostG(i) : \tau_j \geq T_{min} + 1 \right\}$. Therefore, depending on whether $J_{i,j}^{max+1}$ is empty or not, $C_{i,j}^{max+1}$ is equal to $PreInv_{i}^{f}(C_{i,j})$ or $PreInv_{i}^{f}(C_{i,j}^{max+1} \cap \bigcap_{i \in PostG(i)} C_{i,j})$ (line 12 and 14), which is guaranteed to have a safety controller for all the situations. The same discussion can be repeated for $t = H(i) - T_{min} - k$ up to $k = H(i) - T_{min}$.
resulting in $C_{T_{\min}+k} = \text{PreInv}^{J(i)}(C_{T_{\min}+k-1}) \cap \bigcap_{j \in J_{T_{\min}+k}} C_{T_{ij}}$ for $J_{T_{\min}+k} = \{ j \in \text{Post}^G(i) : T_{ij} \geq T_{\min} + k \}$. For the case $k = H(i) - T_{\min}$, $t = 0$ and $C_{H(i)}$ is the maximal set of values of $\lambda(0)$ such that if $\lambda(0) \in C_{H(i)}$, there exists a controller that makes sure the closed-loop system satisfy the safety spec, which is the maximal winning set with respect to $i$. Therefore $W^*_i = C_{H(i)} = \text{InvPre}^{J(i)}(G, \{ W^*_j \}_{j \in \text{Post}^G(i)}, S)$ (line 14), and $\{ W^*_j \}_{j \in \text{Qns}}$ is a solution of equations in (6).

We have proved that the maximal winning set must satisfy the equations in (6). On the other hand, by construction of InvPre, it can be verified that any solution $\{ W_i \}_{i \in \text{Qns}}$ with $W_j \subseteq S_j$ for $j \in \text{Qns}$ of the equations in (6) with respect to $\{ W^*_k \}_{k \in \text{Qns}}$ is a winning set (not necessarily maximal). Also, since by definition, the union of winning sets is still a winning set, the maximal winning set must be unique and contain all the winning sets. Therefore, $\{ W^*_i \}_{i \in \text{Qns}}$ must be the maximal solution of equations in (6) with respect to $\{ W^*_k \}_{k \in \text{Qns}}$. \hfill \QED