Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Lazy Safety Controller Synthesis with Multi-Scale Adaptive-Sampling Abstractions of Nonlinear Systems ⋆

Elena Ivanova * **Antoine Girard** *

* *Université Paris-Saclay, CNRS, CentraleSupélec*
Laboratoire des signaux et systèmes
91190, Gif-sur-Yvette, France
(e-mail: {elena.ivanova,antoine.girard}@l2s.centralesupelec.fr).

Abstract: In this paper, we present an abstraction-based approach to safety controller synthesis for continuous-time nonlinear systems. To reduce the computational burden associated with symbolic control approaches, we develop a lazy controller synthesis algorithm, which uses the incremental forward exploration of the symbolic dynamics, allowing us to restrict the controller synthesis computations to reachable states only. We propose using this algorithm with novel multi-scale abstractions, which also use adaptive time sampling. Transition duration is constrained by intervals that must contain the reachable set, which enables better control of the symbolic transitions as opposed to using transitions of predetermined duration. Implementation of the algorithm and controller refinement are discussed. We provide a simple example to illustrate these benefits of the approach.

Keywords: Safety, lazy controller synthesis, multi-scale abstraction, adaptive sampling.

1. INTRODUCTION

Robust control synthesis for nonlinear dynamical systems subject to state and input constraints is a challenging problem in modern control theory. To tackle this problem, computational techniques are used within an abstraction-based framework. This method consists in creating a finite-state abstraction (or a symbolic model) for a continuous or a hybrid system, a procedure where a controller synthesized for the abstraction can be refined to a controller for the original system (Belta et al. (2017), Tabuada (2009)). This approach makes it possible to use discrete controller synthesis techniques (Cassandras and Lafortune (2009), Cormen et al. (2001)), which in turn allow us to address a broad class of specifications given, for instance, by automata or temporal logic formula. In this paper, though, we focus on simple safety specifications, which consist in keeping the state of the system inside a given safe set. This type of specification often appears in real-world problems, e.g. temperature regulation in smart-buildings (Meyer et al. (2018b), Thavlov and Bindner (2015)), blood glucose rate control for diabetic patients (Kushner et al. (2019), Gillis et al. (2007)), safety of vehicle platoons (Saoud et al. (2019), Ames et al. (2017), Alam et al. (2014)), satellite station keeping (Weiss et al. (2018)), etc.

A key-issue in abstraction-based control is computational complexity. Indeed, symbolic models are usually obtained by partitioning or by discretizing the sets of states and inputs, and finer discretizations that provide more accurate abstractions result in symbolic models with a larger number of states and inputs, and thus require more time and memory for computations. Moreover, the computational complexity of the discrete controller synthesis algorithms typically depends on the size of symbolic models. Finally, while all these computations are typically handled off-line, a controller obtained via abstraction-based techniques using symbolic models with a large number of states would require a huge amount of memory for its real-time implementation. Possibilities to mitigate these intensive computational requirements are to refine abstractions iteratively (Gol et al. (2013); Nilson et al. (2017)), e.g. using multi-layered grids (Girard et al. (2016); Hsu et al. (2018b)). Starting from coarse abstractions, these are refined locally only where it is needed.

Symbolic models are often represented as finite transition systems. The computation of the transition relation is a demanding process, requiring over-approximation of the reachable sets. Several methods exist for computing such over-approximations (Girard (2005), Kurzhanski and Varaiya (2014), Reissig et al. (2017), Meyer et al. (2018a)), with a compromise between precision and simplicity of implementation. We can occasionally use structural properties of the dynamics such as monotonicity (Coogan and Arcak (2015)) or incremental stability (Girard et al. (2016)) to ease these computations. However, reducing the number of transitions that should be computed, renders the process less demanding. This is the purpose of lazy controller synthesis algorithms (Girard et al. (2016); Hussien and Tabuada (2018); Hsu et al. (2018a); Saoud et al. (2019)) where the transitions are explored as needed and computed on the fly during a controller synthesis. Given an ordering of transitions, transitions of lower priority are

⋆ This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 725144).

only explored if the specification cannot be enforced using transitions of higher priority.

In the first part of this paper, we present a novel lazy controller synthesis algorithm for safety (set invariance) specifications. A major difference with (Hussien and Tabuada (2018); Hsu et al. (2018a); Saoud et al. (2019)) is that our algorithm uses incremental forward exploration of the symbolic dynamics and thus allows us to restrict the controller synthesis computations to reachable states only. While this idea was used in (Girard et al. (2016)) for deterministic symbolic models, the algorithm presented in this paper makes it possible to deal with non-deterministic transition relations. In the second part of the paper, we propose a novel approach to compute multi-scale abstractions. Similar to (Girard et al. (2016); Hsu et al. (2018a)), our approach uses multi-layered grids on the state-space. The main contribution in this part is a novel method for adaptive time sampling. The duration of the transitions is constrained by a state interval that should contain the reachable set, as opposed to a predetermined duration. This provides the opportunity to control, approximately, where symbolic transitions end. The resulting abstraction is equipped with priorities on transitions, which makes it possible to use lazy controller synthesis algorithms.

This paper is organized as follows. In Section 2, we present a lazy safety controller synthesis algorithm for non-deterministic transition systems. In Section 3, we present an abstraction-based approach for synthesizing safety controllers for nonlinear systems. We introduce a type of multi-scale adaptive-sampling abstractions, which can be used in combination with lazy synthesis algorithms. Practical implementation of the algorithm and controller refinement are discussed. In Section 4, we consider a simple illustrative example to show the benefits of the approach.

2. LAZY SAFETY CONTROLLER SYNTHESIS

In this section, we present a lazy controller synthesis algorithm for finite transition systems and safety specifications.

Definition 1. A finite *transition system* is a tuple $\Sigma = (Q, U, F)$, consisting of a finite set of states Q, a finite set of inputs U, and a transition relation $F \subseteq Q \times U \times Q$.

For every transition $(q, u, q') \in F$ the state q is named *u-predecessor* of q' and similarly the state q' is named *u-successor* of q. For the set of all u-predecessors of the state q the notation $F^{-1}(q, u)$ is used, while the set of all u-successors of a state q is denoted by $F(q, u)$. If there is $q \in Q$, $u \in U$ such that $|F(q, u)| > 1$, then the transition system is called *non-deterministic*, otherwise it is *deterministic*.

Since $F(q, u)$ may be empty let us introduce a set $\mathrm{Enab}_F(q) = \{u \in U \mid F(q, u) \neq \emptyset\}$ of all enabled inputs at a state $q \in Q$. If $\mathrm{Enab}_F(q) = \emptyset$, then q is said to be *blocking*, otherwise it is *non-blocking*. We also use notation $\mathrm{Block}_F(Q')$ to describe the set of all blocking states in a set $Q' \subseteq Q$. If $\mathrm{Block}_F(Q) = \emptyset$, then the transition system is called *non-blocking*.

Definition 2. A *trajectory* of a transition system $\Sigma = (Q, U, F)$ is a finite or infinite sequence of transitions $q_0 \xrightarrow{u_0} q_1 \xrightarrow{u_1} q_2 \xrightarrow{u_2} q_3 \xrightarrow{u_3} \ldots$, s.t. $q^i \in Q, u^i \in U$ and $q^{i+1} \in F(q^i, u^i)$ for all $i \geq 0$.

A state $q' \in Q$ is *reachable* from the state q, if $q' = q$ or there exists a trajectory connecting them. The set of all reachable states from the state q is denoted by $\mathrm{Reach}_F(q)$. This definition could be naturally extended for a subset Q' of the set of states Q: $\mathrm{Reach}_F(Q') = \cup_{q \in Q'} \mathrm{Reach}_F(q)$.

2.1 Safety Controllers

Definition 3. A *controller* for a transition system $\Sigma = (Q, U, F)$ is a map $C: Q \to 2^U$, such that $C(q) \subseteq \mathrm{Enab}_F(q)$ for every $q \in Q$.

Let us use notation $\mathrm{Dom}(C) = \{q \in Q \mid C(q) \neq \emptyset\}$ for a domain of controller C. We also define the controlled transition relation $F_C = \{(q, u, q') \in F \mid u \in C(q)\}$.

Definition 4. A *safety controller* for a transition system $\Sigma = (Q, U, F)$ and a safe set $Q_s \subseteq Q$ is a controller C such that the following two properties holds

(1) $\mathrm{Dom}(C) \subseteq Q_s$;
(2) for all $q \in \mathrm{Dom}(C)$ and for all $u \in C(q)$ the inclusion $F(q, u) \subseteq \mathrm{Dom}(C)$ is satisfied.

For a given specification, there are usually several safety controllers, however, there is a unique maximal one (Tabuada (2009)):

Lemma 5. For a given transition system $\Sigma = (Q, U, F)$, a safety specification $Q_s \subseteq Q$ there is a unique maximal safety controller $\bar{C}$ such that for any safety controller C the following holds

(1) $\mathrm{Dom}(C) \subseteq \mathrm{Dom}(\bar{C})$;
(2) for all $q \in \mathrm{Dom}(C)$, $C(q) \subseteq \bar{C}(q)$.

The maximal safety controller $\bar{C}$ is the best possible safety controller in the sense that any other controller solving the same safety problem would be more restrictive. That justifies the following notion of controllability.

Definition 6. Let Q_s be a safe set. A state $q \in Q$ of the transition system $\Sigma = (Q, U, F)$ is *safety controllable* if and only if $q \in \mathrm{Dom}(\bar{C})$. The set of safety controllable states is denoted $\mathrm{Cont}(\Sigma, Q_s)$.

Though there is (see Tabuada (2009)) a simple fixed-point algorithm converging to the controller $\bar{C}$ for any given finite safe set Q_s, its computational complexity essentially depends on the number of safe states, making $\bar{C}$ too labour-intensive for many real-world problems. In light of this, in the next section, we relax, under certain assumptions, the maximality requirement for a desirable controller, while retaining other important properties.

2.2 Maximal Lazy Safety Controller

Let us suppose that the initial set $Q_{init} \subseteq Q$ is fixed. In this case, we have no interest in providing a controller for non-reachable states. Moreover, if a set of inputs U is equipped with a partial order, and for some state several inputs preserve safety, it would to be reasonable to keep only those which have the highest priority. Here we say that an input $u \in U$ has higher priority than $u' \in U$ if and only if $u \succ u'$.

Assumption 7. Let the set of inputs U be split into N non-intersecting subsets $U = U_1 \cup U_2 \ldots \cup U_N$ and for all $u' \in U_i,\ u'' \in U_j,\ i < j,\ i,j \in \{1,\ldots,N\}$ it holds that $u' \prec u''$, while inputs belonging to the same subset are considered as equivalent.

Definition 8. (Girard et al. (2016)). For a given transition system $\Sigma = (Q, U, F)$, a safety specification $Q_s \subseteq Q$, and a fixed initial set $Q_{init} \subseteq Q$ a *maximal lazy safety (MLS) controller* C^* is a safety controller satisfying the following properties

(1) $Q_{init} \cap \mathrm{Cont}(\Sigma, Q_s) \subseteq \mathrm{Dom}(C^*)$;
(2) $\mathrm{Dom}(C^*) \subseteq \mathrm{Reach}_{F_{C^*}}(Q_{init} \cap \mathrm{Dom}(C^*))$;
(3) for all states $q \in \mathrm{Dom}(C^*)$:
   (a) if $u \in C^*(q)$ then for all $u' \in \mathrm{Enab}_F(q)$ such that $u' \simeq u$, it holds that $u' \in C^*(q)$ if and only if $F(q,u') \subseteq \mathrm{Cont}(\Sigma, Q_s)$;
   (b) if $u \in C^*(q)$, then for all $u' \in \mathrm{Enab}_F(q)$ with $u \prec u'$, it holds that $F(q,u') \cap \mathrm{Cont}(\Sigma, Q_s) \neq F(q,u')$.

The term maximal comes from the fact that all safety controllable initial states are in $\mathrm{Dom}(C^*)$, and if the controller enables an input, it also enables all inputs which have the same priority and preserve safety. The term lazy refers to the fact that while several inputs can preserve safety, the controller enables only inputs with the highest priority. Hence, C^* represents a trade-off between maximal permissiveness and efficiency.

Theorem 9. (Girard et al. (2016)). For a given transition system $\Sigma = (Q, U, F)$, a finite safety specification $Q_s \subseteq Q$, and a fixed initial set $Q_{init} \subseteq Q$ there is a unique MLS controller C^*.

Intuitively, the MLS controller C^* can be obtained from the maximal safety controller $\bar{C}$ by keeping, for every state, only those enabled controls which have a higher priority, and by removing the states that are not reachable from initial states. Of course, it is not the best way to find C^* since it needs first to compute $\bar{C}$.

2.3 Efficient MLS Controller Synthesis

In this section, we provide a more efficient algorithm for computing the MLS controller, which is based on two ideas: to explore, for each state, inputs with a lower priority only if we failed to find a safe input with higher priority, and to set aside states that are non-reachable from the initial set.

To implement the first idea, we introduce for every state $q \in Q$ a notion of a state priority $p(q)$ and define for every given $p\colon Q \to \{0,\ldots,N\}$ a reduced transition relation F_p such that $(q,u,q') \in F_p$ if and only if $p(q) \in \{1,\ldots,N\}$, $u \in U_{p(q)}$, $(q,u,q') \in F$ and for all $q'' \in F(q,u)$ the equality $p(q'') \neq 0$ is satisfied. Intuitively, this means that for states with priority from 1 to N, only transitions with the same priority inputs are considered, while states with priority 0 are blocking and non-reachable. Starting with the highest priority for states in a safe set Q_s and with the lowest one for the others (line 2-6), we iteratively update the function p in the main block of Algorithm 1 (lines 7-12) until a corresponding transition relation F_p can be used as a basis for a desirable safety controller (lines 13-17). As concerns the second idea, only states that are reachable (regarding the current version of F_p) from initial set (lines 7, 12) are explored. Because of space limitations, the following result is stated without proof:

Algorithm 1: MLS Controller Synthesis

Input: $\Sigma = (Q,U,F)$, Q_{init}, Q_s
Output: MLS controller C

```
1  begin
2      for q ∈ Q do
3          if q ∈ Q_s then
4              p(q) := N;
5          else
6              p(q) := 0;
7      R := Reach_{F_p}({q ∈ Q_init | p(q) ≠ 0});
8      while Block_{F_p}(R) ≠ ∅ do
9          B := Block_{F_p}(R);
10         for q ∈ B do
11             p(q) := p(q) − 1;
12         R := Reach_{F_p}({q ∈ Q_init | p(q) ≠ 0});
13     for q ∈ Q do
14         if q ∈ R then
15             C(q) := Enab_{F_p}(q);
16         else
17             C(q) := ∅;
18     return C;
```

Theorem 10. Let C computed by Algorithm 1. Then, C is the MLS controller.

In simple words, Algorithm 1 is an alternation procedure between a forward state space exploration and a backward correction of the obtained transition system in order to satisfy the safety requirements. At the beginning of every iteration, we update for every state a set of enabled inputs, which depends on the priority of the state. Then we try to find a safety controller. If we fail, we reduce the priorities of all uncontrollable states by one point and start again.

3. ABSTRACTION BASED CONTROL SYNTHESIS

In this section, we consider a safety control problem for continuous-time nonlinear systems, and propose a solution based on Algorithm 1, combined with a novel type of multi-scale adaptive sampling abstractions.

3.1 Problem Formulation

A control system $\Sigma = (T, \mathbb{R}^n, U, W, f)$ consists of a time domain $T = [0, +\infty)$, a state space $\mathbb{R}^n$, a compact set $U \subset \mathbb{R}^m$, a compact set $W \subset \mathbb{R}^p$, and a non-linear function $f\colon \mathbb{R}^n \times U \times W \to \mathbb{R}^n$, such that for any control $u(\cdot) \in \mathcal{L}^\infty(T,U)$, any disturbance $w(\cdot) \in \mathcal{L}^\infty(T,W)$ and any initial condition $x(0) \in \mathbb{R}^n$ there is a unique solution $x_f(t \mid x(0), u(\cdot), w(\cdot)),\ t \in T$ of the following differential equation $\dot{x}(t) = f(x(t), u(t), w(t))$ in the sense of Caratheodory. The notation $\mathcal{L}^\infty(T,S)$ is used for the space of all measurable on T functions $s(\cdot)$ such that $s(t) \in S$, for almost all $t \in T$.

In this paper, we are looking for admissible safety controllers, which keep all trajectories of the closed-loop sys-

tem inside a safety set Y, while supposing that an initial set X_0 is known before computation. Here the controller is said to be admissible if it is robust against any measurable bounded disturbance $w(\cdot) \in \mathcal{L}^\infty(T, W)$ and a solution of the closed-loop system exists. To synthesize such a controller, we use an abstraction-based approach. In the next section, the continuous system $\Sigma = (T, \mathbb{R}^n, U, W, f)$ is abstracted by a finite transition system $\Sigma_A = (Q_A, U_A, F_A)$, which approximates the behavior of the original plant. Then, we explain how to refine the controller for the abstraction (which could be found using Algorithm 1) to a controller for the concrete system.

3.2 Symbolic Model for the Original Plant

Let us first remark that all the vector operations appearing in this section are considered as component-wise.

We start the construction of a symbolic model for the original plant $\Sigma = (T, \mathbb{R}^n, U, W, f)$ by discretizing the state space $\mathbb{R}^n$. Let $X = [\underline{x}, \overline{x})$ be an n-dimensional interval including a safe set Y. Given a number $L > 0$, and a state space sampling parameter $n_x > 0$, $n_x \in \mathbb{N}^n$ we introduce for every $l = 1, \ldots, L$ a uniform partition

$$Q_l = \{q \in 2^X \mid \exists z \in \mathbb{Z}^n \text{ s.t. } q = [\underline{x} + z * \eta_l, \underline{x} + (z+1) * \eta_l)\}$$

where $\eta_l = (\overline{x} - \underline{x})/(2^{l-1} n_x)$ and we associate every state $x \in X$ with a unique cell $q \in Q_l$ such that $x \in q$. We also define for every $q \in Q_l$, the center of the interval $q_c = \underline{x} + (z + \frac{1}{2}) * \eta_l$. Depending on the context, we regard an element $q \in Q_l$ either as an atomic symbol, representing an infinite number of states from X, or as a subset of X. It is obvious from the definition that for any $q_i, q_j \in Q_l$, $q_i \neq q_j$ the intersection $q_i \cap q_j = \emptyset$ and that $X = \cup_{q \in Q_l} q$.

Proposition 11. For any $l = \{1, \ldots, L-1\}$ and for any $q \in Q_l$ there exists a unique set a states $\{q_i\}_{i=1}^{2^n}, q_i \in Q_{l+1}$ such that $q = \cup_{i=1}^{2^n} q_i$.

Regarding the safety specification, an element $q \in Q_l$ is a safe state if and only if $q \subseteq Y$, an initial state if and only if $q \cap X_0 \neq \emptyset$. Let us denote by $Q_{l,s}$ a set of all safe states in Q_l, and by $Q_{l,init}$ a set of all initial states.

We then combine all these partitions into a multilayered grid $Q_A = \cup_{l=1}^{L} Q_l$ and use it as a set of states for the abstraction $\Sigma_A = (Q_A, U_A, F_A)$. Considering that for any $q \in Q_A$ there exists a unique layer $Q_l \subset Q_A$ such that $q \in Q_l$, let us define a function $l : Q_A \to \{1, \ldots, L\}$ returning the index of the layer to which q belongs. We also introduce an input *up*, allowing switching from the current layer to the previous (coarser) one.

In the next step, a control set U is approximated by its finite subset $U_\mu \subseteq U$. For every $q \in Q_l, u_\mu \in U_\mu$ we consider the set of all reachable states at the time $t \in T$:

$$\begin{aligned}\text{Reach}(t \mid q, u_\mu) = \{x \in \mathbb{R}^n \mid \exists\, x(0) \in q \\ \text{and } \exists\, w(\cdot) \in \mathcal{L}^\infty([0,t], W) \text{ such that} \\ x_f(t \mid x(0), u^*(\cdot), w(\cdot)) = x\},\end{aligned}$$

corresponding to an initial set q, a constant control function $u^* : [0, t] \to u_\mu$ and all admissible disturbances $w(\cdot)$. Since exact computation of reach set is seldom possible, an over-approximation $\overline{\text{Reach}}(t \mid q, u_\mu)$ is commonly used to built symbolic models.

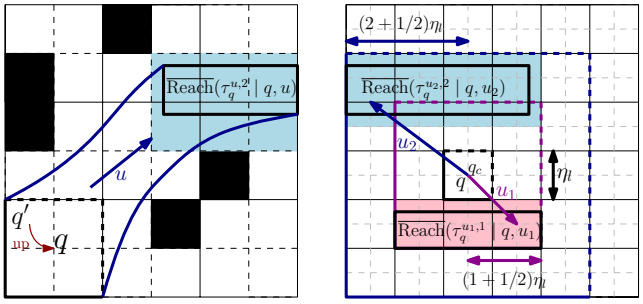

Fig. 1. Transitions on a 2-layered grid. Left figure illustrates (1), (2). Right figure illustrates (3).

In this paper, a multi-scale symbolic model is considered, which means that for every given state $q \in Q_A$ and control $u_\mu \in U_\mu$ we construct n_p transition with different duration. Let us now define a transition relation F_A for every available input in the set $U_A = (U_\mu \times \{1, \ldots, n_p\}) \cup \{up\}$.

- for any $q' \in Q_A \setminus Q_1$ the transition $(q', up, q) \in F_A$ if and only if $q \in Q_{l(q)-1}$ and $q \subset q'$. See Fig.1 (left).
- for any $q \in Q_A$ and any $(u_\mu, j) \in U_\mu \times \{1, \ldots, n_p\}$ the transition $(q, (u_\mu, j), q') \in F_A$ if and only if

$$q' \in Q_L,\ q' \cap \overline{\text{Reach}}(\overline{\tau}_q^{u_\mu, j} \mid q, u_\mu) \neq \emptyset, \qquad (1)$$

and the condition of a collision avoidance

$$\overline{\text{Reach}}(t \mid q, u_\mu) \cap (X \setminus Y) = \emptyset, \quad t \in [0, \overline{\tau}_q^{u_\mu, j}] \qquad (2)$$

is satisfied. Here $\overline{\tau}_q^{u_\mu, j} = \min(\tau_l, \tau_q^{u_\mu, j} - \varepsilon)$, where τ_l is a given parameter which determines the maximal evolution time allowed at this layer, while $\tau_q^{u_\mu, j}$ is a moment in time

$$\begin{aligned}\tau_q^{u_\mu, j} = \inf_{t \in [0, +\infty)} \Big\{ &\overline{\text{Reach}}(t \mid q, u_\mu) \not\subset \\ &\not\subset [q_c - (j + 1/2) * \eta_l, q_c + (j + 1/2) * \eta_l) \Big\} \qquad (3)\end{aligned}$$

when the over-approximation of a reachable set leaves the interval with a radius $(j+1/2)\eta_l$ and a center in q_c. We chose $\varepsilon < \tau_q^{u_\mu, j}$ arbitrary small to stop evolution just before leaving, while τ_l should be big enough since it serves only to manage situations where a solution is stuck within the box. In the Fig.1 there is illustration of the idea.

Such a definition of transition duration using adaptive time-sampling is a contribution of this paper. Instead of using prescribed time sampling parameters (Girard et al. (2016), Hsu et al. (2018b)), this approach allows us to control, approximately, where symbolic transitions finish and better analyze the behaviour of the system. At the same time, the collision avoidance condition allows us to eliminate solutions, which start and end in the safe set but passes some unsafe regions during the evolution. However, since the intersection of the reachable set with the grid is checked at every instant of time, the authors recommend using simple interval over-approximations (Reissig et al. (2017), Moor and Raisch (2002), Meyer et al. (2018a), Zamani et al. (2011), Maidens and Arcak (2014)), trading accuracy for simplicity of implementation. We also finish any non-*up* transition at the finest layer to be more flexible while moving close to the obstacles.

Since the idea of working with a multi-layered grid is to explore states from a coarser level before exploring states from a finer one, control up has the highest priority. We also prefer a transition with a longer duration to a transition with a shorter one. That is why it is reasonable to define a partial order on $U_\mu \times \{1, \dots, n_p\}$ as follows: $(u_\mu, j_1) < (u'_\mu, j_2)$ if and only if $j_1 < j_2$ for all $j_1, j_2 \in \{1, \dots, n_p\}, u_\mu, u'_\mu \in U_\mu$. As illustration in the Fig.1(right) a control $(u_2, 2)$ is more prioritized than a control $(u_1, 1)$. Splitting a set U_A into $n_p + 1$ equivalence classes with respect to the introduced partial order and defining a safe set $Q_{A,s} = \cup_{l=1}^{L} Q_{l,s}$ and an initial set $Q_{A,init} = Q_{L,init}$ we can use now the Algorithm 1 to find a maximal lazy safety controller. Let us underline here that Algorithm 1 does not require that the transition relation F_A is pre-computed and we calculate transitions on the fly and only for reachable states.

3.3 From a Multilayered to an Adaptive Grid

Though the introduction of an artificial input up allows us to manipulate a multi-layered grid and transition duration using one general framework, the abstraction considered in the previous section has two significant disadvantages. First, when we run Algorithm 1, we have to store a multilayered grid Q_A, while it is better to work with an adaptive grid, consisting of cells with different sizes. Second, for every fixed distribution of priorities of states $p\colon Q_A \to \{0, \dots, n_p + 1\}$ a reduced transition relation $F_{A,p}$ (see section 2.2 for the definition) includes a lot of auxiliary transitions which serves only for switching from a finer layer to a coarser one, while we prefer to keep in memory only those transitions which are directly related to the dynamic of the original system.

For a given p let us choose as an adaptive grid $Q_{*,p} = Q^p_{up}(Q_L)$, where $Q^p_{up}\colon Q_L \to Q_A$ is defined as follows

$$\begin{aligned} Q^p_{up}(q) = \{q' \in Q_A \mid F_{A,p}(q', up) = \emptyset \text{ and} \\ \exists \{q_i\}_{i=0}^{N}, q_i \in Q_A, i = \overline{0, N} \text{ s. t. } q_0 = q, \\ q' = q_N, q_{j+1} = F_{A,p}(q_j, up), j = \overline{0, N-1}\} \end{aligned}$$

and its extension for all $Q \subseteq Q_L$ is $Q^p_{up}(Q) = \cup_{q \in Q} Q^p_{up}(q)$. So, a state of multilayered grid Q_A is included in $Q_{*,p}$ if and only if it is reachable from the finest layer only with up transitions and there is no possibility to go higher for this state. Let us remark that for any $q_i, q_j \in Q_{*,p}$, $q_i \neq q_j$ the following $q_i \cap q_j = \emptyset$ is satisfied, moreover $X = \cup_{q \in Q_{*,p}} q$.

For the associated transition relation let us introduce $F_{*,p} \subseteq Q_{*,p} \times U_A \times Q_{*,p}$ such that $(q, u, q') \in F_{*,p}$ if and only if $q' \in Q^p_{up}(F_{A,p}(q, u))$. Its definition is correct because for any $q \in Q_{*,p}, u \in U_A$ the set $F_{A,p}(q, u)$ either empty, or included in Q_L.

The following propositions show why we can operate with a grid $Q_{*,p}$ and a transition relation $F_{*,p}$, instead of Q_A and $F_{A,p}$, respectively.

Proposition 12. For any $q \in Q_A$, $q \in R^*$, where

$$R^* = \mathrm{Reach}_{F_{*,p}}(Q^p_{up}(\{q \in Q_{A,init} \mid p(q) \neq 0\}))$$

if and only if $q \in R$, where

$$R = \mathrm{Reach}_{F_{A,p}}(\{q \in Q_{A,init} \mid p(q) \neq 0\})$$

and $F_{A,p}(q, up) = \emptyset$.

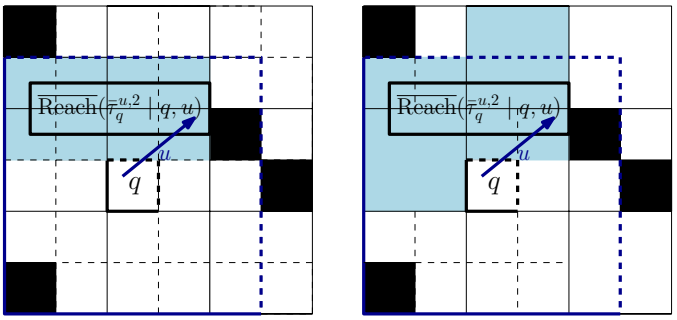

Fig. 2. Left: 2-layered grid. Right: adaptive grid

Proposition 13. Let us run Algorithm 1 for the transition system $\Sigma_A = (Q_A, U_A, F_A)$. The following

$$\mathrm{Block}_{F_{A,p}}(R) = \mathrm{Block}_{F_{*,p}}(R^*)$$

is satisfied at every iteration of loop 8-12.

So, we can reinitialize the priority of states p (line 11), using only knowledge about $Q_{*,p}$ and $F_{*,p}$. Now we explain how to update the adaptive grid $Q_{*,p}$ and a transition relation $F_{*,p}$, while changing p during the evolution of Algorithm 1.

After execution of lines 2-6, a state in Q_A is reachable from the finest layer if and only if it is safe or belonging to Q_L. Hence, if we start from the coarsest layer Q_1 and recursively split all unsafe states into 2^n pieces while it is possible (i.e. while they do not belong to the finest layer Q_L) we finally get a grid $Q_{*,p}$, corresponding to the distribution of priority of states p just before the loop 8-12 execution. Since with every iteration of the loop only states included in $Q_{*,p}$ change their priorities (see Proposition 13), we can also update our adaptive grid $Q_{*,p}$ without direct usage of the transition relation $F_{A,p}$. Indeed, if a state $q \in Q_A \setminus Q_L$ gets a priority 0, then, from Lemma 11 and definition of up transition, it follows that we should replace it by 2^n states $q'_i \in Q_{l(q)+1}$, such that $q = \cup_{i=1}^{2^n} q'_i$.

There also exists a way to compute the transition relation $F_{*,p}$, using only current version of $Q_{*,p}$.

Proposition 14. For any $q \in Q_{*,p}, u \in U_A \setminus \{up\}$ a state $q' \in Q_{*,p}$ belongs to a set $F_{*,p}(q, u)$ if and only if

$$q' \cap \overline{\mathrm{Reach}}(\overline{\tau}_q^{u_\mu, j} \mid q, u_\mu) \neq \emptyset,$$

and the condition of a collision avoidance

$$\overline{\mathrm{Reach}}(t \mid q, u_\mu) \cap (X \setminus Y) = \emptyset$$

is satisfied for all $t \in [0, \overline{\tau}_q^{u_\mu, j}]$.

We illustrate the difference between a multilayered grid and an adaptive grid in Fig. 2. There, unsafe states are marked with black boxes, and successors filled with blue color.

Finally, we show that we can fully simulate the main part of Algorithm 1, using only adaptive grid $Q_{*,p}$ and a transition relation $F_{*,p}$. It also important to mention that the abstraction is not required to be pre-computed, but constructed on the fly. Let us write, $C_*(q) := \mathrm{Enab}_{F_{*,p}}(q)$ for all $q \in R^*$ and empty otherwise, for a controller initialization part (lines 13-17). It is obvious, that $C_*(q)$ is a safety controller for a transition system $\Sigma_* = (Q_*, U_A, F_*)$, where a state space Q_* and a transition relation F_* are

correspondingly $Q_{*,p}$ and $F_{*,p}$ after we exit the loop. Moreover, since for any $p: Q_A \to \{0, \dots, n_p + 1\}$, any $q \in Q_{*,p}$ the set $F_{*,p}(q, up) = \emptyset$, the input up never appears in a final safety controller C_*. So, we can just skip it from the earlier beginning by initializing in line 4 every safe state with priority $N-1$, instead of N.

3.4 Refinement of the controller

In the previous section, we computed a safety controller $C_*(q)$ for a transition system $\Sigma_* = (Q_*, U_\mu \times \{1, \dots, n_p\}, F_*)$, which is an abstraction for the original plant. Let us now provide a safety controller for the system $\Sigma = (T, \mathbb{R}^n, U, W, f)$.

First of all we introduce a controller $C_*^{dur} : Q_* \to U_\mu \times T$, such that for every $q \in Q_* \setminus \mathrm{Dom}(C_*)$ we say $C_*^{dur}(q) = \emptyset$, while if $q \in \mathrm{Dom}(C_*)$ and $(u_\mu, j) \in \mathrm{Enab}_{F_*}(q)$, then the pair $(u_\mu, \overline{\tau}_q^{u_\mu, j}) \in C_*^{dur}(q)$. Hence, a controller C_*^{dur} store real durations of safe transitions, instead of the sizes of the boxes.

We then explain with the following proposition how to refine the controller C_*^{dur} to a safety controller u for the original continuous system.

Proposition 15. There exists a unique trajectory of the closed-loop system

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), w(t)),\ t \in T \\ \dot{u} = 0,\ t \in [t_k, t_{k+1}) \\ t_{k+1} = t_k + \tau \\ (u(x(t_k), t_k), \tau) = C_*^{dur}(q), \text{where } q \in Q_* \text{ s.t. } x(t_k) \in q \\ t_0 = 0 \\ x(t_0) = x_0,\ x_0 \in X_0 \cap \mathrm{Dom}(C_*^{dur}) \end{cases}$$

and it remains inside the safe set Y, no matter which disturbance $w(\cdot) \in \mathcal{L}(T, W)$ has been applied.

The statement of the proposition directly follows from the discussion above. We only remark here, that the computation of the controller for the abstraction is usually implemented off-line and then stored in a control device memory, while a controller for the continuous system is calculated online using this pre-computed information.

4. NUMERICAL ILLUSTRATION

In this section, we consider a problem of temperature regulation in a two-room building. Each room is equipped with a heater and T_i is the temperature in the room i, $i = 1, 2$. The evolution of the temperatures is described by the following system of the differential equations

$$\dot{T}_1 = \alpha(T_2 - T_1) + \beta_1(t_e - T_1) + \gamma_1(t_{h_1} - T_1)u_1$$
$$\dot{T}_2 = \alpha(T_1 - T_2) + \beta_2(t_e - T_2) + \gamma_2(t_{h_2} - T_2)u_2$$

Here t_e is the temperature of the external environment of the building, t_{h_1}, t_{h_2} are temperatures of the heaters, α is the conduction factor between rooms, β_1, β_2 are conduction factors between external environment and the first room and the second room respectively, γ_1, γ_2 are conduction factor between heater and rooms. Control parameter u_i equals 1 if the room i is heated and 0 otherwise. Temperature t_e is considered as a bounded disturbance.

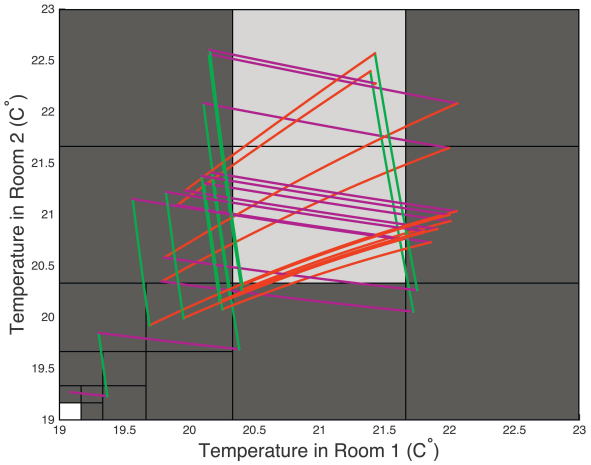

Fig. 3. Temperature regulation in two rooms building.

We run our simulation for the following set of parameters

$$\alpha = 1/2 * 10^{-4}\, W/J,\ \beta_1 = 1/6 * 10^{-4}\, W/J,$$
$$\beta_2 = 1/11 * 10^{-4}\, W/J,\ \gamma_1 = 1.5 * 10^{-4}\, W/J,$$
$$\gamma_2 = 1.5 * 10^{-4}\, W/J,\ t_{h_1} = 30\, C^\circ,\ t_{h_1} = 40\, C^\circ.$$

A safety specification is given by an initial set $X_0 = [19, 23] \times [19, 23]$, safe set $Y = [19, 23] \times [19, 23]$, and a disturbance $t_e \in [-10, 10]$. We also suppose that at given instant at most one heater is switched on, i.e. a control set $U = \{\{0, 0\}, \{0, 1\}, \{1, 0\}\}$. For the abstraction, we chose $L = 4$, $n_x = [4; 4]$, $U\mu = U$, and $n_p = 2$.

In Fig.3, the results of the simulation are provided. We use a dark grey and a light grey for states, which are controllable with a $(u_\mu, 1)$, $u_\mu \in U_\mu$ and with a $(u_\mu, 2)$, $u_\mu \in U_\mu$ correspondingly, while white region is uncontrollable. The closed loop trajectory is simulated for 24 hours, supposing that external temperature varies between $-10C^\circ$ and $10C^\circ$ and initial point $x_0 = [19.084; 19.27]$. The orange color correspond to a control $\{0, 0\}$, green to a control $\{0, 1\}$, violet to a control $\{1, 0\}$.

To evaluate efficiency, we also run a simulation for two extreme cases: $L = 1, n_x = [4; 4]$ and $L = 1, n_x = [25; 25]$, which correspond to a coarsest grid and to a finest grid of considered four-layered adaptive grid. One can see the comparison of the results in Tab.1. The controllable sets coincides for the adaptive grid and the finest grid, but we get a noticeable time and memory gain. The implementations has been done in C++, processor Intel Core i7-8700, 2.5 Hg, RAM 16 GB.

Table 1. Numerical results

Grid	Number of states	Time	Cont. Ratio
Adaptive grid	18	7 s	98%
Coarsest grid	9	5 s	89%
Finest grid	625	50 s	98%

5. CONCLUSION

In this paper, we introduced a new method of construction of symbolic models for a continuous dynamic system, based on the adaptive sampling of time and multi-layered state-space discretization. Then, we proposed an

efficient way of computing a safety controller for the nondeterministic transition system. The lazy algorithm is based on the forward exploration of the state space, while the term lazy refers to the fact, that if we can guarantee safety using a higher priority input, we do not explore inputs with smaller priorities at all.

Future work will focus on extending the approach to other types of specification such as reachability or more general properties specified by automata or temporal logical formula.

REFERENCES

Alam, A., Gattami, A., Johansson, K.H., and Tomlin, C.J. (2014). Guaranteeing safety for heavy duty vehicle platooning: Safe set computations and experimental evaluations. *Control Engineering Practice*, 24, 33–41.

Ames, A.D., Xu, X., Grizzle, J.W., and Tabuada, P. (2017). Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8), 3861–3876.

Belta, C., Yordanov, B., and Ǵol, E.A. (2017). *Formal Methods for Discrete-Time Dynamical Systems.* Springer.

Cassandras, C.G. and Lafortune, S. (2009). *Introduction to Discrete Event Systems, 2nd ed.* Springer.

Coogan, S. and Arcak, M. (2015). Efficient finite abstraction of mixed monotone systems. *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, 58–67.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, C. (2001). *Introduction to Algorithms, 2nd ed.* MIT Press.

Gillis, R., Palerm, C.C., Zisser, H., Jovanovič, L., Seborg, D.E., and Doyle, F.J. (2007). Glucose estimation and prediction through meal responses using ambulatory subject data for advisory mode model predictive control. *Journal of Diabets Science and Technology*, 1(6), 825—-833.

Girard, A. (2005). Reachability of uncertain linear systems using zonotopes. *Hybrid Systems: Computation and Control. HSCC 2005. Lecture Notes in Computer Science*, 3414, 291—-305.

Girard, A., Gössler, G., and Moueli, S. (2016). Safety controller synthesis for incrementally stable switched systems using multiscale symbolic models. *IEEE Transactions on Automatic Control*, 61(6), 1537–1549.

Gol, E.A., Lazar, M., and Belta, C. (2013). Language-guided controller synthesis for linear systems. *IEEE Transactions on Automatic Control*, 59(5), 1163–1176.

Hsu, K., Majumdar, R., Mallik, K., and Schmuck, A.K. (2018a). Lazy abstraction-based controller synthesis. *IEEE Conference on Decision and Control (CDC)*, 4902–4907.

Hsu, K., Mallik, K., Majumdar, R., and Schmuck, A.K. (2018b). Multi-layered abstraction-based controller synthesis for continuous-time systems. *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control*, 120–129.

Hussien, O. and Tabuada, P. (2018). Lazy controller synthesis using three-valued abstractions for safety and reachability specifications. In *2018 IEEE Conference on Decision and Control (CDC)*, 3567–3572. IEEE.

Kurzhanski, A.B. and Varaiya, P. (2014). *Dynamics and Control of Trajectory Tubes. Theory and Computation.* Birkhäuser.

Kushner, T., Bequette, B.W., Cameron, F., Forlenza, G., Maahs, D., and Sankaranarayanan, S. (2019). Models, devices, properties, and verification of artificial pancreas systems. *Automated Reasoning for Systems Biology and Medicine*, 825—-833.

Maidens, J. and Arcak, M. (2014). Reachability analysis of nonlinear systems using matrix measures. *IEEE Transactions on Automatic Control*, 60(1), 265–270.

Meyer, P.J., Coogan, S., and Arcak, M. (2018a). Sampled-data reachability analysis using sensitivity and mixed-monotonicity. *IEEE Control Systems Letters*, 2(4), 761–766.

Meyer, P.J., Girard, A., and Witrant, E. (2018b). Compositional abstraction and safety synthesis using overlapping symbolic models. *IEEE Transactions on Automatic Control*, 63(6), 1835–1841.

Moor, T. and Raisch, J. (2002). Abstraction based supervisory controller synthesis for high order monotone continuous systems. *Modelling, Analysis, and Design of Hybrid Systems, Springer*, 247—-265.

Nilson, P., Ozay, N., and Liu, J. (2017). Augmented finite transition systems as abstractions for control synthesis. *Discrete Event Dynamic Systems*, 27(3), 301–340.

Reissig, G., Weber, A., and Rungger, M. (2017). Feedback refinement relations for the synthesis of symbolic controllers. *IEEE Transactions on Automatic Control*, 62(4), 1781—-1796.

Saoud, A., Ivanova, E., and Girard, A. (2019). Efficient synthesis for monotone transition systems and directed safety specifications. *IEEE Conference on Decision and Control*.

Tabuada, P. (2009). *Verification and control of hybrid systems: a symbolic approach.* Springer.

Thavlov, A. and Bindner, H.W. (2015). A heat dynamic model for intelligent heating of buildings. *International Journal of Green Energy*, 12(3), 240–247.

Weiss, A., Kalabić, U.V., and Cairano, S.D. (2018). Station keeping and momentum management of low-thrust satellites using mpc. *Aerospace Science and Technology*, 76, 229–241.

Zamani, M., Pola, G., Mazo, M., and Tabuada, P. (2011). Symbolic models for nonlinear control systems without stability assumptions. *IEEE Transactions on Automatic Control*, 57(7), 1804–1809.