Guaranteed safe controller synthesis for switched systems using analytical solutions*

Martijn A. Goorden[1], Kim G. Larsen[1], Jesper E. Nielsen[2], Thomas D. Nielsen[1], Weizhu Qian[1], Michael R. Rasmussen[2], and Guohan Zhao[2]

***Abstract*— In this paper we present a method for synthesizing safe controllers for continuous-time sampled switched systems, where the analytical solution for the state trajectories is available. The method creates integer-valued lower and upper bounds on the evolution of the system, so that the tool UPPAAL TIGA can be used to synthesize a guaranteed safe controller for the continuous system. We compare our method to the Euler method, which our method is based upon, using two case studies covering cruise control and management of storm water detention ponds. The cruise control example shows that the proposed method can obtain tighter bounds, allowing the controller to be less conservative. The industrial case study on storm water detention ponds shows that we can obtain reasonable bounds in situations where the Euler method fails to obtain them.**

I. INTRODUCTION

Safety-critical systems have become ubiquitous where system failures can result in significant damages or even human fatalities. Hence, increasing emphasis is put into designing correct controller software to ensure safe operation of these systems. Formal methods like controller synthesis are concerned with obtaining correct-by-construction controllers, thereby improving the safety guarantees of the system.

In this work we consider continuous-time sampled switched systems, which is a sub-class of hybrid systems. In these systems, the dynamics of the system is time driven, often given by differential equations, while discrete-state changes, like the control mode, only occur periodically at fixed intervals. Models of switched systems have been used across several domains, such as automotive [1]–[3], green housing [4]–[6], power electronics [7], and urban water management [8]. The goal is to synthesize control strategies for continuous-time sampled switched systems that are guaranteed to be safe, where safety is defined in terms of strategies based on which the system avoids reaching a predefined set of unsafe states.

Recent advances in control synthesis for switched systems involve the use of symbolic methods, where the aim is to represent the continuous and infinite state-space of the system with a finite number of symbols, like discrete points [9], [10] or sets of states [11]–[13]. These approaches are well suited for safety critical systems. Incorporating uncontrollable components is however more challenging. Methods like [2], [14]–[16] model the adversary as a bounded perturbation, often resulting in too pessimistic safety controllers.

In [3] a method is presented to abstract a switched system into a timed game, so that the tool UPPAAL TIGA [17] can synthesize a safe strategy for the original system. A timed game requires that all bounds in the system are integer valued. Therefore, [3] introduces a guaranteed Euler method based on [18]. Instead of using a standard scheme of discretization for computing the successor state, [3] uses a guaranteed set-based Euler scheme in combination with lower and upper integer approximations.

While [3] shows promising results for the cruise control example from [19], we experience problems obtaining useful under and upper approximations of the continuous dynamics for the storm water detention pond case study from [8]. The bounds explode so quickly that they cannot be stored as integer values in memory, thus rendering the method useless for this use case. One reason for this could be that the guaranteed sets in [3] are spheres, i.e., the approximation error at any point in time is equal in all states variables of the system. But this is not always realistic to have. For example, when it is not raining, the rain intensity can be precisely represented with the integer 0 regardless of the approximation errors for other parts of the system. Proper dimension scaling cannot prevent approximation errors from 'spoiling' over to other state variables.

In this paper, we propose a new method to obtain lower and upper bound integer approximations for continuous-time sampled switched systems for which analytical solutions are available describing the trajectories of the state of the system. The new method uses the analytical solutions of the state trajectories, so can relax the idea of having a guaranteed spherical approximation set in [3] and instead obtain integer approximations for each individual state variable. This allows the bounds to be stricter in those state variables where it is possible. Our proposed method uses the same implementation idea as with the Euler method, enabling a seamless integration in UPPAAL TIGA. Experimental results show that we obtain the tightest possible integer under and upper approximations for the cruise control example, allowing UPPAAL TIGA to obtain a more optimistic safe controller than previously possible. Our proposed method obtains useful bounds for the stormwater pond case study, i.e., the approximation bounds are not exploding any more. Thus a guaranteed safe controller can be obtained for fixed

*This work is partly supported by the Villum Synergy project CLAIRE and the Villum Investigator project 4SOS.

[1] Martijn A. Goorden, Kim G. Larsen, Thomas D. Nielsen and Weizhu Qian are with the Department of Computer Science, Aalborg University, 9220 Aalborg East, Denmark {mgoorden, kgl, tdn, wqian}@cs.aau.dk

[2] Jesper E. Nielsen, Michael R. Rasmussen and Guohan Zhao are with the Department of the Built Environment, Aalborg University, 9220 Aalborg East, Denmark {jen, mmr, guohanz}@build.aau.dk

horizons.

The paper is structured as follows. Section II introduces the preliminaries for this paper: continuous-time sample switched systems modeled as a hybrid Markov decision process, synthesis for timed games, and the Euler-based method from [3]. We present our new method in Section III where we derive integer-valued bounds for cases where explicit solutions exist. Then, in Section IV we apply the new method on the cruise control example and compare experimental results with those obtained with the Euler-based method. In Section V we apply the new method successfully to the stormwater pond industrial case study, so that we actually can obtain a guaranteed safe controller. Finally, Section VI concludes the paper.

II. Preliminaries

A. Hybrid Markov decision processes

We apply the mathematical modeling framework of hybrid Markov decision process (HMDP), adapted from [4], [20], for modeling a hybrid switched system. This notion of an HMDP describes an uncountable and infinite state Markov Decision Process, see [21], where both the controller mode and environment mode switches periodically with interval $\tau \in \mathbb{R}_{\geq 0}$.

Definition 1: A *hybrid Markov decision process* (HMDP) $\mathcal{M}$ is a tuple (C, U, X, F, δ) where:

- the controller C is a finite set of (controllable) modes $C = \{c_1, \dots, c_k\}$,
- the uncontrollable environment U is a finite set of (uncontrollable) modes $U = \{u_1, \dots, u_l\}$,
- $X = \{x_1, \dots, x_n\}$ is a finite set of continuous (real-valued) variables,
- for each $c \in C$ and $u \in U$, the flow function $F_{c,u} : \mathbb{R}_{\geq 0} \times \mathbb{R}^X \rightarrow \mathbb{R}^X$ describes the evolution of the continuous variables over time in the combined mode (c, u).
- ρ is a family of density functions $\rho_\gamma : U \rightarrow [0, 1]$, where $\gamma = (c, u, \boldsymbol{x})$ is a global configuration with $\boldsymbol{x} : X \rightarrow \mathbb{R}$ being a valuation. More precisely, $\rho_\gamma(u')$ is the probability that in the global configuration $(c, u, \boldsymbol{x})$ the uncontrollable mode u will switch to mode u' at the end of the switching period.

In the rest of the paper, we denote by $\mathbb{C}$ the set of global configurations $C \times U \times (X \rightarrow \mathbb{R})$ of an HMDP.

For continuous-time systems, the flow function F is often defined as the solution to a set of (nonlinear) differential equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x} = f_{c,u}(\boldsymbol{x}) \tag{1}$$

A *run* of a switched HMDP is a sequence $\pi \in \mathbb{C}\mathbb{C}^*$ of configurations, starting with the initial configuration γ_0:

$$\pi = \gamma_0\gamma_1\gamma_2 \cdots$$

where $\gamma_i = (c_i, u_i, \boldsymbol{x}_i)$ and for all i

1) the continuous states evolve as $\boldsymbol{x}_{i+1} = F_{c_i,u_i}(\tau, \boldsymbol{x}_i)$,
2) the environment changes to any possible new mode, i.e., $u_{i+1} \in U$ and $\rho_{(c_i,u_i,\boldsymbol{x}_{i+1})}(u_{i+1}) > 0$, and
3) the controller changes to any possible new mode, i.e., $c_{i+1} \in C$.

For a given HMDP, a memoryless and possibly nondeterministic *strategy* σ determines which of the control modes can be used in the next period. Formally, a strategy is a function $\sigma : \mathbb{C} \rightarrow 2^C$ that returns a nonempty set of allowed control modes in a configuration. A strategy is called *deterministic* if exactly one control mode is permitted in each configuration.

The behavior of an HMDP $\mathcal{M}$ under supervision of a strategy σ is defined as follows. A run π is *according to the strategy* σ if the controller changes mode according to the strategy σ, i.e., $c_{i+1} \in \sigma((c_i, u_i, \boldsymbol{x}_{i+1}))$. A strategy σ is called *safe* with respect to a set of states $S \subseteq \mathbb{R}^X$ if for any run π according to σ all states encountered are within the safe set S, i.e., for all i and $\forall t \in [0, \tau]$ it must hold that $F_{c_i,u_i}(t, \boldsymbol{x}_i) \in S$. A safe strategy is called *maximally permissive* if for each configuration it returns the largest set of possible actions [22].

B. Synthesis for games

UPPAAL TIGA is able to synthesize controllers for (timed) games, where it can be either a reachability game or a safety game. Timed games are represented by (a network of) timed game automata, which are an extension to timed automata. Conceptually, a timed game automata consists of locations, clocks, discrete variables, actions (partitioned into controllable and uncontrollable actions), transitions between locations, and location invariants. The reader is referred to [23] for the formal definition of timed game automata.

A switching HMDP is related to a timed game automaton as follows. The sets of controller modes and environment modes can be directly mapped to locations. Continuous variables with a fixed derivative of 1 are clocks, and can thus directly be transferred to timed game automata. Furthermore, the switching period τ can be tracked by a new global clock (e.g. x_τ). Controllable and uncontrollable mode switching can be represented by transitions that are only enabled at the switching times. Finally, all other continuous variables need to be approximated by discrete variables, like integers. For example, one could round the continuous state to the nearest integer at each switching interval [19].

A control strategy for a timed game is considered to be safe if all reachable states remain within the set of safe states. In UPPAAL TIGA, the set of safe states can be specified using a fragment of timed computational tree logic (TCTL) [17].

C. Euler method

In [3], a set-based Euler method is proposed to synthesize guaranteed safe controllers for continuous-time switched systems with UPPAAL TIGA. The continuous dynamics of the system are approximated by the explicit forward Euler scheme of order 1: given a configuration $\gamma = (c, u, \boldsymbol{x})$ at the start of a period where $\boldsymbol{x} \in S$, the linear approximate

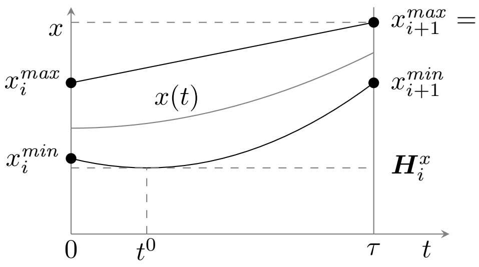

Fig. 1. The purpose of $\boldsymbol{x}^{min}$, $\boldsymbol{x}^{max}$, $\boldsymbol{H}$, and $\boldsymbol{G}$ for variable x.

solution $\tilde{\phi}(t,\gamma)$ for $t \in [0,\tau]$ is

$$\tilde{\phi}(t,\gamma) = \boldsymbol{x} + t f_{c,u}(\boldsymbol{x}).$$

The approximation error of a state is then captured by a closed ball around center $\boldsymbol{x} \in \mathbb{R}^X$ and radius r, denoted by $B(\boldsymbol{x}, r)$. An upper bound on the evolution of the approximation error r over time is expressed by $\delta_{c,u}(r,t)$, which depends on three system-based constants including the one-sided Lipschitz (OSL) constant (see [3] for the full expression of $\delta_{c,u}(r,t)$ and the three constants). Then the following theorem captures the bounds on the evolution of the continuous dynamics.

Theorem 1: [3] Given a switched system that is locally and one-sided Lipschitz in S, point $\tilde{\boldsymbol{x}}_0 \in S$, and a positive real r. Then $\forall \boldsymbol{x}_0 \in B(\tilde{\boldsymbol{x}}_0, r)$, $t \in [0,\tau]$, $c \in C$, and $u \in U$: $F_{c,u}(t, \boldsymbol{x}_0) \in B(\tilde{\phi}(t,(c,u,\tilde{\boldsymbol{x}}_0)), \delta_{c,u}(r,t))$.
This theorem assumes that the step size h in the Euler's method is equal to the switching period τ. To obtain better approximations, i.e., a smaller radius of the ball B, one can consider a uniform subdivision of $[0,\tau]$ into k steps and apply the Euler's method with step size $h = \frac{\tau}{k}$.

III. Safe synthesis using analytical solutions

While the Euler method is applicable to almost any system using (1), some systems have closed-form solutions $\phi(t,\gamma)$ describing the continuous state evolution, resulting in $F_{c,u}(t,\boldsymbol{x}) = \phi(t,(c,u,\boldsymbol{x}))$ where $\gamma = (c,u,\boldsymbol{x})$. We can utilize the closed-form solution to obtain tighter integer bounds for the transformation of a switched HMDP to a timed game automaton, because using the closed-form solution eliminates the error from approximating the state trajectory with the Euler method. With our new method the only source of uncertainty is the approximation of the state by integers at the end of each switching interval.

At each step i of a run of a switched HMDP, we need to obtain an integer-valued lower bound $\boldsymbol{x}_i^{min}$ and upper bound $\boldsymbol{x}_i^{max}$ such that the true state $\boldsymbol{x}_i$ is within the bounds, i.e., $\forall x \in X : x_i^{min} \leq x_i \leq x_i^{max}$. For guaranteed safety synthesis, we also need to keep track of the minimum lower bound and maximum upper bound during a period, as the true state trajectory might have local minima or maxima, see the example in Fig. 1. We denote with $\boldsymbol{H}_i^x$ and $\boldsymbol{G}_i^x$ the integer-valued lower and upper bound, respectively, of state variable x in interval $[i\tau, (i+1)\tau]$, i.e., $\forall t \in [i\tau, (i+1)\tau], \forall x \in X : \boldsymbol{H}_i^x \leq x(t) \leq \boldsymbol{G}_i^x$.

The following analysis holds when we have the following two assumptions, which hold for many physical systems.

Assumption 1: $\phi(t,\gamma)$ is differentiable on $[0,\tau]$, i.e., $\forall t \in [0,\tau] : \frac{\mathrm{d}}{\mathrm{d}t}\phi(t,\gamma)$ exists.

Assumption 2: $\phi(t,\gamma)$ is monotone with respect to ordering $\leq$, i.e., $\forall \boldsymbol{x_1}, \boldsymbol{x_2} \in X \rightarrow \mathbb{R}$, if $\boldsymbol{x}_1 \leq \boldsymbol{x}_2$, then $\forall t > 0, c \in C, u \in U : \phi(t,(c,u,\boldsymbol{x}_1)) \leq \phi(t,(c,u,\boldsymbol{x}_2))$.

A. 1-dimensional system

We first focus the analysis on an 1-dimensional system, i.e., $X = \{x\}$. In each of the switching periods, the dynamics of state variable x can be either increasing, decreasing, or neither of those depending on the initial value of x in that period. Since we approximate x_i with x_i^{min} and x_i^{max}, we have to check the evolution of both bounds. It can be easily true that the lower bound is increasing while the upper bound is decreasing. Nonetheless, in all cases we have that $x_{i+1}^{min} = \lfloor \phi(\tau,(c_i,u_i,x_i^{min})) \rfloor$ and $x_{i+1}^{max} = \lceil \phi(\tau,(c_i,u_i,x_i^{max})) \rceil$.

If a system is *increasing* in period i, it holds that $\forall t \in [0,\tau] : \frac{\mathrm{d}}{\mathrm{d}t}\phi(t,\gamma_i) \geq 0$. If we are analyzing the evolution of the lower bound, i.e. $\gamma_i = (c_i,u_i,x_i^{min})$, we immediately know that the lowest value of $\phi(t,\gamma_i)$ in this interval is the initial state, i.e., $\boldsymbol{H}_i^x = x_i^{min}$. Similarly, if we are analyzing the evolution of the upper bound with $\gamma_i = (c_i,u_i,x_i^{max})$, the greatest value of $\phi(t,\gamma_i)$ is the final state, i.e., $\boldsymbol{G}_i^x = x_{i+1}^{max}$.

If a system is *decreasing* in period i, it holds that $\forall t \in [0,\tau] : \frac{\mathrm{d}}{\mathrm{d}t}\phi(t,\gamma_i) \leq 0$. Using the same analysis as for increasing functions, we can immediately note that when analyzing the lower bound, $\boldsymbol{H}_i^x = x_{i+1}^{min}$, and when analyzing the upper bound, $\boldsymbol{G}_i^x = x_i^{max}$.

If a system has one or more *local minima or maxima*, it holds that for some $t_1, t_2 \in [0,\tau] : \frac{\mathrm{d}}{\mathrm{d}t}\phi(t_1,\gamma_i) < 0 \wedge \frac{\mathrm{d}}{\mathrm{d}t}\phi(t_2,\gamma_i) > 0$. We can obtain the local minima and maxima by solving for $\frac{\mathrm{d}}{\mathrm{d}t}\phi(t,\gamma_i) = 0$. Let $X_{\gamma_i}^0 = \{\phi(t^0,\gamma_i) \mid t^0 \in [0,\tau], \frac{\mathrm{d}}{\mathrm{d}t}\phi(t^0,\gamma_i) = 0\}$ be the state values of these local minima and maxima. Now, if $\gamma_i = (c_i,u_i,x_i^{min})$, then $\boldsymbol{H}_i^x = \lfloor \min\{X_{\gamma_i}^0 \cup \{x_i^{min}, x_{i+1}^{min}\}\} \rfloor$; similarly, if $\gamma_i = (c_i,u_i,x_i^{max})$, then $\boldsymbol{G}_i^x = \lceil \max\{X_{\gamma_i}^0 \cup \{x_i^{max}, x_{i+1}^{max}\}\} \rceil$. Fig. 1 shows an example where the lower bound of variable x has a single local minimum at t^0. Here $\boldsymbol{H}_i^x = \lfloor \min\{\{\phi(t^0,\gamma_i)\} \cup \{x_i^{min}, x_{i+1}^{min}\}\} \rfloor = \lfloor \phi(t^0,\gamma_i) \rfloor$.

Now, Theorem 2 confirms that $\boldsymbol{x}^{min}$, $\boldsymbol{x}^{max}$, $\boldsymbol{H}$, and $\boldsymbol{G}$ as calculated above are indeed bounds for the continuous evolution of a 1-dimensional system.

Theorem 2: Given a one-dimensional switched system with the closed-form solution $\phi(t,\gamma)$ that is differentiable and monotone, and bounds $\boldsymbol{x}_i^{min}$ and $\boldsymbol{x}_i^{max}$. Then $\forall \boldsymbol{x}_i \in [\boldsymbol{x}_i^{min}, \boldsymbol{x}_i^{max}]$, $t \in [0,\tau]$, $c \in C$, and $u \in U$: $F_{c,u}(t,\boldsymbol{x}_i) \in [\boldsymbol{H}_i, \boldsymbol{G}_i]$ and $F_{c,u}(\tau,\boldsymbol{x}_i) \in [\boldsymbol{x}_{i+1}^{min}, \boldsymbol{x}_{i+1}^{max}]$.

B. n-dimensional system

When the system consists of multiple state variables, the analysis from the previous section becomes more involved. Nonetheless, the fundamental idea behind the analysis remains the same. Due to the monotonicity assumption of the system, it suffice the perform the 1-dimensional analysis on

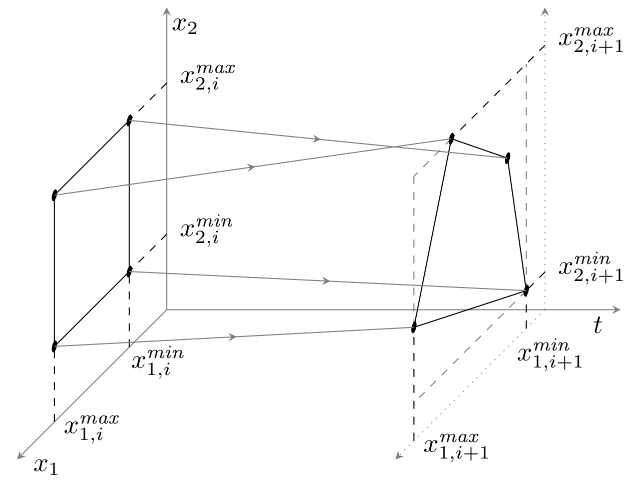

Fig. 2. Visualize the trajectories of a 2-dimensional plane in a period, i.e., the corners $\{(x_{1,i}^{min}, x_{2,i}^{min}), (x_{1,i}^{min}, x_{2,i}^{max}), (x_{1,i}^{max}, x_{2,i}^{min}), (x_{1,i}^{max}, x_{2,i}^{max})\}$.

each of the corners of the plane spanned by the state bounds, i.e., in period i we have to analyze the trajectories originating from $P = \{(x_{1,i}, x_{2,i}, \ldots, x_{n,i}) \mid \forall x_j \in X : x_{j,i} \in \{x_{j,i}^{min}, x_{j,i}^{max}\}\}$. For example, consider a 2-dimensional system. Then the four initial states to consider are $\{(x_{1,i}^{min}, x_{2,i}^{min}), (x_{1,i}^{min}, x_{2,i}^{max}), (x_{1,i}^{max}, x_{2,i}^{min}), (x_{1,i}^{max}, x_{2,i}^{max})\}$.

Now, for each corner in the n-dimensional plane $\boldsymbol{x}_i^c \in P$, we perform the 1-dimensional analysis for each state variable $x \in X$ separately using $\boldsymbol{x}_i^c$ as the initial state. This results in lower and upper bounds specifically for that initial state, i.e., $x_{i+1,\boldsymbol{x}_i^c}^{min}$ or $x_{i+1,\boldsymbol{x}_i^c}^{max}$, and $\boldsymbol{H}_{i,\boldsymbol{x}_i^c}^x$ or $\boldsymbol{G}_{i,\boldsymbol{x}_i^c}^x$ (depending on whether $\boldsymbol{x}_i^c$ contains the lower or upper bound of x). Now, for each state variable x the overall lower and upper bound at the end of the period are given by $x_{i+1}^{min} = \min\{x_{i+1,\boldsymbol{x}_i^c}^{min} \mid \boldsymbol{x}_i^c \in P\}$ and $x_{i+1}^{max} = \max\{x_{i+1,\boldsymbol{x}_i^c}^{max} \mid \boldsymbol{x}_i^c \in P\}$, respectively, and the overall lower and upper bound during the period are given by $H_i^x = \min\{H_{i,\boldsymbol{x}_i^c}^x \mid \boldsymbol{x}_i^c \in P\}$ and $G_i^x = \min\{G_{i,\boldsymbol{x}_i^c}^x \mid \boldsymbol{x}_i^c \in P\}$, respectively.

Fig. 2 shows an example of a 2-dimensional system and the evolution of the corners in the plane spanned by $x_{1,i}^{min}$, $x_{1,i}^{max}$, $x_{2,i}^{min}$, and $x_{2,i}^{max}$. In this particular period, all trajectories evolve linear over time, but the slopes in the x_1 and x_2 domains are dependent on the initial location of the trajectory. Looking at the corner $(x_{1,i}^{min}, x_{2,i}^{min})$, we can see that the trajectory ϕ is decreasing in both state variables. But from the corner $(x_{1,i}^{max}, x_{2,i}^{max})$, ϕ is decreasing in x_1 and increasing in x_2.

Theorem 3: Given an n-dimensional switched system with the closed-form solution $\phi(t, \gamma)$ that is differentiable and monotone, and bounds $\boldsymbol{x}_i^{min}$ and $\boldsymbol{x}_i^{max}$. Then $\forall \boldsymbol{x}_i \in [\boldsymbol{x}_i^{min}, \boldsymbol{x}_i^{max}]$, $t \in [0, \tau]$, $c \in C$, and $u \in U$: $F_{c,u}(t, \boldsymbol{x}_i) \in [\boldsymbol{H}_i, \boldsymbol{G}_i]$ and $F_{c,u}(\tau, \boldsymbol{x}_i) \in [\boldsymbol{x}_{i+1}^{min}, \boldsymbol{x}_{i+1}^{max}]$.

IV. Cruise Control Example

In [3], the Euler method is illustrated with a cruise control example from [19]. In this case study there are two cars, Ego and Front, driving on a road, see Fig.3. We are able to control the acceleration of Ego, but not of Front. Each car can accelerate with -2 m/s^2, 0 m/s^2, and 2 m/s^2, between which they can switch instantaneously. Furthermore, both cars are capable of driving maximally 20 m/s forward and 10 m/s backwards. Ego is able to detect the distance to Front within a range of 200 m. If the distance is larger, Front is considered far away and out of range. The aim is to synthesize a controller such that the distance between Ego and Front is never less than 5 m.

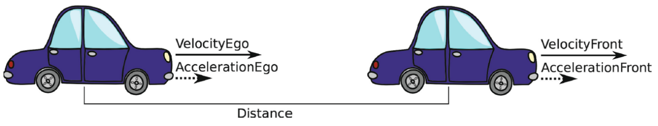

Fig. 3. Overview of the cruise control example. From [19]

This system can be described using the following system of differential equations:

$$\frac{\mathrm{d}}{\mathrm{d}t} v_f = a_f \qquad \frac{\mathrm{d}}{\mathrm{d}t} v_e = a_e \qquad \frac{\mathrm{d}}{\mathrm{d}t} d = v_f - v_e$$

where v_f and v_e are the speed of Front and Ego, respectively, a_f and a_e the acceleration of Front and Ego, respectively, which can take the values -2, 0, and 2, and d is the distance between Front and Ego. The set of safe states S is considered to be $S = [-10, 20] \times [-10, 20] \times [5, 200]$. The controller has three possible modes relating to the acceleration of Ego, i.e., $C = \{a_e = -2, a_e = 0, a_e = 2\}$, while the environment has three possible modes relating to the acceleration of Front, i.e., $U = \{a_f = -2, a_f = 0, a_f = 2\}$. Finally, the switching period is $\tau = 1$.

We can solve the system of differential equations analytically to obtain a closed-form of $\phi(t, (c_i, u_i, \boldsymbol{x}_i))$ where $\boldsymbol{x} = (v_{f,i}, f_{e,i}, d_i)$:

$$\begin{aligned} \phi_{v_f}(t, (c_i, u_i, \boldsymbol{x}_i)) &= v_{f,i} + t a_f \\ \phi_{v_e}(t, (c, u, \boldsymbol{x}_i)) &= v_{e,i} + t a_e \\ \phi_d(t, (c_i, u_i, \boldsymbol{x}_i)) &= d_i + \frac{a_f - a_e}{2} t^2 + (v_{f,i} + v_{e,i}) t \end{aligned}$$

Observe that with the chosen values for a_f, a_e, and τ, $v_{f,i+1}$ and $v_{e,i+1}$ are integers if $v_{f,i}$ and $v_{e,i}$ are integers as well. Therefore, if the initial state is known precisely, i.e., $\boldsymbol{x}_0 = \boldsymbol{x}_0^{min} = \boldsymbol{x}_0^{max}$, we have that $v_{f,i}^{min} = v_{f,i}^{max}$ and $v_{e,i}^{min} = v_{e,i}^{max}$ for all periods i.

For all state variables we can also derive the derivative:

$$\frac{\mathrm{d}}{\mathrm{d}t} \phi_{v_f}(t, (c, u, \boldsymbol{x})) = a_f \qquad \frac{\mathrm{d}}{\mathrm{d}t} \phi_{v_e}(t, (c, u, \boldsymbol{x})) = a_e$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \phi_d(t, (c, u, \boldsymbol{x})) = (a_f - a_e) t + (v_{f,0} + v_{e,0})$$

We can now immediately observe that both v_f and v_e are either increasing or decreasing in all periods, since the derivatives are constant over a period. But d might have local minima or maxima within a period. If we look at the second derivative of ϕ_d, i.e., $\frac{\mathrm{d}^2}{\mathrm{d}t^2} \phi_d(t, (c, u, \boldsymbol{x})) = (a_f - a_e)$, we see that this is constant within a period. Therefore, d can have

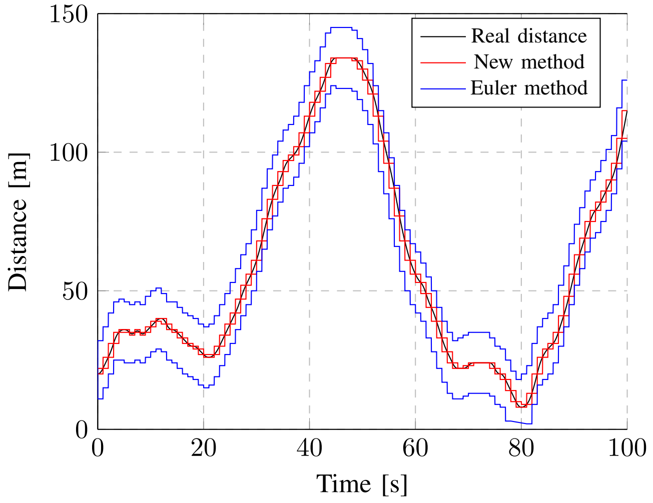

Fig. 4. Simulation of the real distance and the guaranteed lower and upper bounds provided by the Euler method and the newly presented method.

at most one local minimum or maximum within a period. If there is such a local minimum or maximum, it will be at $t_m = \frac{v_{e,0}-v_{f,0}}{a_f-a_e}$, so $\phi_d(t_m,(c,u,\boldsymbol{x})) = d_0 - \frac{(v_{e,0}-v_{f,0})^2}{2(a_f-a_e)}$.

We have implemented our new integer approximation into the UPPAAL model of the cruise control example from [3] so we can compare our approximations with the ones obtained using the Euler method. The model can be found at [24]. Fig. 4 shows the real distance and the calculated guaranteed lower and upper bounds using both methods. For the Euler method, we set the step size $h = \tau/5$. The results show that the new method outperforms the Euler method in the sense that the bounds are as tight as possible in this special case: in any interval the guaranteed lower and upper bound touches the true distance within that interval. This is due to the fact that the parameters chosen by the original creators of the case study ensure that the true values of the state variables at any switching point are precisely integer values. Finally, as mentioned in [3], the bounds of the Euler method can be made tighter by lowering the step size to $h = \tau/k$. After some experimentation it was found that the Euler bounds approximate the new bounds using a value of $k = 60$.

The resources required to synthesize a safe strategy are similar for both approximation methods. The same number of state variables are calculated at each switching period, resulting in a similar size of the explored state space. While the Euler method might require more calculation time when $k > 1$, memory is the limiting factor for this use case. In both situations, synthesis had to be performed on a dedicated computation server having 100 GB of memory available. After synthesizing the strategies, UPPAAL model checking is used explore the distance between the cars. Fig. 5 shows the infimum distance between the cars for different relative velocities using the strategies from both methods. Note that UPPAAL can only calculate the infimum using symbolic methods, hence the variables used are also the approximated ones instead of the continuous ones. The results show that the two strategies have similar performance.

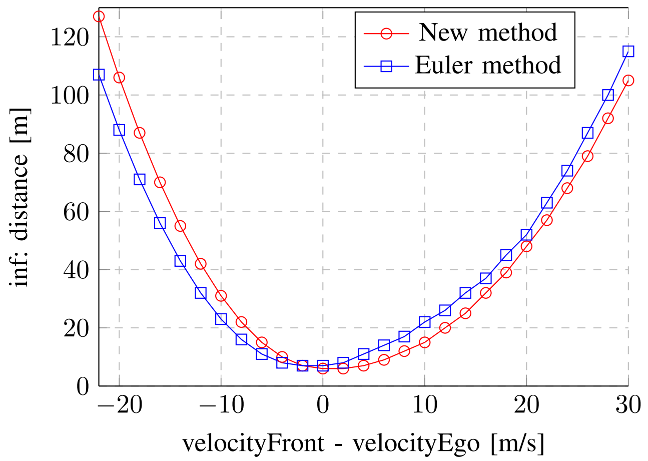

Fig. 5. The smallest distance between the two cars under the safe strategy synthesized by the Euler method and our new method as a function of the relative velocity. The query 'inf$\{v_{f,i}^{min} - v_{e,i}^{min} = n\} : H^d$ under safe' has been executed for different values of n.

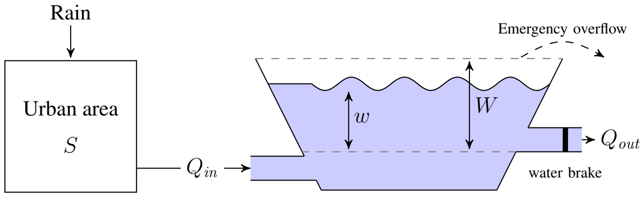

Fig. 6. Overview of the storm water detention pond. From [8]

V. Stormwater Detention Pond Example

In [8] a HMDP of a stormwater detention pond is provided. If we set the switching period to $\tau = 1$ minute, the model becomes a switched HMDP. Applying the Euler method to this case study resulted in exploding bounds. While the main theorem of [3] still holds, i.e., the true value is always between the bounds, exploding bounds are useless in a practical sense for synthesizing a guaranteed safe controller.

Fig. 6 shows a schematic overview of a stormwater detention pond. The stormwater detention pond contains two continuous state variables: the height of the stormwater S on the surface of the urban catchment area and the height of the water w in the stormwater detention pond. The environment rainfall intensity $rain$ and the controlled outflow Q_{out} are determined by the environment mode u and the control mode c, respectively, and constant within a switching period. I.e., each uncontrollable environment mode u is associated with a particular rainfall intensity and the switching density function family ρ is based on the weather forecast. The dynamics of the system are given by [8]

$$\frac{\mathrm{d}}{\mathrm{d}t}S = f_S rain - kS$$
$$\frac{\mathrm{d}}{\mathrm{d}t}w = \frac{f_S}{f_w}k\frac{A_{uc}}{A_p}S - f_w\frac{Q_{out}}{A_p}$$

where f_S and f_w are scaling factors for S and w, respectively, k is the urban surface reaction factor, A_{uc} the surface area of the urban catchment area, and A_p the surface area of the stormwater pond.

From this system of differential equations we obtain the exact solutions where $\gamma_i = (c_i, u_i, \boldsymbol{x}_i)$ and $t \in [0, \tau]$

$$\begin{aligned}
\phi_S(t, \gamma_i) &= \frac{f_S rain}{k} + \left(S_i - \frac{f_s rain}{k}\right) e^{-kt} \\
\phi_w(t, \gamma_i) &= w_i + f_w \frac{rain A_{uc} - Q_{out}}{A_p} t \\
&\quad + \frac{f_w}{f_S} \frac{A_{uc}}{A_p} \left(S_i - \frac{f_s rain}{k}\right) (1 - e^{-kt})
\end{aligned}$$

Thus the derivatives are given by

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} \phi_S(t, \gamma_i) &= -k \left(S_k - \frac{f_S rain}{k}\right) e^{-kt} \\
\frac{\mathrm{d}}{\mathrm{d}t} \phi_w(t, \gamma_i) &= f_w \frac{rain A_{uc} - Q_{out}}{A_p} \\
&\quad + k \frac{f_w}{f_S} \frac{A_{uc}}{A_p} \left(S_i - \frac{f_S rain}{k}\right) e^{-kt}
\end{aligned}$$

In these equations, f_S, f_w, k, A_{uc}, and A_p are time-global constants > 0, and $rain$ and Q_{out} are period-local constants ≥ 0. Therefore, $\frac{\mathrm{d}}{\mathrm{d}t} \phi_S(t, \gamma_i)$ is either increasing or decreasing in any period i. For the water level w in the pond we have

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} \phi_w(t, \gamma_i) = -k^2 \frac{f_w}{f_S} \frac{A_{uc}}{A_p} \left(S_i - \frac{f_S rain}{k}\right) e^{-kt}$$

So, $\forall t \in [0, \tau] : \frac{\mathrm{d}^2}{\mathrm{d}t^2} \phi^w(t, \gamma_i) \leq 0$ or $\forall t \in [0, \tau] : \frac{\mathrm{d}^2}{\mathrm{d}t^2} \phi_w(t, \gamma_i) \geq 0$. Therefore, we can conclude that $\frac{\mathrm{d}}{\mathrm{d}t} \phi_S(t, \gamma_i)$ is either increasing, decreasing, or has a single local minimum or maximum within a period i. If there is such a local minimum or maximum, it will be at

$$t_m = -\frac{1}{k} \ln \left(f_S \frac{rain A_{uc} - Q_{out}}{A_{uc} \left(f_S rain - k S_i\right)} \right)$$

resulting in a water level of

$$\begin{aligned}
\phi_w(t_m, \gamma_i) &= w_i + f_w \frac{rain A_{uc} - Q_{out}}{A_p} \left(t_m + \frac{1}{k}\right) \\
&\quad + \frac{f_w}{f_S} \frac{A_{uc}}{A_p} \left(S_i - \frac{f_s rain}{k}\right).
\end{aligned}$$

We have incorporated this new method into the UPPAAL model of the stormwater detention pond from [8]. The model can be found at [24]. Compared to the cruise control example, we scaled the dimensions of S and w to allow for reasonable integer approximations of the continuous dynamics. Technically speaking rationals are also possible, but they can be transformed into integers by using a proper scaling. In [8], the least intense rain event has an intensity of 0.00952 cm/min. So rounding this to the nearest integers introduces a significant rounding error. A first reasonable scaling of S is $f_S = 1000$, so the least rain intensity becomes $9.52/f_S$ cm/min. But this scaling is insufficient. After the rain, the surface stormwater level will asymptotically approach 0, but the integer approximation will suffer again from rounding errors once the change is less than one. With $f_S = 1000$, the upper bound will stuck at $S = 5$ in this situation. After some experimentation we found that reducing this rounding error by fivefold results in an acceptable error. Thus we obtained $f_S = 5000$. Using a similar analysis, we obtained $f_w = 200$.

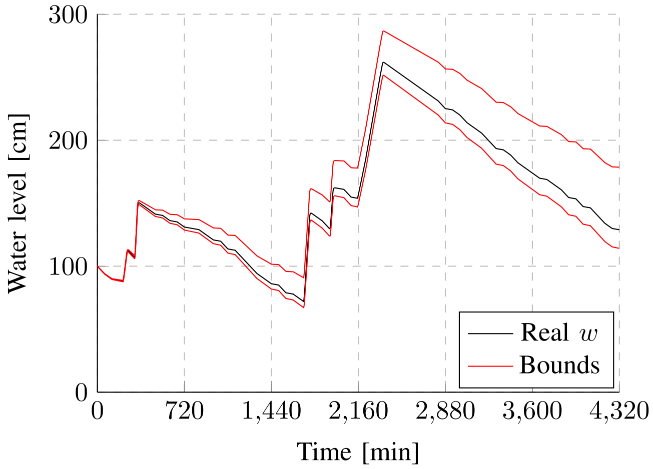

Fig. 7. Simulation of the stormwater pond model including integer bounds. Solid lines indicate the water level in the pond; the black and dotted line indicates the rain fall. As a quick reference, 1440 minutes is 1 day, and the total time scale is 3 days.

Fig. 7 shows simulation results using the new method to obtain integer bounds for both w (the bounds for S too tight to be visible in the figure). As can be seen, we no longer have integer bounds that explode, i.e., the bound captures the continues dynamics reasonably well. Furthermore, the bounds of w are not symmetric around the true value of w. This can mostly be explained by the fact that the upper bound of S does not asymptotically reach 0 but the value 4 (or the lower bound reaching the equilibrium when raining). So, in dry periods when it is not raining, the non-zero upper bound of S still causes inflow into the pond for the upper bound analysis.

We are now able to synthesize a guaranteed safe controller for the stormwater detention pond. While an infinite horizon controller has been synthesized for the cruise control example, we synthesize finite horizon controllers for stormwater detention ponds, as weather forecasts are only available for a limited horizon into the future and the accuracy of the predicted rain decreases quickly the longer in the future the prediction is for. We were able to synthesize a safe controller for a time horizon of 5 hours using a Macbook in 2 minutes and using 4.6 GB RAM. Increasing the time horizon to 6 hours fails due to running our-of-memory, also on the computational cluster having 200 GB RAM available. A control horizon of 5 hours might be sufficient when controller synthesis is used in a model-predictive control setting [25].

VI. Conclusion

In this paper an new method is presented to obtain integer-valued bounds for continuous-time sample switched systems where analytical solutions for the trajectories are available.

With these bounds, Uppaal Tiga is able to synthesize maximally permissive guaranteed safe controllers for the original continuous switched system. After obtaining a maximally permissive controller, further analysis can now be done to obtain an optimal and safe controller, for example using reinforcement learning as implemented in Uppaal Stratego [22].

The new method is applied on a cruise control case study and an stormwater detention pond case study. For the cruise control example, the new method results in tighter bounds compared to the Euler method from [3]. Therefore, the synthesized safe controller can be more optimistic, as the approximation error is less. For the stormwater detention pond example, we are now able to obtain reasonable bounds, so that we can now actually synthesize a guaranteed safe controller for a fixed control horizon.

Future work could investigate whether we can obtain tighter bounds in the n-dimensional case, as we now only keep track of the minimum and maximum value of each state variable regardless of the values in combination with the other state variables. Furthermore, it would help if the integer bounds also approximate the same limit as the underlying continuous dynamics. But this might require additional variables that act as temporarily memory. Another direction could be to relax the assumption that the environment also switches periodically, so we can, e.g., synthesize a safe strategy for the bouncing ball example of [26]. Finally, memory usage of Uppaal Tiga seems to be the limiting factor in applying the method to large-scale systems. Our proposed method turned the original continuous-time model of the cruise control example into effectively an untimed model where the progress of a period can be represented by a single event. It might be interesting to see whether untimed synthesis methods like [27], [28] can help.

Acknowledgment

The authors thank Adrien Le Coënt for discussions on the implementation of the Euler method for the cruise control example.

References

[1] H. Lin and P. J. Antsaklis, "Stability and stabilizability of switched linear systems: a survey of recent results," *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 308–322, 2009.

[2] A. Saoud, A. Girard, and L. Fribourg, "Contract based design of symbolic controllers for vehicle platooning," in *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control*, 2018, pp. 277–278.

[3] K. G. Larsen, A. Le Coënt, M. Mikučionis, and J. H. Taankvist, "Guaranteed control synthesis for continuous systems in Uppaal Tiga," in *Cyber Physical Systems. Model-Based Design*, ser. LNCS, no. 11615. Springer International Publishing, 2019, pp. 113–133.

[4] K. G. Larsen, M. Mikučioni, M. Muñiz, J. Srba, and J. H. Taankvist, "Online and compositional learning of controllers with application to floor heating," in *TACAS*, ser. LNCS, 2016, pp. 244–259.

[5] P.-J. Meyer, A. Girard, and E. Witrant, "Robust controlled invariance for monotone systems: application to ventilation regulation in buildings," *Automatica*, vol. 70, pp. 14–20, 2016.

[6] L. Fribourg, U. Kühne, and N. Markey, "Game-based synthesis of distributed controllers for sampled switched systems," in *SynCoP*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

[7] L. Fribourg and R. Soulat, *Control of switching systems by invariance analysis: applcation to power electronics*. John Wiley & Sons, 2013.

[8] M. A. Goorden, K. G. Larsen, J. E. Nielsen, T. D. Nielsen, M. R. Rasmussen, and J. Srba, "Learning safe and optimal control strategies for storm water detention ponds," *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 13–18, 2021.

[9] M. Rungger and M. Zamani, "SCOTS: A tool for the synthesis of symbolic controllers," in *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*. New York, NY, USA: Association for Computing Machinery, 2016, p. 99–104. [Online]. Available: https://doi.org/10.1145/2883817.2883834

[10] A. Girard, "Controller synthesis for safety and reachability via approximate bisimulation," *Automatica*, vol. 48, no. 5, pp. 947–953, 2012.

[11] A. Le Coënt, J. Alexandre dit Sandretto, A. Chapoutot, and L. Fribourg, "An improved algorithm for the control synthesis of nonlinear sampled switched systems," *Formal Methods in System Design*, vol. 53, no. 3, pp. 363–383, 2018.

[12] L. Doyen, G. Frehse, G. J. Pappas, and A. Platzer, "Verification of hybrid systems," in *Handbook of Model Checking*. Springer, 2018, pp. 1047–1110.

[13] P. Nilsson, O. Hussien, A. Balkan, Y. Chen, A. D. Ames, J. W. Grizzle, N. Ozay, H. Peng, and P. Tabuada, "Correct-by-construction adaptive cruise control: Two approaches," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 4, pp. 1294–1307, 2015.

[14] A. Girard and S. Martin, "Synthesis for constrained nonlinear systems using hybridization and robust controllers on simplices," *IEEE Trans. Autom. Control*, vol. 57, no. 4, pp. 1046–1051, 2011.

[15] A. Le Coënt, L. Fribourg, N. Markey, F. De Vuyst, and L. Chamoin, "Compositional synthesis of state-dependent switching control," *Theoretical Computer Science*, vol. 750, pp. 53–68, 2018.

[16] J. Jerray, L. Fribourg, and É. André, "Robust optimal periodic control using guaranteed Euler's method," in *ACC*. IEEE, 2021, pp. 986–991.

[17] G. Behrmann, A. Cougnard, A. David, E. Fleury, K. G. Larsen, and D. Lime, "Uppaal-tiga: Time for playing games!" in *CAV*, ser. LNCS, 2007, pp. 121–125.

[18] A. Le Coënt, F. De Vuyst, L. Chamoin, and L. Fribourg, "Control synthesis of nonlinear sampled switched systems using Euler's method," in *International Workshop on Symbolic and Numerical Methods for Reachability Analysis*, ser. Electronic Proceedings in Theoretical Computer Science, 2017.

[19] K. G. Larsen, M. Mikučioni, and J. H. Taankvist, "Safe and optimal adaptive cruise control," in *Olderog-Festschrift*, ser. LNCS, 2015, pp. 260–277.

[20] P. Ashok, J. Křetínský, K. G. Larsen, A. Le Coënt, J. H. Taankvist, and M. Weininger, "SOS: Safe, optimal and small strategies for hybrid Markov decision processes," in *QEST*, ser. LNCS, 2019, pp. 147–164.

[21] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994.

[22] A. David, P. G. Jensen, K. G. Larsen, M. Mikučionis, and J. H. Taankvist, "Uppaal Stratego," in *TACAS*, ser. LNCS, 2015, pp. 206–211.

[23] O. Maler, A. Pnueli, and J. Sifakis, "On the synthesis of discrete controllers for timed systems," in *STACS*, ser. LNCS. Springer, 1995, pp. 229–242.

[24] M. A. Goorden, K. G. Larsen, J. E. Nielsen, T. D. Nielsen, W. Qian, M. R. Rasmussen, and G. Zhao, "Models and data for "Guaranteed safe controller synthesis for switched systems using analytical solutions"," Zenodo, 2023.

[25] M. A. Goorden, P. G. Jensen, K. G. Larsen, M. Samusev, J. Srba, and G. Zhao, "STOMPC: Stochastic model-predictive control with Uppaal Stratego," in *ATVA*. Cham: Springer, 2022, pp. 327–333.

[26] M. Jaeger, P. G. Jensen, K. G. Larsen, A. Legay, S. Sedwards, and J. H. Taankvist, "Teaching stratego to play ball: Optimal synthesis for continuous space MDPs," in *ATVA*, ser. LNCS, 2019, pp. 81–97.

[27] S. Miremadi, B. Lennartson, and K. Akesson, "A BDD-based approach for modeling plant and supervisor by extended finite automata," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 6, pp. 1421–1435, 2012.

[28] M. Goorden, J. van de Mortel-Fronczak, M. Reniers, W. Fokkink, and J. Rooda, "Structuring multilevel discrete-event systems with dependence structure matrices," *IEEE Trans. Autom. Control*, vol. 65, no. 4, pp. 1625–1639, 2019.