Hybrid Automata and STL properties for Experiments

1 Benchmark Models for bounded model checking of STL

We perform bounded model checking of STL to compare the efficiency of our algorithm and the previous algorithm. We consider the following hybrid automata models adapted from [1,2,3,4]: autonomous driving of cars, the networked of thermostat and watertank controllers, a controller for a railroad gate, turning an airplain, and a load management controller for two batteries, with variants dynamics. For each model, we consider four STL formulas of different sizes and complexity.

1.1 Networked thermostat controllers

There are two rooms and they are interconnected by an open door. The temperature x_i of each $room_i$ is separately controlled by each thermostat, depending on both the heater's mode $m_i \in \{on, off\}$ and the temperature of the adjacent room. Initially, the temperature x_1 of the $room_1$ is $19.9 \leq x_1 \leq 20.1$, and the temperature x_2 of the $room_2$ is $23 \leq x_2 \leq 23.5$. For both rooms, the heaters turn off initially.

The behavior of each heater is as follows. If the temperature is higher than the mean value of both temperatures, the heater will be turned off. Otherwise, the heater will be turned on. The temperatures change according to the following ODEs.

1. Linear dynamics (the solutions of the ODEs are linear functions)

$$\dot{x_1} = \begin{cases} -0.4 & if\ m_1 = off \\ 0.7 & if\ m_1 = on \end{cases} \qquad \dot{x_2} = \begin{cases} -0.6 & if\ m_2 = off \\ 1 & if\ m_2 = on \end{cases}$$

2. Polynomial dynamics (the solutions of the ODEs are polynomials)

$$\dot{x_1} = \begin{cases} -K_1\big((1-2c)cx_1 + c * cx_2\big) & if\ m_1 = off \\ K_1\Big(h_1 - \big((1-2c)cx_1 + c * cx_2\big)\Big) & if\ m_1 = on \end{cases} \qquad c\dot{x}_1 = c\dot{x}_2 = 0$$

where $c, K_i, h_i \in \mathbb{R}$ are constants depending on the size of the door, the size and the heater's power. The variable cx_i is a constant (with derivative 0) that captures the value of x_1 at a certain moment, so that the dynamics approximately becomes a polynomial. The dynamics of x_2 is similar to one of x_1. We simplify the controllers by considering only one thermostat controller for nonlinear-ode dynamics.

3. Nonlinear-ode dynamics

$$\dot{x_1} = \begin{cases} -K_1\big((1-2c)cx_1\big) & if\ m_1 = off \\ K_1(h_1 - (1-2c)cx_1) & if\ m_1 = on \end{cases}$$

The following STL formulas are considered for linear and polynomial dynamics. We consider two simple formula with only one temporal operator (φ_1, φ_2) and a nontrivial formula with a nested temporal operator (φ_3, φ_4). We also consider existence of counterexamples. There exist a counterexample for φ_1 and φ_3 and no counterexamples for φ_2 and φ_4.

STL formula	ID	Explanation
$\Diamond_{[0,40]}(x_2 > 24)$	φ_1	Within 40 time units, x_2 becomes greater than 24.
$\Box_{[10,30]}(x_1 < 22)$	φ_2	For 20 time units, x_1 is always less than 22.
$\Box_{[0,10]}(off_1 U_{[0,15)}(x_1 < 20))$	φ_3	Within 10 time units, the heater of the $room_1$ turns on, until, x_1 is less than 20
$\Diamond_{(5,30)}((x_2 > 19) \mathrm{R}_{[0,30]} on_2)$	φ_4	Within 25 time units, the heater of the $room_2$ turns on, thereafter x_2 is greater than 19 within 30 time units.

Similarly, we designed STL formulas for nonlinear-ode. We consider different STL formulas for nonlinear-ode, since there is an only one thermostat controller for nonlinear-ode dynamics model.

STL formula	ID	Explanation
$\Diamond_{[0,40]}(x_1 \geq 26)$	φ_1	Within 40 time units, x_1 becomes greater than or equal to 26.
$(x_1 < 23)\ R_{[4,8]} on_1$	φ_2	The heater turns off sometime within 4 time units, thereafter x_1 is less than 23 for the interval [4, 8].
$\Diamond_{[0,30]}(off_1 U_{[5,15]}(x_1 < 18))$	φ_3	Within 30 time units, m_1 is turned off, until, x_1 is less than 18
$\Box_{[0,20]}((x_1 > 27) \rightarrow \Diamond_{[0,10)} off_1)$	φ_4	For 20 time units, if x_1 is greater than 27, m_1 is turned off within 10 time units.

1.2 Networked water tank controllers

There are two water tanks and they are connected by a pipe. The water level x_i of each tank is controlled by each pump, depending on the pump's mode $m_i \in \{on, off\}$ and the water level of the adjacent water tank. Initially, the water level of each water tank is higher than 4.9 and less than 5.1, and the both pumps are on.

The behavior of each pump is as follows. If the water level x_1 is less than 1, the $pump_1$ is turned on. If the water level x_1 is greater than the differences of both water levels, the $pump_1$ is turned off. For $pump_2$, it is on only if x_2 is less than 1. The water level of each tank changes according to the following dynamics.

1. Linear dynamics

$$\dot{x_1} = \begin{cases} -0.2 & if\ m_1 = off \\ 0.5 & if\ m_1 = on \end{cases} \qquad \dot{x_2} = \begin{cases} -0.3 & if\ m_2 = off \\ 0.6 & if\ m_2 = on \end{cases}$$

2. Polynomial dynamics (using Tylor approximation of square root at t = 1)

$$\dot{x_1} = \begin{cases} -(ag * cx_1)/2A_1 & if\ \ m_1 = off \\ (q_1 - ag * cx_1)/2A_1 & if\ \ m_1 = on \end{cases}$$

$$\dot{x_2} = \begin{cases} (ag * (cx_1 - cx_2))/2A_2 & if\ \ m_1 = off \\ (q_2 + ag * (cx_1 - cx_2))/2A_2 & if\ \ m_1 = on \end{cases}$$

where $A_i, q_i, a \in \mathbb{R}$ are constants determined by the size of the tank, the power of the pump, and the width of the pipe, and g is the standard gravity constant. . The variable cx_i is a constant (with derivative 0) that captures the value of x_1 at a certain moment, so that the dynamics is a polynomial. We simplify the controllers by considering only one watertank controller for nonlinear-ode dynamics.

3. Nonlinear-ode dynamics

$$\dot{x_1} = \begin{cases} -a\sqrt{2g}\sqrt{x} & if\ m_1 = off \\ q - a\sqrt{2g}\sqrt{x} & if\ m_1 = on \end{cases}$$

We designed STL formulas for the watertank model in a similar way to the thermostat model. The following table shows STL formulas for linear and polynomial dynamics models.

STL formula	**ID**	**Explanation**
$\mathrm{off}_1 U_{[0,30)}\ (x_1 < 4)$	φ1	The pump for the second watertank turns off, until, the water level of the $tank_2$ is less than 4.
$\Box_{[0,50]}(\mathrm{x}_1 > 2)$	φ2	For 50 time units, x_1 is greater than 2.
$\Box_{[5,30]}(\Diamond_{(0,15]}(x_2 > 6))$	φ3	For 10 time units, if x_1 is less than 5, for 5 time units, the pump$_1$ can be on within 5 time units.
$\Diamond_{[0,40]}(x_1 > 3\ R_{[0,40)} on_1)$	φ4	For 20 time units, whenever both pumps are on, one of them is off within 5 time units.

Similarly, we consider different STL formulas for the nonlinear-ode dynamics model. The STL formulas for nonlinear-ode dynamics model is summarized as follows.

STL formula	**ID**	**Explanation**
$\Diamond_{[0,50)}\ (x_1 \geq 5.6)$	φ1	The water level of the $tank_1$ is greater than or equal to 5.6 within 50 time units.
$(x_1 > 3)\ R_{(2,12]} off_1$	φ2	Within 10 time units, the pump of the tank turns off, thereafter x_1 is greater than 5.
$\Diamond_{[4,12]}\left(\mathrm{off}_1\ R_{[1,12]}(x_1 > 4)\right)$	φ3	Within 8 time units, x_1 is greater than 4 for the interval [1,12], thereafter the pump turns off.
$\Box_{[0,30]}(x_1 < 3 \rightarrow \Diamond_{[0,10]} on_1)$	φ4	For 30 time units, if x_1 is less than 3, the pump turns on within 10 time units.

1.3 Driving Simple Cars

1.3.1 Linear dynamics

Two cars car_1 and car_2 are running in a straight road, while car_1 follows car_2. The velocity of each car depends on the distance between two cars. Each car can move at different velocities, depending on its mode $m_i \in \{fast, mid, slow\}$. Initially, both cars are in the $fast$ mode, where the position x_1 of car_1 is in [0, 1], and the position x_2 of car_2 is in [5, 10]. The dynamics of two cars are as follows.

$$\dot{x_1} = \begin{cases} 65 & if\ m_1 = fast \\ 30 & if\ m_1 = mid \\ 25 & if\ m_1 = slow \end{cases} \qquad \dot{x_2} = \begin{cases} 60 & if\ m_2 = fast \\ 40 & if\ m_2 = mid \\ 35 & if\ m_2 = slow \end{cases}$$

The mode of each car is determined by the distance $x_2 - x_1$. For example, if the distance is less than 1, car_1 moves slow and car_2 moves fast. If the distance is greater than or equal to 3 and less than 4, car_1 moves fast and car_2 moves at normal speed. If the distance is greater than or equal to 5, car_1 accelerates and car_2 decelerates. We designed STL formulas for the car model in a similar way to the previous models.

STL formula	ID	Explanation
$\Diamond_{[10,30]}(x_2 - x_1) > 20$	φ_1	Within 20 time units, the distance between car_1 and car_2 is greater than 20.
$\Box_{[0,100]}\ x_2 \geq x_1$	φ_2	For 100 time units, car_2 is always ahead of car_1.
$\Diamond_{[0,20]}(\Box_{[0,5]}((x_2 - x_1) > 10))$	φ_3	Within 20 time units, the distance is greater than 10 for the interval [0,5].
$\Box_{[0,60]}((x_2 - x_1) < 2) \rightarrow \Diamond_{[0,10)}(v_1 \leq 30))$	φ_4	For 60 time units, if the distance is less than 2, the velocity car_1 is less than or equal to 30 within 10 time units.

1.3.2 Polynomial dynamics

Two cars are running in sequence, while each car follows the behavior of the car in front (the first car moves according to its own scenario). Each car can rotate and it can move at different velocities, depending on its mode $m_i \in \{stay, acc, dec\}$. Initially, the value of each variables is as follows.

Variables	Initialized value
the position x_1 of car_1	$0 < x_1 < 3$

the position y_1 of car_1	$3 < y_1 < 10$
the position x_2 of car_2	$5 < x_2 < 10$
the position y_2 of car_2	$3 < y_2 < 10$
the velocity v_1 of the car_1	$1 \leq v_1 \leq 3$
the velocity v_2 of the car_2	$3 \leq v_2 \leq 4$
the direction θ_1 of car_1	$0 < \theta_1 < 1$
the direction θ_2 of car_2	$-1 < \theta_2 < 0$
the steering angle ϕ_1 of car_1	$0 < \phi_1 < 1$
the steering angle ϕ_2 of car_2	$-1 < \phi_2 < 0$

The behavior of each car is as follows. If the $distance(car_1, car_2)$ is $distance(car_1, car_2) < 6$, then the car_2 changes its velocity to -5. If the distance is $6 \leq distance(car_1, car_2) \leq 9$, then the car_2 changes its velocity to $-(v_2 - v_1)$. If the distance is $distance(car_1, car_2) > 9$, then the car_2 changes its velocity to 5. The dynamics of two cars are as follows.

We used Tylor approximation of linearization of trigonometric functions at t = 2.)

$$\dot{x}_i = v_i \cos\theta_i, \ \ \dot{y}_i = v_i \sin\theta_i, \ \ \dot{\theta}_i = v_i \tan\phi_i$$

$$\dot{\phi}_2 = \phi_2 - \phi_1$$

$$\dot{v}_2 = \begin{cases} -(cv_2 - cv_1) & if\ m_2 = stay \\ 5 & if\ m_2 = acc \\ -5 & if\ m_2 = dec \end{cases}$$

The variable cx_i is a constant (with derivative 0) that captures the value of x_i at a certain moment, so that the dynamics is a polynomial. The values of ϕ_1 and v_1 change depending on the scheduled scenario of the car_1. We use the same STL formulas as the linear car model.

1.3.3 Nonlinear-ode dynamics

We consider a controller for an autonomous car. The car changes velocity and direction non-deterministically. The dynamics of the car as follows.

$$\dot{x} = v \cos\theta, \ \ \dot{y} = v \sin\theta, \ \ \dot{\theta} = v \tan\phi$$

$$\dot{v} = \begin{cases} -cv & if\ m_2 = stay \\ 5 & if\ m_2 = acc \\ -5 & if\ m_2 = dec \end{cases}$$

We designed four STL formulas in a similar way to the previous models.

STL formula	ID	Explanation
$\Diamond_{[10,50)} x > 60$	φ_1	Within 40 time units, x position of the car is greater than 60.

$(\theta < 30)\ U_{[4,10]}\ (y < 40)$	φ2	For the interval [4,10], y position of the car is less than 40, until θ is less than 30.
$\Diamond_{[20,50]}(\Box_{[15,20]}(x > 70))$	φ3	Within 30 time units, x position of the car is greater than 70 for 5 time units.
$\Box_{[0,10]}((y < 70)\ \mathrm{U}_{[10,20]}(\phi > -5)$	φ4	For 10 time units, ϕ is greater than -5, until y position of the is less than 70.

1.4 Railroad

This is a system of modeling a crossing barrier controller and a train on a track. There is a circular railroad track and there is a crossing barrier on the track. The tracks are 100 meters long. A train is going around and around the track. The angular between ground and the crossing barrier $theta$ changes depending on its mode $t \in \{Far, Approach, Near, Past\}$. Initially, the relative distance of the train to the barrier $distance(t,b)$ is $60 \leq distance(t,b) \leq 70$, the train far away from the crossing barrier.

The mode of the train is determined by the distance between the train and the crossing barrier. If the distance is $distance(t,b) \geq 50$, then the train's mode is Far. If the $distance(t,b)$ is $40 \leq distance(t,b) < 50$, then the train's mode is $Approach$. If the $distance(t,b)$ is $20 \leq distance(t,b) < 30$, then the train's mode is $Near$. If the $distance(t,b)$ is $-5 \leq distance(t,b) < 0$, then the train's mode is $Past$. If the $distance(t,b)$ is $-10 \leq distance(t,b) < -5$, then the train's mode is Far and the relative distance is updated to '$100 + current\ relative\ distance'$. The dynamics of the train and the crossing barrier are as follows.

1. Linear dynamics

$$\dot{theta} = \begin{cases} 0 & if\ t = Far \\ 5 & if\ t = Approach \\ 10 & if\ t = Near \\ -5 & if\ t = Past \end{cases} \qquad \dot{trainPosition} = -5$$

2. Polynomial dynamics

$$\dot{theta} = \begin{cases} 0 & if\ t = Far \\ cv_{approach} & if\ t = Approach \\ cv_{near} & if\ t = Near \\ cv_{past} & if\ t = Past \end{cases} \qquad \dot{v}_t = \begin{cases} 5 & if\ t = Approach \\ 10 & if\ t = Near \\ -5 & if\ t = Past \end{cases}$$

$$\dot{trainPosition} = -5$$

where $trainPosition$ is the current degree between the crossing bar and ground. The variable cv_i is a constant (with derivative 0) that captures the value of v_i at a certain moment, so that the dynamics is a polynomial. We designed STL formulas for the model in a similar way to the previous models.

STL formula	ID	Explanation
$\Diamond_{[0,20]}(pos < -5)$	φ1	Within time interval [0,20], the position of the train is less than -5.

$\square_{[0,50)}(pos > 0)$	φ2	For time interval [0,50], the position of the train is greater than 0.
$(bar < 10)\ U_{[10,40]}(\Diamond_{[0,20]}(pos < 40))$	φ3	Within 20 time units, the position of the train is less than 40, until, the position of the bar is less than 10.
$\square_{[0,40]}((bar > 80) \rightarrow \Diamond_{[0,20)}(pos > 10))$	φ4	For 40 time units, if the bar position is greater than 80, the train position is greater than 10 within 20 time units.

1.5 Battery

There are two of fully charged batteries, and a control system switches load between these batteries to achieve longer lifetime out of the batteries. There are three modes $m_i \in \{on, off, dead\}$ for each battery. Initially, the total energy of $battery_1$ is 8.5 and $battery_2$ is 7.5. The both batteries are switched on.

The behavior of each battery is as follows. If the total energy of the battery is smaller than $(1-c) * its\ kinetic\ energy$, $c \in [0,1]$ is threshold, then the battery is dead. Otherwise, the battery can be either turned on or turned off. The dynamics of two batteries are as follows.

1. Linear dynamics

.

$$\dot{d}_1 = \dot{d}_2 = \frac{1}{2C}, \quad \dot{g}_1 = \dot{g}_2 = -\frac{1}{2} \qquad if\ m_1 = on, m_2 = on$$

$$\dot{d}_i = \frac{0.7}{C}, \dot{d}_j = -C, \quad \dot{g}_i = -1, \dot{g}_j = 0 \qquad if\ m_i = on, m_j = off, i \neq j, i,j \in \{1,2\}$$

$$\dot{d}_i = \frac{0.7}{C}, \dot{d}_j = 0, \quad \dot{g}_i = -1, \dot{g}_j = 0 \qquad if\ m_i = on, m_j = dead, i \neq j, i,j \in \{1,2\}$$

$$\dot{d}_1 = \dot{d}_2 = 0, \quad \dot{g}_1 = \dot{g}_2 = 0 \qquad if\ m_1 = on, m_2 = on$$

2. Polynomial dynamics and nonlinear-ode

$$\dot{d}_i = \frac{L}{C} - kd_i, \quad \dot{g}_i = -L \qquad if\ m_i = on$$

$$\dot{d}_i = kd_i, \quad \dot{g}_i = 0 \qquad if\ m_i = off$$

$$\dot{d}_i = 0, \quad \dot{g}_i = 0 \qquad if\ m_i = dead$$

where variable d_i is its kinetic energy, variable g_i is its total charge, and constant $C \in [0,1]$ is its threshold. For polynomial dynamics, the variable cd_i is a constant (with derivative 0) that captures the value of d_i at a certain moment to make the dynamics to polynomial function. The following STL formulas are considered for linear and polynomial models.

STL formula	ID	Explanation
$\lozenge_{[0,100]} d_1 > 0.2$	φ_1	Within 100 time units, the total charge of $battery_1$ is greater than 0.2.
$(g_2 > 0.5)\ R_{[0,40]}\ dead_2$	φ_2	Within the time interval [0,40], $battery_2$ is dead, thereafter the total energy of $battery_2$ is greater than 0.5.
$\lozenge_{(10,50]}(g_2 \geq 0\ U_{(1,15)}(d_1 < 0.2))$	φ_3	Within time interval (10,50], if the total charge of $battery_1$ is less than 0.2, the total energy of $battery_2$ is greater than or equal to 0.
$\square_{[0,40]}(g_2 > 4 \rightarrow \lozenge_{[0,20)} d_2 > 0)$	φ_4	For time interval [0,40], if the total energy of $battery_2$ is greater than 4, the total charge of $battery_2$ is less than 0 within 20 time units.

1.6 Airplain

There are controllers for turning an airplane. An aircraft turns by controlling two ailerons and a rudder. There are four different with different rotation directions and angles. Initially, the yaw angle (β), the rolling moment (p), the yawing moment (r), the roll angle (ϕ), two ailerons ($xAIL, gAIL$), and rudders ($xRDR, gRDR$) are all zero.

The controller makes a turn according to the value of ailerons and rudders every 0.5 time units. Dynamics of each variables in the model are defined as follows.

$$\dot{\beta} = yBTA * \beta - r + \left(\frac{g}{vT}\right) * \phi + yRDR * xRDR$$
$$\dot{p} = lBTA * \beta + lP * r + lAIL * xAIL + lRDR * xRDR$$
$$\dot{r} = nBTA * \beta + nP * p + nR * r + nAIL * xAIL + nRDR * xRDR$$
$$\dot{\phi} = p,\ \ g\dot{AIL} = 0,\ \ g\dot{RDR} = 0,\ \ t\dot{ime} = 1$$
$$\begin{cases} x\dot{AIL} = 0.25,\ x\dot{RDR} = 0.5 & if\ m_2 = stay \\ x\dot{AIL} = 0.25,\ x\dot{RDR} = -0.5 & if\ m_2 = stay \\ x\dot{AIL} = -0.25,\ x\dot{RDR} = 0.5 & if\ m_2 = stay \\ x\dot{AIL} = -0.25,\ x\dot{RDR} = -0.5 & if\ m_2 = stay \end{cases}$$

The following STL formulas are considered for the airplain model with nonlinear-ode dynamics. We designed these formulas in a similar way to the previous models.

STL formula	ID	Explanation
$(\phi > 0)\ R_{[10,20]}\ (p < 0)$	φ_1	Within 100 time units, the rolling moment is less than 0, thereafter the roll angle is greater than 0.
$\lozenge_{[0,100)} xAIL \geq 0$	φ_2	Within the time interval [0,100], x ailerons is greater than or equal to 0.
$\square_{[5,35]}(p > 0\ U_{[2,8]} \beta < 1)$	φ_3	For time interval [5,35], the yaw angle is less than 1, until the rolling moment is greater than 0.

$\Diamond_{[10,40]}(\beta < -0.2 \rightarrow \Diamond_{[5,15]} p < 0)$	φ_4	Within time interval [10,40], if the yaw angle is less than -0.2, the rolling moment is less than 0 within 10 time units.

2 STL formulas for STL satisfiability checking

We perform STL bounded satisfiability checking on STL properties using our algorithm (new) and the previous algorithm (old) respectively, up to step bound n = 20. We consider 10 STL formulas for each nesting depth $d = 1, 2, 3$. We randomly generated formulas considering the frequency of unary temporal operators, □ and ◊, and binary temporal operators, U and R, similar. Please refer model files in the "experiment/exp2" directory for details.

References

1. Kyungmin Bae and Sicun Gao: Modular SMT-Based Analysis of Nonlinear Hybrid Systems. In Formal Methods in Computer-Aided Design (2017), pp 180-187.
2. T. A. Henzinger: The theory of hybrid automata. In Logic in Computer Science (1996).
3. A. Platzer. Logical analysis of hybrid systems: proving theorems for complex dynamics. Springer Science & Business Media, 2010.
4. Steven M La Valle: Planning algorithms. Cambridge university press (2006).