Dynamic logic for Hybrid systems

Differential Dynamic Logic for Verifying Parametric Hybrid Systems

by

Andre Platzer

presented by

Hallstein Asheim Hansen

An example of a hybrid system: Thermostat

A room is heated by a radiator controlled by a thermostat.

When the radiator is off, the temperature decreases (over time) according to the differential equation:

$$\dot{y} = -ay$$

When the radiator is on, the temperature increases towards 30:

$$\dot{y} = -a(y - 30)$$

Solution: Solve the equation...

$$\dot{y} = -ay, \qquad \dot{y} + ay = 0$$

$$e^{at}\,\dot{y} + e^{at}ay = 0, \qquad e^{at}\,\dot{y} + e^{at}y = 0$$

$$(e^{at}y)' = 0, \qquad \int (e^{at}y)' = \int 0 + C$$

$$e^{at}y = C, \qquad y = Ce^{-at}$$

Dynamic logic for Hybrid systems

The other equation is $y(t) = 30a - De^{-at}$

Can we combine the two equations in this way:?

$y(t) = Ce^{-at}$ if $y(t) > 21$

$y(t) = 30a - De^{-at}$ if $y(t) < 19$

Dynamic logic for Hybrid systems

No, we want to express system properties like:

- If the temperature rises above 21, then the raditator should be switched off
- If the temperature falls below 19, then the raditator should be switched on
- We accept a small error margin of one degree. (Radiator *may* switch off at 21 degrees, *must* switch off at 22).

Dynamic logic for Hybrid systems

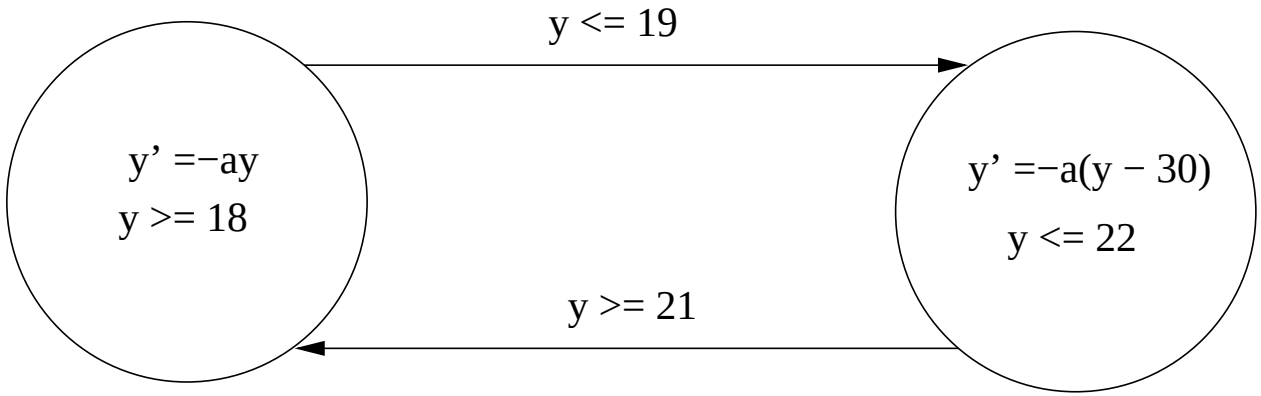
Hybrid system

According to Wikipedia, the accurate source of all wisdom: *A hybrid system is a dynamic system that exhibits both continuous and discrete dynamic behavior*

Modelling and analysis of hybrid systems combines techniques from mathematics (behavior) and computer science (statefulness, nondeterminism).

Dynamic logic for Hybrid systems

The traditional way of formally modelling hybrid systems is through hybrid automata:



In this paper, however, a state-transition system is used, combined with a dynamic logic.

Dynamic logic for Hybrid systems

d$\mathcal{L}$

The *differential dynamic logic* is a first-order dynamic logic. Its programs are *hybrid programs*. The definition of the hybrid programs motivates the semantics and verification calculus of d$\mathcal{L}$

Dynamic logic for Hybrid systems

Hybrid programs

- V is the set of variables, such as x.
- $Trm(V)$ is the set of terms - *polynomials*, for example $x^2 + y + c$.
- $Fml(V)$ is the set of formulas, such as $x > 0 \wedge x < 42$.

Hybrid programs consist of:

- Discrete jumps (variable resets): $x := \theta$, where x is a variable and θ a term.
- Continuous evolutions: $\dot{x}= \theta \& \chi$, where χ defines the region the evolution must remain in.
- $\alpha; \gamma$, $\alpha \cup \gamma$, α^* and $?\chi$ represent sequential composition, nondeterministic choice, nondeterministic repetition and condition.

Dynamic logic for Hybrid systems

Modalities

We want to relate programs, α, with requirements, ϕ:

We write: $[\alpha]\phi$

This means that: ϕ must hold after every terminating execution of the program α.

($< \alpha > \phi$ means that ϕ must hold after at least one (terminating) execution of α)

Example program

$$[x := 0; \dot{x}= 2](x \geq 0 \land x \leq 42)$$

First, x is set to 0. Then x evolves continuously according to $\dot{x}= 2$. Does the property hold?

States

The variables will have certain values at every point in our program - that's the *state*. For example: $\nu = \{x = 42, y = 666\}$.

- $Sta(V)$ is the set of all states.
- $val(\nu, x)$ gives us the value of the variables x in state ν.
  $val(\nu, x) = 42$

State transitions

A state transition is a relation between a state before the execution of a program, and a state after the execution of a program.

$(x = 42, x = 0)$ is a state transition.

The set of all state transitions of a program α is called the *semantics* of that program, $\rho(\alpha)$.

$(x = 42, x = 0) \in \rho(x := 0)$

$(x = 666, x = 0) \in \rho(x := 0)$

Dynamic logic for Hybrid systems

Example program

A Formula 1 racing car (x) does 0-100 km/h in 3.7 seconds, a while a Lada (y) does it, eventually, in 11.2 seconds. Assuming a constant acceleration, how fast are they going after 10 seconds (τ)?

$$\tau := 0; x := 0; y := 0;$$

$$((?\tau < 10); \dot{x}{=}\ 27, \dot{y}{=}\ 8.9, \dot{\tau}{=}\ 1 \& \tau \leq 10)^*; (?\tau >= 10)$$

We usually rewrite this to:

$$\tau := 0; x := 0; y := 0;$$

$$\mathit{while}\ (?\tau < 10)\ \mathit{do}\ \ \dot{x}{=}\ 27, \dot{y}{=}\ 8.9, \dot{\tau}{=}\ 1 \& \tau \leq 10$$

We know the velocities at time 0 (0 km/h!), but we want to know the function that computes it at time 10.

We can construct those functions easily. Why? Because we know the derivatives of the functions! If $\dot{x}= 27$, then $x(t) = 27t + C$. The constant C is the function's value at time 0: $x(t) = 27t + 0$

Dynamic logic for Hybrid systems

Thus we have:

- $x(t) = 27t$
- $y(t) = 8.9t$
- $\tau(t) = t$

The function x, y and τ are, of course, continuous on the interval $(0, 10)$, and the derivative never changes from the formula specified.

Dynamic logic for Hybrid systems

How can we relate these functions to states and transitions?

Like this:

Start state : $\{x = 0, y = 0, \tau = 0\} = \{x(0), y(0), \tau(0)\}$

We also have a state for each value in the $(0, 10)$ interval:

$\{x(1), y(1), \tau(1)\} = \{x = 27, y = 8.9, \tau = 1\}$

$\{x(2), y(2), \tau(2)\} = \{x = 54, y = 17.8, \tau = 2\}$

$\{x(3.14), y(3.14), \tau(3.14)\} = \{x = 84.78, y = 27.94, \tau = 3.14\}$

Dynamic logic for Hybrid systems

Let's define a function f. It takes a number as input, and produces a desirable state as output:

$$f(t) = \{x(t), y(t), \tau(t)\}, t \in [0, 10]$$

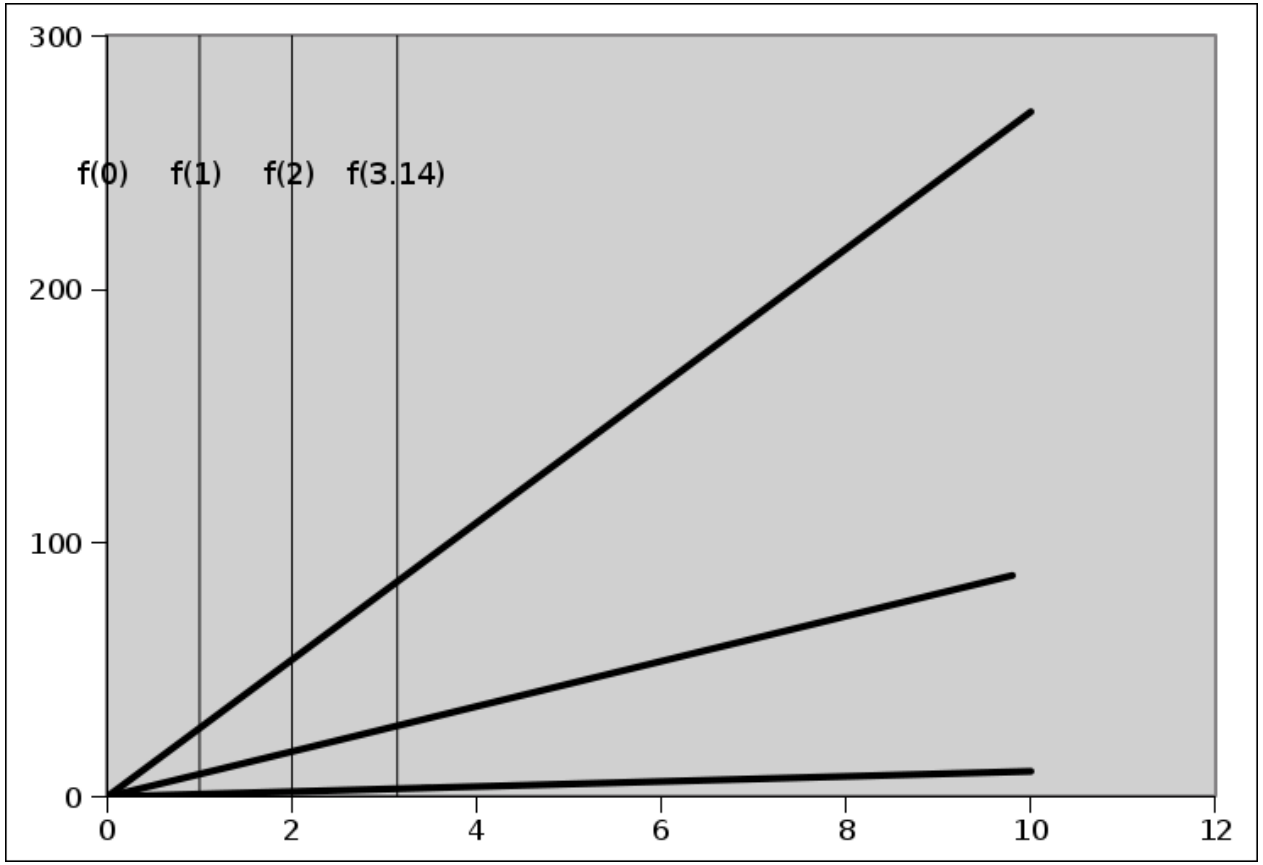
We have, because the *val* function gives us the value of a variable in a state, the following:

$val(f(1), x) = x(1) = 27$

$val(f(2), y) = y(2) = 17.8$

$val(f(3.14), \tau) = \tau(3.14) = 3.14$

Illustration of continuous evolution



Dynamic logic for Hybrid systems

Semantics: Continuous evolution

$(\nu, \omega) \in \rho(\dot{x} = \theta \& \chi) :\Leftrightarrow$ there is a function $f : [0, t] \to Sta(V), t \geq 0$ such that:

- $f(0) = \nu, f(t) = \omega$
- $val(f(\zeta), x)$ is continuous in ζ on $[0, t]$
- $val(f(\zeta), \theta)$ is the derivative of f at each time $\zeta \in (0, t)$.
- For $y \neq x$ and $\zeta \in (0, t)$ we have that $val(f(\zeta), y) = val(\nu, y)$
- $val(f(\zeta), \chi) = true$ for each $\zeta \in (0, t)$

If we have several differential equations then they they are treated accordingly.

Dynamic logic for Hybrid systems

Semantics (simplified!)

- $x := d$, $(\nu, \nu[x \mapsto d])$
- $\chi?$, (ν, ν) if χ
- $\alpha; \gamma$, $(\nu, z); (z, \omega)$ gives (ν, ω)
- $\alpha \cup \gamma$, $\rho(\alpha) \cup \rho(\gamma)$
- α^*, $(\nu_0, \nu_1); \ldots; (\nu_i, \nu_{i+1})$ gives $(\nu_i, \nu_{i+1}), 0 \leq i$

Example: 'Speed supervision in Train Control'

Part of the safety measures of trains is that they must be told how much further they can move safely.

A train driver must then consider something like: 'I have driven 34 km, I can drive until the 35 km mark, and my speed is 120 km/h. When I hit the brakes, the speed decreases at 1 km/h per second. Am I able to stop?'

Dynamic logic for Hybrid systems

Example: 'Speed supervision in Train Control'

- Let m be the 'mark' (35000 m)
- Let z be the 'current position' (34000 m)
- Let s be the safety margin (500 m)
- Let ν be the speed (35 m/s)
- Let b be the brake force (1 m/s^2)
- Let a be the (de)acceleration
- Let τ be the time since the driver last considered his options
- Let ϵ be maximal time the driver may spend between considerations (5 s)

Dynamic logic for Hybrid systems

Hybrid program

$$\psi \rightarrow [(corr; drive)^*]z \leq m$$

$$corr \equiv (?m - z < s; a := -b) \cup (?m - z \geq s; a := 0)$$

$$drive \equiv \tau := 0; (\dot{z} = v, \dot{v} = a, \dot{\tau} = 1 \& v \geq 0 \wedge \tau \leq \epsilon)$$

We will examine the program in detail by simulating an execution.

Dynamic logic for Hybrid systems

Outer program

$$[(corr; drive)^*]z \leq m$$

Informally:

'Whenever we *corr*ect the speed and *drive* the train, we should never pass our mark.'

Let the start state ν be
$\{v = 35, z = 34000, m = 35000, b = 1, a = 0, s = 500, \epsilon = 5, \tau = 0\}$

Dynamic logic for Hybrid systems

corr

$$(?m - z < s; a := -b) \cup (?m - z \geq s; a := 0)$$

ν:

$\{v = 35, z = 34000, m = 35000, b = 1, a = 0, s = 500, \epsilon = 5, \tau = 0\}$

Let's compute ρ of this program, assuming ν. We need the union of the left and right sides.

Left side is $\emptyset$ because
$(?m - z < s) = (?35000 - 34000 < 500) = false.$

Right side:

$(?m - z \geq s) = (?35000 - 34000 \geq 500) = true$ gives us (ν, ν)

$a := 0$ gives us $(\nu, \nu[a \mapsto 0]) = (\nu, \nu)$

'We are not inside the safety margin, do not adjust speed by (de)accelerating.'

drive

$$\tau := 0; (\dot{z}{=}\, v, \dot{v}{=}\, a, \dot{\tau}{=}\, 1 \& v \geq 0 \wedge \tau \leq \epsilon)$$

ν:$\{v = 35, z = 34000, m = 35000, b = 1, a = 0, s = 500, \epsilon = 5, \tau = 0\}$

First, τ is set to 0, not changing ν. Then we need to calculate the values of the f function for each variable:

Let $t = 5$. $val(\omega, .) = val(f(5), .)$:

- $val(\omega, z) = z(5) = 35 * 5 + 34000 = 34175$
- $val(\omega, v) = v(5) = 0 * 5 + 35 = 35$
- $val(\omega, \tau) = \tau(5) = 1 * 5 + 0 = 5$

corr

Later...

$$(?m - z < s; a := -b) \cup (?m - z \geq s; a := 0)$$

ν:

$\{v = 35, z = 34525, m = 35000, b = 1, a = 0, s = 500, \epsilon = 5, \tau = 0\}$

The left side is not empty now because

$(?m - z < s) = (?35000 - 34525 < 500) = true.$

The ω state is now $\nu[a \mapsto -b] = \nu[a \mapsto -1]$

Dynamic logic for Hybrid systems

drive

$$\tau := 0; (\dot{z}{=}\, v, \dot{v}{=}\, a, \dot{\tau}{=}\, 1 \& v \geq 0 \wedge \tau \leq \epsilon)$$

ν:
$\{v = 35, z = 34525, m = 35000, b = 1, a = -1, s = 500, \epsilon = 5, \tau = 5\}$

First, τ is set to 0. Second, we see that the speed v now changes over time, and this has effect on the evolution of z (the train won't travel as far when it slows down).

- $val(\omega, v) = val(\omega, v(5)) = -1 * 5 + 35 = 30$
- $val(\omega, z) = val(\omega, z(5)) = \int_0^5 (-1t + 35)dt + 34525 =$
  $[-0.5t^2 + 35r]_0^5 + 34525 = 162.5 + 34525 = 34687.5$

drive

Since *corr* won't change anything (the train is braking until it stops) we will combine several instances of *drive* for brevity (time now passes for 19 seconds):

- $val(\omega, v) = val(\omega, v(19)) = -1 * 19 + 35 = 11$
- $val(\omega, z) = val(\omega, z(19)) = \int_0^{19}(-1t + 35) + 34525 = [-0.5t^2 + 35t]_0^{19} + 34525 = 484.5 + 34525 = 35009.5$

After 19 seconds the train unfortunately crosses the 'mark'.

(The train stops after 35 seconds with $z = 35137.5$)

Note!

The hybrid programs are *not* meant to be executed, but analyzed.
The previous example was just to explain how they work!

Verification calculus

Assuming a system on the form:

$$\psi \rightarrow [\alpha]\phi$$

A sequent calculus is used for verfying whether α always will satifsfy ϕ, given initial requirement ψ.

Rules of the calculus

The rules are comprised of the following:

- Rules of propositional logic
- Rules of dynamic logic
- Rules for discrete change and continuous evolution
- Rules for quantifier elimination and an induction schema

Dynamic logic for Hybrid systems

Dynamic logic

$$D1 : \frac{\phi \wedge \psi}{<?\phi > \psi} \qquad D2 : \frac{\phi \rightarrow \psi}{[?\phi]\psi} \qquad D3 : \frac{< \alpha > \phi \vee < \gamma > \phi}{< \alpha \cup \gamma > \phi}$$

$$D4 : \frac{[\alpha]\phi \wedge [\gamma]\phi}{[\alpha \cup \gamma]\phi} \qquad D5 : \frac{\phi \vee < \alpha; \alpha^* > \phi}{< \alpha^* > \phi}$$

$$D6 : \frac{\phi \wedge [\alpha; \alpha^*]\phi}{[\alpha^*]\phi} \qquad D7 : \frac{< [\alpha] >< [\gamma] > \phi}{< [\alpha; \gamma] > \phi}$$

Discrete change

D8:

$$\frac{\phi_x^\theta}{< [x := \theta] > \phi}$$

Only applicable if the substitution introduces no new bindings

Continuous evolution

The rule D14 says that we can substitute the derivatives of functions for the functions themselves (we go from $\dot{x}= \theta$ to $x := y_x(t)$). The parameter t (time) may have many values, so we introduce an universial quantifier.

$$\frac{\forall t \geq 0(\overline{\chi} \rightarrow [x := y_x(t)]\phi)}{[\dot{x}= \theta \& \chi]}\phi$$

So what is $\overline{\chi}$? An assertion that χ is never violated by any value that t may have during the continuous evolution:
$\forall 0 < \overline{t} < t, [x := y_x(\overline{t})]\chi$

Dynamic logic for Hybrid systems

D13 is the corresponding rule for the <> modality.

$$\frac{\exists t \geq 0(\bar{\chi} \rightarrow < x := y_x(t) > \phi)}{< \dot{x} = \theta \& \chi >} \phi$$

The tool Mathematica is used in the author's implementation for D13 and D14.

Induction schema

D14:

$$\frac{\vdash p \qquad \vdash [\alpha^*](p \rightarrow [\alpha]p)}{\vdash [\alpha^*]p}$$

Quantifier elimination

D9 (D10, D11 and D12 are similar)

$$\frac{qelim(\exists x \wedge_i (\Gamma_i \vdash \Delta_i))}{\Gamma \vdash \Delta, \exists x \phi}$$

These rules combine side deductions with quantifier elimination.

Dynamic logic for Hybrid systems

Let's demonstrate the rule D9 (and some of the previous rules) with an example: We are interested in knowing the following: If we start an emergency brake procedure - can we expect to cross the 'mark'?

Note!

We are now interested in universal properties of the system. For example whether $z \leq m$ instead of one special case where $34000 < 35000$.

Dynamic logic for Hybrid systems

If we start an emergency brake procedure - can we expect to cross the 'mark'?

$$\vdash v > 0 \land z < m \rightarrow <\dot{z}= v, \dot{v}= -b > z \geq m$$

We use a (well-known) rule from propositional logic before we come to the interesting stuff:

$$\frac{v > 0, z < m \vdash <\dot{z}= v, \dot{v}= -b > z \geq m}{\vdash v > 0 \land z < m \rightarrow <\dot{z}= v, \dot{v}= -b > z \geq m}$$

Dynamic logic for Hybrid systems

The rule D13 can now be applied:

$$\frac{v > 0, z < m \vdash \exists t \geq 0 < z := -\frac{b}{2}t^2 + vt + z > z \geq m}{v > 0, z < m \vdash <\dot{z}= v, \dot{v}= -b > z \geq m}$$

Dynamic logic for Hybrid systems

Now we need to remove the quantifier, which we do by a side deduction where we make t a free variable and try to remove it from inside the modalities.

From...

$$v > 0, z < m \vdash \exists t \geq 0 < z := -\frac{b}{2} + vt + z > z \geq m$$

...to start of side deduction...

$$v > 0, z < m \vdash t \geq 0 \wedge < z := -\frac{b}{2}t^2 + vt + z > z \geq m$$

... and, by propositional logic, get

$$\frac{v > 0, z < m \vdash t \geq 0 \qquad v > 0, z < m \vdash < z := -\frac{b}{2}t^2 + vt + z > z \geq m}{v > 0, z < m \vdash t \geq 0 \wedge < z := -\frac{b}{2}t^2 + vt + z > z \geq m}$$

Dynamic logic for Hybrid systems

We are done with the left branch, on the right branch we can use substitution (D8):

$$\frac{v > 0, z < m \vdash -\frac{b}{2}t^2 + vt + z \geq m}{v > 0, z < m \vdash < z := -\frac{b}{2}t^2 + vt + z > z \geq m}$$

Dynamic logic for Hybrid systems

Now we feed the conjuction of the open branches into Mathematica (or a similar tool), in order to eliminate the quantifier:

$qelim(\exists t(v > 0, z < m \vdash t \geq 0 \wedge -\frac{b}{2}t^2 + vt + z \geq m))$

Quantifier elimination - manually

$-\frac{b}{2}t^2 + vt + (z - m) \geq 0$ is a quadratic (in)equation. It has solutions only if the polynomial has roots:

$$t = \frac{-B \overset{+}{-} \sqrt{B^2 - 4AC}}{2A}, \frac{-v \overset{+}{-} \sqrt{v^2 - 4(-\frac{b}{2})(z-m)}}{2(-\frac{b}{2})} = \frac{v \overset{+}{-} \sqrt{v^2 + 2b(z-m)}}{b}$$

So, the root only exists if $v^2 + 2b(z - m) \geq 0$, that is $v^2 \geq 2b(m - z)$.

Soundness and incompleteness

All the rules of the logic are locally sound, but the logic is incomplete.