Formal Simulation and Visualisation of Hybrid Programs in Lince

Pedro Mendes, Ricardo Correia, Renato Neves, **José Proença**

FMAS @ iFM, 11 Nov 2024

CISTER
Research Center in
Real-Time & Embedded
Computing Systems

INESCTEC

University of Minho
School of Engineering

This work is financed by National Funds through FCT - Fundação para a Ciência e a Tecnologia, I.P. (Portuguese Foundation for Science and Technology) within the project IBEX, with reference 10.54499/PTDC/CCI-COM/4280/2021. This work is also partially supported by National Funds through FCT/MCTES, within the CISTER Unit (UIDP/UIDB/04234/2020); and by the EU/Next Generation, within the Recovery and Resilience Plan, within project Route 25 (TRB/2022/00061 -- C645463824-00000063).

Ibex

FCT Fundação para a Ciência e a Tecnologia

REPÚBLICA PORTUGUESA

Formal Simulation and Visualisation of Hybrid Programs in Lince

José Proença
CISTER & U. Porto
Portugal

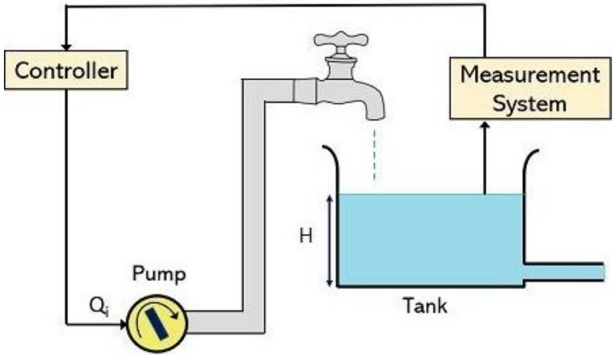
Renato Neves
INESC-TEC & U. Minho
Portugal

Ricardo Correia
U. Minho
Portugal

Daniel Mendes
U. Minho
Portugal

Hybrid systems

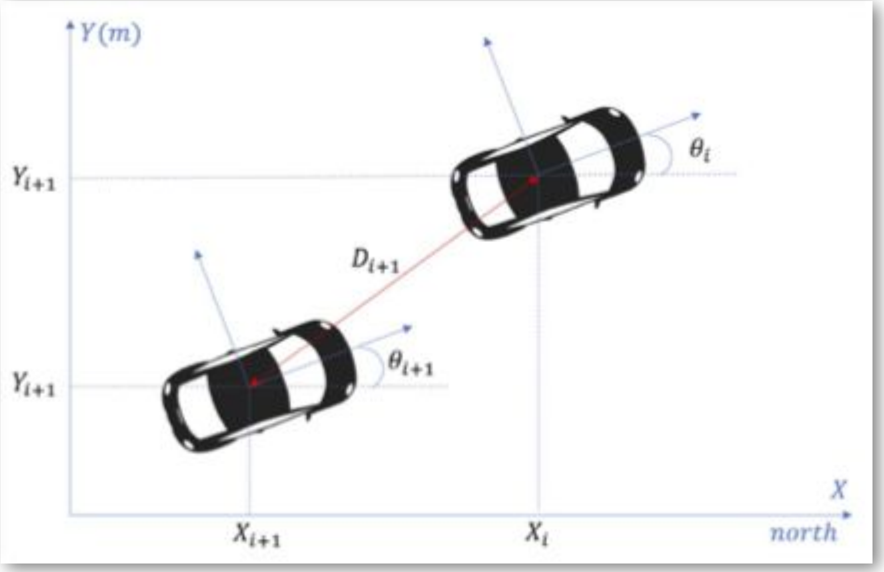
Computational devices that interact with **physical** environment



Another hybrid system

Platooning

- Acceleration (1D)
- Steering (2D)
- Failures



By Ênio Filho, Anis Koubâ, Ricardo Severino, Eduardo Tovar @ CISTER

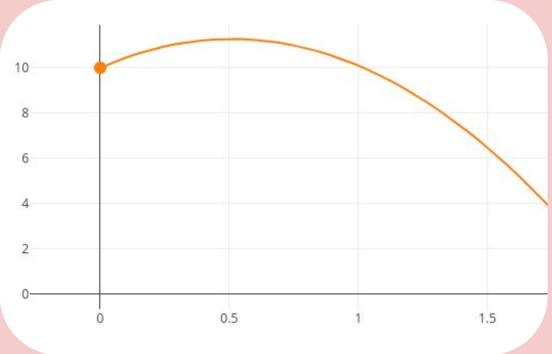
Discrete behaviour

+

Continuous behaviour

```
...
v := 10;
while v <= 10 do {
  ...
}
```

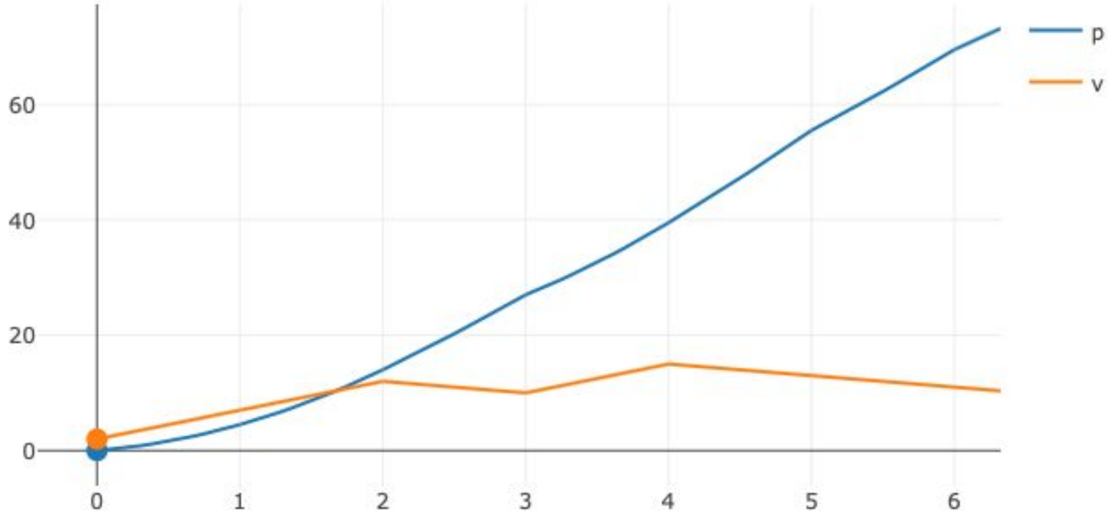
$$\begin{cases} v' = -9.8 \\ p' = v \end{cases}$$




```
p:=0; v:=2;
while true do {
  if v<=10
  then p'=v,v'=5  for 1
  else p'=v,v'=-2 for 1
}
```

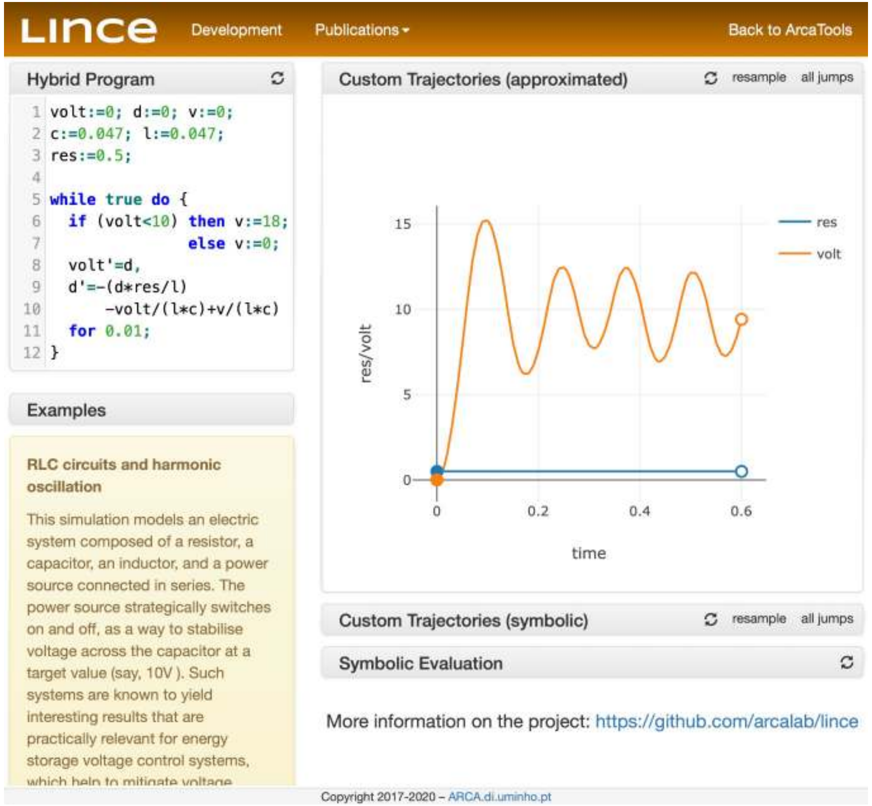
A cruise controller in Lince

```
// Cruise control
p:=0; v:=2;
while true do {
  if v<=10
  then p'=v,v'=5  for 1
  else p'=v,v'=-2 for 1
}
```



The expected output

Why Lince?



- **No installation**
  - just a website (+ server)
- **Easy** to experiment
- **Simple** language
  - No need for complex frameworks
- **Precise** semantics
- Low effort to **extend**
  - new extensions
  - involve students
  - involve partners

Hybrid programs - **today**

```
// Cruise control
p:=0; v:=2;
while true do {
  if v<=10
  then p'=v,v'=5  for 1
  else p'=v,v'=-2 for 1
}
```

1. Syntax & semantics

2. Examples

3. Recent extensions

4. Roadmap

Syntax

Discrete control

```
p,q ∋ x := t
    | p ; q
    | if b then p else q
    | while b do { p }
    | deq
```

Continuous control

deq ∋ $x_1' = t_1, \ldots, x_n' = t_n$

(systems of differential equations)

```
t,s ∋ real | real * x | t + s | x
```

(linear terms)

Syntax

Discrete control

```
p,q ∋ x := e
    | p ; q
    | if e then p else q
    | while b do { p }
    | deq
```

Continuous control

$$\mathbf{deq} \ni x_1' = t_1, \ldots, x_n' = t_n$$

(systems of differential equations)

$t, s \ni \text{real} \mid \text{real} * x \mid t + s \mid x$ (linear terms)

new

$e \ni e \mid f(e_1, \ldots, e_1)$ (non-linear terms)

(Big-Step) Semantics

(diff-skip)
$$\frac{[\![e]\!](\sigma) = t}{\vec{x}' = \vec{\ell}\ \texttt{for}\ e, \sigma, t \Downarrow skip, \sigma[\vec{x} \mapsto \phi(t)]}$$

(diff-stop)
$$\frac{[\![e]\!](\sigma) > t}{\vec{x}' = \vec{\ell}\ \texttt{for}\ e, \sigma, t \Downarrow stop, \sigma[\vec{x} \mapsto \phi(t)]}$$

(diff-err)
$$\frac{[\![e]\!](\sigma)\ \text{undefined}}{\vec{x}' = \vec{\ell}\ \texttt{for}\ e, \sigma, t \Downarrow err}$$

(asg-skip)
$$\frac{[\![e]\!](\sigma)\ \text{defined}}{x := e, \sigma, 0 \Downarrow skip, \sigma[x \mapsto [\![e]\!](\sigma)]}$$

(asg-err)
$$\frac{[\![e]\!](\sigma)\ \text{undefined}}{x := e, \sigma, t \Downarrow err}$$

(seq-skip)
$$\frac{\texttt{p}, \sigma, t \Downarrow skip, \tau \qquad \texttt{q}, \tau, u \Downarrow v}{\texttt{p;q}, \sigma, t + u \Downarrow v}$$

(if-rules) … **(while-rules) …**

(Big-Step) Semantics

(diff-stop)
$$\frac{[\![e]\!](\sigma) > t}{\vec{x}' = \vec{\ell}\ \texttt{for}\ e, \sigma, t \Downarrow stop, \sigma[\vec{x} \mapsto \phi(t)]}$$

Statement (syntax)

State (var -> value)

Target time (real value)

(Big-Step) Semantics

$$\textbf{(diff-stop)} \quad \frac{[\![e]\!](\sigma) > t}{\vec{x}' = \vec{\ell}\ \texttt{for}\ e, \sigma, t \Downarrow \boxed{\mathit{stop}}, \boxed{\sigma[\vec{x} \mapsto \phi(t)]}}$$

Statement
(syntax)

State
(var -> value)

Target time
(real value)

Reached the end?

Final state (var -> value)

(Big-Step) Semantics

$$\textbf{(diff-err)} \qquad \frac{[\![e]\!](\sigma)\ \boxed{\text{undefined}}}{\vec{x}' = \vec{\ell}\ \texttt{for}\ e, \sigma, t \Downarrow \boxed{err}}$$

new

Partial functions
(can throw errors)

Hybrid programs in Lince

- Demo -

Run in **our server**

https://arcatools.org/lince

- *Internet browser*

Download and run **your server**

https://github.com/arcalab/lince

- SageMath (http://www.sagemath.org/)
- SBT (https://www.scala-sbt.org)
- Java runtime
- *Internet browser*

Examples

```
// Simple composition
p:=0; v:=0;
p'=v,v'=-2  for 1;
p'=v,v'=-2  for 1;
```

```
// Initial values of the water tank
level := 5; drain := -1/2;

while true do {
  // keep level between 3..10
  if        level<=3   then drain:= 1/2;
  else if level>=10 then drain:=-1/2;
  else      skip;
  level'= drain, drain'=0 for 0.1;
}
```

More Examples

```
// Cruise control
p:=0; v:=2;
while true do {
  if v<=10
  then p'=v,v'=5  for 1
  else p'=v,v'=-2 for 1
}
```

```
// Adaptive Cruise control
p:=0;    v:=0;    a:=5; b:=-2; // follower
pl:=50; vl:=10; al:=0;         // leader
period:=1;

while true do {
  ???
}
```

More Examples

```
// Cruise control
p:=0; v:=2;
while true do {
  if v<=10
  then p'=v,v'=5  for 1
  else p'=v,v'=-2 for 1
}
```

```
// Adaptive Cruise control
p:=0;    v:=0;   a:=5; b:=-2; // follower
pl:=50; vl:=10; al:=0;        // leader
period:=1;

while true do {
  if ((p+v*period+ a/2*period^2 <
       pl+vl*period+al/2*period^2) &&
      (((v-vl+(a-al)*period)^2 -
        4*(p-pl+(v-vl)*period +
        (a-al)/2*period^2)*(b-al)/2) < 0))
  then p'=v,v'=a,pl'=vl,vl'=al for period;
  else p'=v,v'=b,pl'=vl,vl'=al for period;
}
```

More Examples

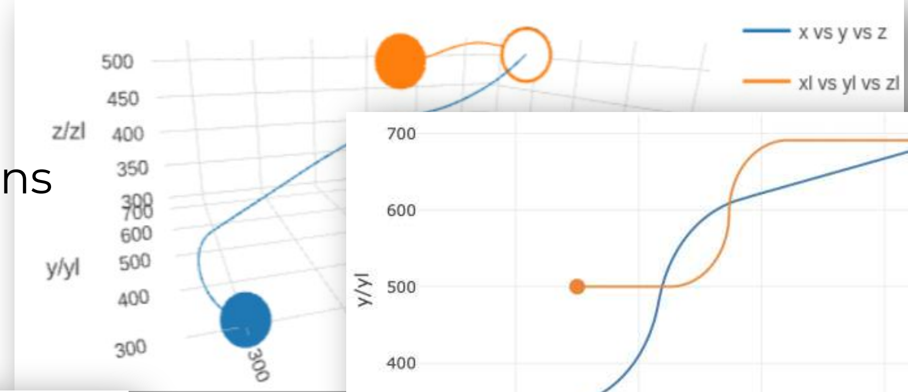
```
// Cruise control
p:=0; v:=2;
while true do {
  if v<=10
  then p'=v,v'=5  for 1
  else p'=v,v'=-2 for 1
```

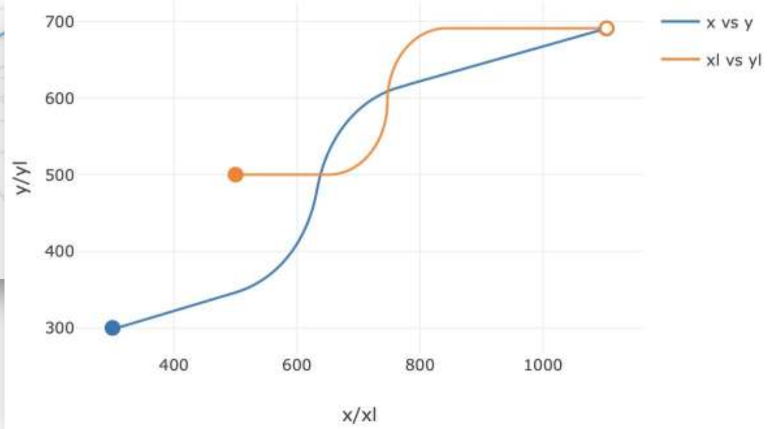
```
// Adaptive Cruise control
p:=0;    v:=0;   a:=5; b:=-2; // follower
pl:=50; vl:=10; al:=0;       // leader
period:=1;

while true do {                          new
  if ((p+v*period+ a/2*period^2 <
       pl+vl*period+al/2*period^2) &&
      (((v-vl+(a-al)*period)^2 -
        4*(p-pl+(v-vl)*period +
        (a-al)/2*period^2)*(b-al)/2) < 0))
  then p'=v,v'=a,pl'=vl,vl'=al for period;
  else p'=v,v'=b,pl'=vl,vl'=al for period;
```

```
// MORE:
// Automated braking system
// Pursuit games
//    (2D, 3D views)
// Electric RLC circuit
//    (many simulations, approx)
// ...
```

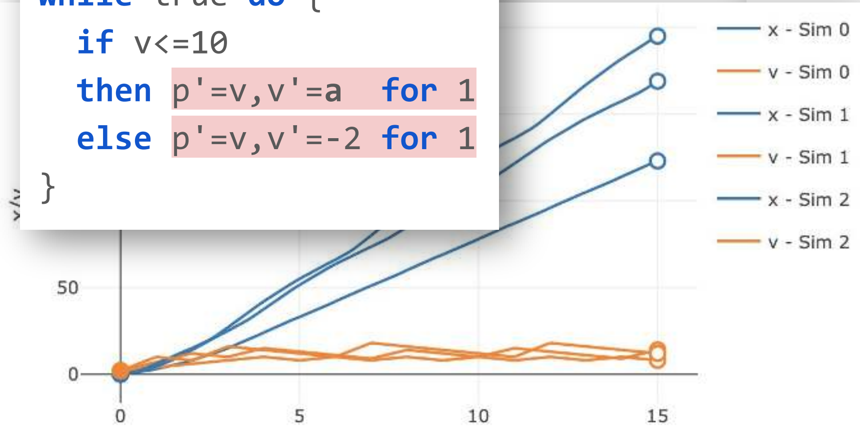
Recent improvements to Lince

- Customisable 2D/3D visualizations
- Overlay multiple simulations






```
// Cruise control
p:=0; v:=2; a:=[2,5,8];
while true do {
  if v<=10
  then p'=v,v'=a  for 1
  else p'=v,v'=-2 for 1
}
```



- More functions (and scenarios)
- Better error handling
- New approximated engine
- (Randomness)

Conclusions and challenges

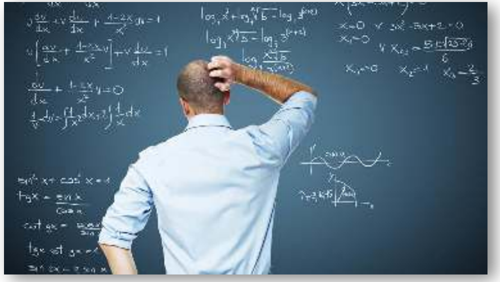
What to do with hybrid programs?

Generate software (e.g., controllers)

Predict physical behaviours

Simulate scenarios

Model checking
(verify properties of scenarios)



Theorem proving
(deductive reasoning to prove generic properties)

What to do with hybrid programs?

Generate software (e.g., controllers)

Predict physical behaviours

MATLAB SIMULINK

Simulate scenarios

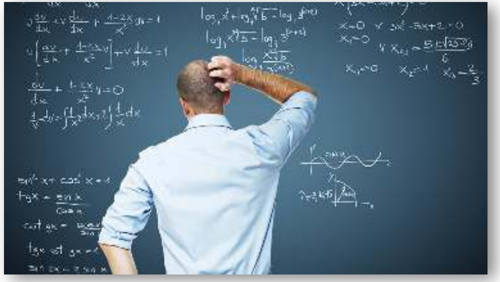
LINCE

Model checking
(verify properties of scenarios)

Hybrid ABS

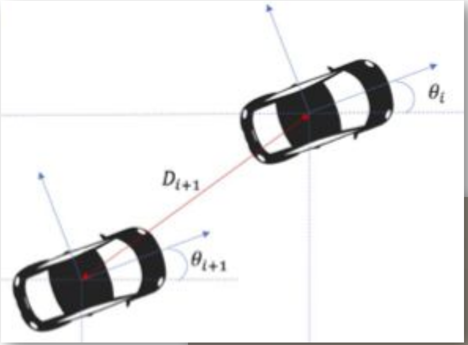
KeYmaera X

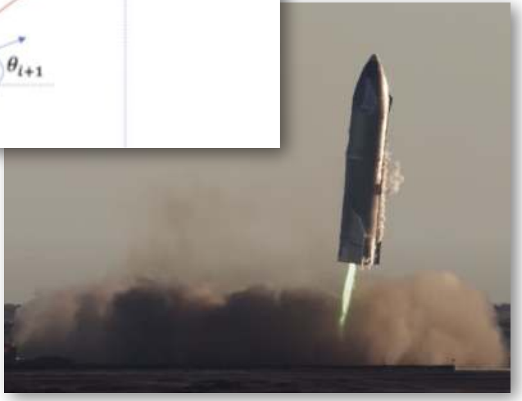
Theorem proving
(deductive reasoning to prove generic properties)



Challenge 1

Model more concrete scenario in Lince





```
p:=0; v:=2;
while x<=50 do {
  // extend the language if needed
  x(t) = 5/2*t^2+10*t+x0  for 1
}
// Find the need for new extensions
```

Challenge 2

Export program to another tool

```
// Cruise control
p:=0; v:=2;
while true do {
  if v<=10
  then p'=v,v'=5  for 1
  else p'=v,v'=-2 for 1
}
```

MATLAB SIMULINK

KeYmaeraX.org

Show some property?

Extend Lince?

```
// Cruise control
p:=0; v:=2;
while true do {...}
check [v>=10?...] p>5
```

Challenge 3

Import program from another tool

```
// Cruise control
p:=0; v:=2;
while true do {
  if v<=10
  then p'=v,v'=5  for 1
  else p'=v,v'=-2 for 1
}
```

MATLAB SIMULINK

KeYmaeraX.org

Useful subset?

Extend Lince?

```
// Cruise control
p:=0; v:=2;
while true do {...}
check [v>=10?...] p>5
```

Challenge 4

Checking properties: using logics for runtime verification

```
// Cruise control
p:=0; v:=2;
while true do {
  if v<=10
  then p'=v,v'=5  for 1
  else p'=v,v'=-2 for 1

  assert <formula>
}

monitor <formula>
```

Metric temporal logic

RMTL-∫3 (three-valued restricted metric temporal logic with durations)

(Interval) duration logic

Challenge 5

Running many simulations:
Statistical Analysis

```
// Cruise control
p:=0; v:=2; a:=Unif[2..8];
while true do {
  if v<=10
  then p'=v,v'=5  for Exp[0..1]
  else p'=v,v'=-2 for Exp[0..1]
}
check Prob(conf=95%, v<=10)
```

Running multiple times

Aim at statistical relevance

Control number of runs, range of a run, confidence, error margin, etc.

Challenge 6

Improve framework (technology)

JavaScript

Wrap up

```
// Cruise control
p:=0; v:=2;
while true do {
  if v<=10
  then p'=v,v'=5  for 1
  else p'=v,v'=-2 for 1
}
```

1. Syntax & semantics

2. Examples

3. Recent improvements

4. Challenges