First results on nonlinear hybrid reachability combining interval Taylor method and IBEX library

M. Maiga$^{1,2}$, N. Ramdani$^1$, and L. Travé-Massuyès$^2$

$^1$ Université d’Orléans, PRISME, 18020 Bourges, France
$^2$ CNRS, LAAS, 31400 Toulouse, France
(e-mail: mmaiga@laas.fr, nacim.ramdani@univ-orleans.fr, louise@laas.fr)

SWIM, Oldenburg, Germany
June 4-6, 2012
1 Introduction

2 Hybrid System

3 Interval Taylor Methods

4 Hybrid Transitions

5 IBEX library

6 Evaluation on Benchmarks
   - A simple illustrative exemple : 2 modes, continuous state dim=2
   - Benchmark 1
   - Benchmark 2
   - Benchmark 3

7 Conclusion
ANR-Project : MAGIC-SPS

Goal: To develop guaranteed methods and algorithms for integrity control and preventive monitoring of systems

Different work package:
* WP1: Modelling and identification of systems with bounded uncertainties;
* WP2: Identifiability and diagnosability of systems with bounded uncertainties;
* WP3: Preventive monitoring of continuous systems with bounded uncertainties;
* WP4: Preventive monitoring of hybrid systems with bounded uncertainties;
* WP5: Dissemination

Project duration = October 2012 to December 2014

Partners
Our work package : WP4

* Computing nonlinear hybrid reachability;
* State estimation of HDS;
* Feasibility of a fault prognosis for HDS
Outline

1. Introduction
2. Hybrid System
3. Interval Taylor Methods
4. Hybrid Transitions
5. IBEX library
6. Evaluation on Benchmarks
   - A simple illustrative exemple : 2 modes, continuous state dim=2
   - Benchmark 1
   - Benchmark 2
   - Benchmark 3
7. Conclusion
Hybrid System example: Bouncing ball

- Continuous dynamic (Free fall) → **Condition 1**
  \[
  x \geq 0 \\
  \dot{x} = v \\
  \dot{v} = -g
  \]

- Discrete dynamic (Bouncing) → **Condition 2**
  if \( x = 0 \) and \( v < 0 \); \( v := -cv \)
  * Velocity change direction
  * loss of velocity (deformation, friction)
  * \( 0 \leq c \leq 1 \)
Hybrid System example: Bouncing ball

**Condition 1 + Condition 2**

\[
x = x_0 \\
v = 0
\]

- **Initial conditions**
  - free fall:
    - \( x \geq 0 \)
    - \( \dot{x} = v \)
    - \( \dot{v} = -g \)

- **Location**
- **Invariant**
- **Flow**
- **Discrete transition**

\[
x_0 \in [5, 5.1], v_0 = 0 \\
c = 0.8, g = 9.8, t \in [0, 4.5]
\]
Hybrid Reachability Computation

Hybrid automaton (Alur, et al., 95)

\[ H = (Q, \mathcal{D}, \mathcal{P}, \Sigma, \mathcal{A}, \text{Inv}, \mathcal{F}), \]

\text{flow}(q): \quad \dot{x}(t) = f_q(x, p, t),

\text{Inv}(q): \quad \nu_q(x(t), p, t) < 0,

\( e : (q \rightarrow q') = (q, \text{guard}, \sigma, \rho, q') \),

\text{guard}(e): \quad \gamma_e(x(t), p, t) = 0,

\( t_0 \leq t \leq t_N, \quad x(t_0) \in X_0 \subseteq \mathbb{R}^n, \quad p \in \mathbb{P} \)
Set reachable in finite time

\[ \nu_0(.) < 0 \]

\[ \nu_1(.) < 0 \]
Outline

1. Introduction
2. Hybrid System
3. Interval Taylor Methods
4. Hybrid Transitions
5. IBEX library
6. Evaluation on Benchmarks
   - A simple illustrative exemple : 2 modes, continuous state dim=2
   - Benchmark 1
   - Benchmark 2
   - Benchmark 3
7. Conclusion
Guaranteed set integration with Taylor methods
(Moore, 66) (Eijgenraam, 81) (Lohner, 88) (Rihm, 94) (Berz, 98) (Nedialkov, 99)

\[ \dot{x}(t) = f(x, p, t), \quad t_0 \leq t \leq t_N, \; x(t_0) \in [x_0], \; p \in [p] \]

**Time grid** → \( t_0 < t_1 < t_2 < \cdots < t_N \)

![Diagram](image)

- **Analytical solution** for \([x](t), \; t \in [t_j, t_{j+1}]\)

\[ [x](t) = [x_j] + \sum_{i=1}^{k-1} (t - t_j)^i f[i]([x_j], [p]) + (t - t_j)^k f[k]([\tilde{x}_j], [p]) \]
Guaranteed set integration with Taylor methods

\[ \dot{x}(t) = f(x, p, t), \quad t_0 \leq t \leq t_N, \quad x(t_0) \in [x_0], \ p \in [p] \]

Mean-value approach

- mean value forms + matrice preconditioning + linear transforms
- \[ [x](t) \in \{ v(t) + A^a(t)r(t) \mid v(t) \in [v](t), \ r(t) \in [r](t) \}. \]

a. Several methods
Outline

1. Introduction
2. Hybrid System
3. Interval Taylor Methods
4. Hybrid Transitions
5. IBEX library
6. Evaluation on Benchmarks
   - A simple illustrative example: 2 modes, continuous state dim=2
   - Benchmark 1
   - Benchmark 2
   - Benchmark 3
7. Conclusion
Hybrid Transitions

Computing flow/guards intersection

Time grid → \( t_0 < t_1 < t_2 < \cdots < t_N \)

\[
\gamma_e(.) = 0
\]

Compute \([t^*, \bar{t}^*] \times [x_j^*]\)
Computing flow/guards intersection

Time grid $\rightarrow \quad t_0 < t_1 < t_2 < \cdots < t_N$

- $[x](t) = \text{Interval Taylor Series (ITS)}(t, [x_j], [\tilde{x}_j])$
- $\gamma([x](t)) = 0$

$\Rightarrow \gamma \circ \text{ITS}(t, x_j, [\tilde{x}_j]) \rightarrow \psi(t, x_j)$

To compute $[t^*, \bar{t}^*] \times [x_j^*] \Rightarrow \text{Solve CSP}^1 ([t_j, t_{j+1}] \times [x_j], \psi(., .) = 0)$

1. Handbook of Constraint Programming, Rossi et al., 2006

M. Maiga (PRISME & LAAS)
Hybrid Transitions

Computing flow/guards intersection

Time grid → \( t_0 < t_1 < t_2 < \cdots < t_N \)

- \([x](t) = \text{Interval Taylor Series (ITS)}(t, [x_j], [\tilde{x}_j])\)
- \(\gamma([x](t)) = 0\)

\(\Rightarrow \gamma \circ \text{ITS}(t, x_j, [\tilde{x}_j]) \rightarrow \psi(t, x_j)\)

To compute \([t^\ast, \bar{t}^\ast] \times [x_j^\ast] \Rightarrow \text{Solve CSP}^1 ([t_j, t_{j+1}] \times [x_j], \psi(., .) = 0)\)

How to solve this CSP?

1. Handbook of Constraint Programming, Rossi et al., 2006
Hybrid Reachability

Ramdani & Nedialkov, 2011

- **Interval Taylor methods**
  - Analytical expressions for the boundaries of the continuous flows,
  - Controlling Wrapping effect

- **Interval constraint propagation techniques**
  - Solve event detection/localization problems
  - Flow/sets intersection with ALIAS\(^a\) CSP solver.

\(^a\) [http://www-sop.inria.fr/coprin/logiciels/ALIAS/](http://www-sop.inria.fr/coprin/logiciels/ALIAS/)

This talk

- Use IBEX\(^a\)

- Test this new interface on benchmarks!

\(^a\) [http://www.ibex-lib.org/](http://www.ibex-lib.org/)
IBEX library (G. Chabert 2007)
Input IBEX = an interval vector \( XX \) (box) of dimension \( \text{dim}(XX) \)
\( XX=(XX(1);XX(2);\ldots;XX(\text{dim}(XX)) \)
Input IBEX = an interval vector $XX$ (box) of dimension $\text{dim}(XX)$

$XX=(XX(1);XX(2);........;XX(\text{dim}(XX))$

build a symbolic box of dimension $\text{dim}(XX)$

const Symbol & Xx=env.add_symbol("Xx",\text{dim}(XX));
Input IBEX = an interval vector \( XX \) (box) of dimension \( \text{dim}(XX) \)

\( XX = (XX(1);XX(2);\ldots;XX(\text{dim}(XX)) \)

build a symbolic box of dimension \( \text{dim}(XX) \)

\[
\text{const Symbol } \& \text{ Xx=env.add_symbol("Xx",dim(XX));}
\]

Initialize rechearch domain of each variable

\[
\text{space.box}(1)=\text{time};
\]

\[
\text{for(int } \text{ii=2;ii<=dim(XX);ii++}
\]

\[
\text{space.box(ii)=XX(ii)};
\]
- **Input IBEX** = an interval vector $XX$ (box) of dimension $\dim(XX)$
  $XX=(XX(1);XX(2);\ldots;XX(\dim(XX))$

- build a symbolic box of dimension $\dim(XX)$
  ```
  \text{const Symbol \& \ Xx=env.add_symbol("Xx","dim(XX));}
  ```

- Initialize research domain of each variable
  ```
  \text{space.box(1)=time;}
  \text{for(int ii=2;ii<=dim(XX);ii++)}
  \text{space.box(ii)=XX(ii);} 
  ```

- Solve CSP (Invariant and guard function) according to current location.
switch(mode)
{
    index_contraint[1]=env_.add_ctr(G(Xx[1],Xx[2],...,Xx[n])=0);
    index_contraint[2]=env.add_ctr(G(Xx[1],Xx[2],...,Xx[n])=0);
    index_contraint[m]=env.add_ctr(G(Xx[1],Xx[2],...,Xx[n])=0);

    vector<const Constraint*> vec_constraint;
    vec_constraint.push_back(&env.constraint(index_contraint[1....m])); vector of Constraint

    CSP csp(vec_constraint,space); Create a system of constraints (list of constraints)
    HC4 hc(csp); propagation with a system of constraints

    RoundRobin rr(csp.space, seuiB); Create a bisector with round-robin heuristic

    Paver paver(hc,rr, seuiP); a classical branch & bound algorithm

    paver.explore(); start research potential solutions
    paver.report(); report all solutions find after exploration
}
The output of Ibex:
Each solution of CSP is given by paver.box\((i,j)\)(i=number of contractor\(^2\), j\(^3\)=jth box solution) which is an interval vector

2. i=1
3. j=1...Nsol
For guard solving, \( \text{time} = [t_j, t_{j+1}] \rightarrow [\underline{t}^*, \overline{t}^*] \times [\mathcal{X}_j^*] \)
For guard solving, \[\text{time} = [t_j, t_{j+1}] \rightarrow [t^*, \bar{t}^*] \times [\mathcal{X}_j^*]\]

For invariant solving, \[\text{time} = [t_j] \rightarrow [\mathcal{X}_j^{inv}]\]
Outline

1. Introduction
2. Hybrid System
3. Interval Taylor Methods
4. Hybrid Transitions
5. IBEX library
6. Evaluation on Benchmarks
   - A simple illustrative exemple: 2 modes, continuous state dim=2
   - Benchmark 1
   - Benchmark 2
   - Benchmark 3
7. Conclusion
Outline

1. Introduction
2. Hybrid System
3. Interval Taylor Methods
4. Hybrid Transitions
5. IBEX library
6. Evaluation on Benchmarks
   - A simple illustrative exemple : 2 modes, continuous state dim=2
     - Benchmark 1
     - Benchmark 2
     - Benchmark 3
7. Conclusion
Example

$q = 1, 2 e = 1 \rightarrow 2$:

\[
\begin{align*}
\text{flow}(1) & : f_1(x_1, x_2) = (x_2, -px_2 - g \sin(x_1)) \\
\text{inv}(1) & : \nu_1(x_1, x_2) = x_2 - 1.5 \\
\text{flow}(2) & : f_2(x_1, x_2) = (x_2, -3px_2 - g \sin(x_1)) \\
\text{inv}(2) & : \nu_2(x_1, x_2) = -\nu_1(x_1, x_2) \\
\text{guard}(1) & : \gamma_1(x_1, x_2) = \nu_1(x_1, x_2) \\
\text{reset}(1) & : \rho_1(x_1, x_2) = (\alpha_1 x_1, \alpha_2 x_2)
\end{align*}
\]

avec $\alpha_1 = -1, \alpha_2 \in [-2.05, -2], g = 10, p \in [6, 6.3]$ et $x_0 \in [-0.9, -0.8] \times [3, 3.5]$. 
Small comparison about CPU times

\[
\begin{align*}
\text{flow}(1) & : \ f_1(x_1, x_2) = (x_2, -px_2 - g \sin(x_1)) \\
\text{inv}(1) & : \ \nu_1(x_1, x_2) = \cos(x_1) - x_2/10 - 0.7 \\
\text{flow}(2) & : \ f_2(x_1, x_2) = (x_2, -3px_2 - g \sin(x_1)) \\
\text{inv}(2) & : \ \nu_2(x_1, x_2) = -\nu_1(x_1, x_2) \\
\text{guard}(1) & : \ \gamma_1(x_1, x_2) = \nu_1(x_1, x_2) \\
\text{reset}(1) & : \ \rho_1(x_1, x_2) = (\alpha_1 x_1, \alpha_2 x_2)
\end{align*}
\]

(1)

with \( \alpha_1 = -1, \alpha_2 \in [-2.05, -2], g = 10, p \in [6, 6.3] \) and \( x_0 \in [-0.9, -0.8] \times [3, 3.5] \).

<table>
<thead>
<tr>
<th>ALIAS(^4)</th>
<th>26 s</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBEX(^5)</td>
<td>0.204 s</td>
</tr>
</tbody>
</table>

4. PIV 2GHz
5. Core i5 2.4GHz
Outline

1. Introduction
2. Hybrid System
3. Interval Taylor Methods
4. Hybrid Transitions
5. IBEX library
6. Evaluation on Benchmarks
   - A simple illustrative exemple : 2 modes, continuous state dim=2
   - Benchmark 1
   - Benchmark 2
   - Benchmark 3
7. Conclusion
Evaluation on Benchmarks: Glycemic Control in Diabetic Patients (Type 1 diabetes)

Bergman minimal model: \((G, I, X)\)

\[
\begin{align*}
\frac{dG}{dt} &= -p_1 G - X(G + G_B) + g(t) \\
\frac{dX}{dt} &= -p_2 X + p_3 I \\
\frac{dl}{dt} &= -n(l + l_b) + \frac{1}{V_l} i(t)
\end{align*}
\]

Initial conditions:

\[
G(0) \in [-2, 2] \quad X(0) = 0 \quad l(0) = \in [−0.1, 0.1]
\]

\[p_1 = 0.01, \quad p_2 = 0.025, \quad p_3 = 1.3 \times 10^{-5}, \quad V_l = 12, \quad n = 0.093, \quad G_B = 4.5, \quad l_b = 15.\]

The goal of this benchmark\(^6\) is to compute the reachable set over the time horizon \(t \in [0, 360]\).

---

6. Bench proposed by Xin Chen and Sriram Sankaranarayanan.
Evaluation on Benchmarks: Glycemic Control in Diabetic Patients (Type 1 diabetes)

Model 1

\[ i(t) = \begin{cases} 
\frac{1 + \frac{2G(t)}{9}}{3} & G(t) < 6 \\
\frac{50}{3} & G(t) \geq 6
\end{cases} \]

\[ g(t) = \begin{cases} 
\frac{t}{60} & t \leq 30 \\
\frac{120 - t}{180} & t \in [30, 120] \\
0 & t \geq 120
\end{cases} \]
Evaluation on Benchmarks: Glycemic Control in Diabetic Patients (Type 1 diabetes)
Evaluation on Benchmarks: Glycemic Control in Diabetic Patients (Type 1 diabetes)

CPU time = 1m11.500s core i5, 64 bits
Evaluation on Benchmarks: Glycemic Control in Diabetic Patients (Type 1 diabetes)

CPU time = 1m11.500s core i5, 64 bits
Evaluation on Benchmarks: Glycemic Control in Diabetic Patients (Type 1 diabetes)

CPU time = 1m11.500s core i5, 64 bits
Outline

1. Introduction
2. Hybrid System
3. Interval Taylor Methods
4. Hybrid Transitions
5. IBEX library
6. Evaluation on Benchmarks
   - A simple illustrative exemple: 2 modes, continuous state dim=2
   - Benchmark 1
   - Benchmark 2
   - Benchmark 3
7. Conclusion
**Evaluation on Benchmarks : Glycemic Control in Diabetic Patients (Type 1 diabetes)**

**Model 2**

\[
i(t) = \begin{cases} 
\frac{25}{3} \frac{G(t) \leq 4}{G(t)} & G(t) \in [4, 8] \\
\frac{125}{3} \frac{G(t) \geq 8}{G(t)} & G(t) \in [4, 8]
\end{cases}
\]

\[
g(t) = \begin{cases} 
\frac{t}{60} & t \leq 30 \\
\frac{120 - t}{180} & t \in [30, 120] \\
0 & t \geq 120
\end{cases}
\]
Evaluation on Benchmarks: Glycemic Control in Diabetic Patients (Type 1 diabetes)
Evaluation on Benchmarks: Glycemic Control in Diabetic Patients (Type 1 diabetes)

CPU time = 4m15.652s core i5, 64 bits

M. Maiga (PRISME & LAAS)
Evaluation on Benchmarks: Glycemic Control in Diabetic Patients (Type 1 diabetes)

CPU time = 4m15.652s core i5, 64 bits
Evaluation on Benchmarks: Glycemic Control in Diabetic Patients (Type 1 diabetes)

CPU time = 4m15.652s core i5, 64 bits
Outline

1. Introduction
2. Hybrid System
3. Interval Taylor Methods
4. Hybrid Transitions
5. IBEX library
6. Evaluation on Benchmarks
   - A simple illustrative exemple : 2 modes, continuous state dim=2
   - Benchmark 1
   - Benchmark 2
   - Benchmark 3
7. Conclusion
Evaluation on Benchmarks: Vehicle Model

\[
\begin{align*}
\frac{dx}{dt} &= \nu c_t; \quad \frac{dy}{dt} = \nu s_t; \quad \frac{dv}{dt} = u_1 \\
\frac{dc_t}{dt} &= \sigma \nu^2 s_t; \quad \frac{ds_t}{dt} = -\sigma \nu^2 c_t; \quad \frac{d\sigma}{dt} = u_2
\end{align*}
\]

\[x \in [1, 1.2] \quad y \in [1, 1.2] \quad \nu \in [0.8, 0.81] \quad s_t \in [0.6, 0.61] \quad c_t \in [0.7, 0.71] \quad \sigma = [0, 0.05]\]

The goal of this benchmark is to compute the reachable set over the time horizon \( t \in [0, 10] \).

7. Bench proposed by Xin Chen and Sriram Sankaranarayanan
Evaluation on Benchmarks : Vehicle Model

CPU time = 24.678 s core i5, 64 bits

\[
x \in [1, 1.2] \quad y \in [1, 1.2] \quad v \in [0.8, 0.81]
\]
\[
s_t \in [0.6, 0.61] \quad c_t \in [0.7, 0.71] \quad \sigma = 0.05
\]
Evaluation on Benchmarks : Vehicle Model

CPU time = 16.885 s core i5, 64 bits

\[
\begin{align*}
\frac{dx}{dt} &= v c_t; \quad \frac{dy}{dt} = v s_t; \quad \frac{dv}{dt} = u_1 \\
\frac{dc_t}{dt} &= \sigma v^2 s_t; \quad \frac{ds_t}{dt} = -\sigma v^2 c_t; \quad \frac{d\sigma}{dt} = u_2
\end{align*}
\]

\[x \in [1, 1.2] \quad y \in [1, 1.2] \quad v \in [0.8, 0.81] \]
\[s_t \in [0.6, 0.61] \quad c_t \in [0.7, 0.71] \quad \sigma = 0.05\]
Evaluation on Benchmarks : Vehicle Model

CPU time = 24.678 s core i5, 64 bits

\[
\begin{align*}
\frac{dx}{dt} &= vc_t; \quad \frac{dy}{dt} = vs_t; \quad \frac{dv}{dt} = u_1 \\
\frac{dct}{dt} &= \sigma v^2 s_t; \quad \frac{dst}{dt} = -\sigma v^2 c_t; \quad \frac{d\sigma}{dt} = u_2
\end{align*}
\]

\[x \in [1, 1.2] \quad y \in [1, 1.2] \quad v \in [0.8, 0.81] \]

\[s_t \in [0.6, 0.61] \quad c_t \in [0.7, 0.71] \quad \sigma \in [0.05, 0.05] \]
### Evaluation on Benchmarks: CPU Times

<table>
<thead>
<tr>
<th>Bench</th>
<th>Ss</th>
<th>NM</th>
<th>NT</th>
<th>NET</th>
<th>CPU times</th>
</tr>
</thead>
<tbody>
<tr>
<td>G M 1</td>
<td>3</td>
<td>6</td>
<td>10</td>
<td>4</td>
<td>1m11.500s</td>
</tr>
<tr>
<td>G M 2</td>
<td>3</td>
<td>9</td>
<td>16</td>
<td>6</td>
<td>4m15.652s</td>
</tr>
<tr>
<td>Vehicle model</td>
<td>6</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0m24.678s</td>
</tr>
</tbody>
</table>

- **Ss** = Size of system (continuous state vector dimension).
- **NM** = Number of Modes.
- **NT** = Number of Transitions.
- **NET** = Number of Enabled Transitions (with initial conditions given).
Outline

1 Introduction
2 Hybrid System
3 Interval Taylor Methods
4 Hybrid Transitions
5 IBEX library
6 Evaluation on Benchmarks
   - A simple illustrative exemple : 2 modes, continuous state dim=2
   - Benchmark 1
   - Benchmark 2
   - Benchmark 3
7 Conclusion
Concluding remarks

Conclusion

- Analytical expression for the continuous flows
- Interval constraint programming for solving flow/guards intersection
- CSP solving IBEX
Concluding remarks

Conclusion

- Analytical expression for the continuous flows
- Interval constraint programming for solving flow/guards intersection
- CSP solving IBEX

→ Controlling the number of the box in the list
Concluding remarks

Conclusion

- Analytical expression for the continuous flows
- Interval constraint programming for solving flow/guards intersection
- CSP solving IBEX

→ Controlling the number of the box in the list

→ Merging boxes without over-approximation (A Rauh, et al., 2006) (Benazera and Louise, 2009)
Conclusion

- Analytical expression for the continuous flows
- Interval constraint programming for solving flow/guards intersection
- CSP solving IBEX

→ Controlling the number of the box in the list

→ Merging boxes without over-approximation (A Rauh, et al., 2006) (Benazera and Louise, 2009)

→ Use VNODE-LP (N. Nedialkov, 2010)