Optimal switching instants for the Control of Hybrid Systems

J. Alexandre dit Sandretto, A. Chapoutot, and O. Mullier

Generic case

$$\left[\begin{array}{lll} \max\limits_{u(.)} & J(u(.)) = \int_0^T h(x(t), u(t))dt + g(x(T)) & \text{(cost function)} \\ \text{s. t.} & \dot{x} = f(x(t), u(t)),\ 0 < t \leqslant T & \text{(dynamical constraint)} \\ & x(0) = x_0,\ h(x(t)) \in \mathcal{H},\ 0 < t \leqslant T & \text{(boundary conditions)} \\ & u(t) \in \mathcal{U},\ \forall t & \text{(bounded control)} \end{array}\right]$$

- The dynamical constraint coupled with the boundary conditions is an IVP-ODE depending on a control function $u(t)$;
- the cost function is a pay-off function where $x(t)$ is the solution for the dynamical constraint, h is the running cost and g is the terminal cost.

An IVP-ODE is defined by

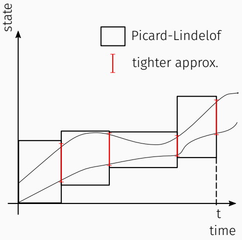
$$\begin{cases} \dot{x} = f(x) \\ x(0) \in \mathcal{X}_0 \subseteq \mathbb{R}^n, \ t \in [0, t_{\text{end}}] \end{cases} .$$

The goal is to compute $x(t; \mathcal{X}_0) = \{x(t; x_0) \mid x_0 \in \mathcal{X}_0\}$.

Phase 1 a priori enclosure $[\tilde{x}_i]$ of

$$\{x(t_k; x_i) \mid t_k \in [t_i, t_{i+1}], x_i \in [x_i]\}$$

Phase 2 tight enclosure of $[x_{i+1}]$ at time t_{i+1}.



Dynibex

- C++ library using ibex (constraint processing over real numbers);
- proof of existence and uniqueness of solution for ODEs and DAEs;
- combined with contractors (HC4), easy to use in branching algorithms;
- verification of temporal constraints.

Example of temporal constraints

- Stayed in $\mathcal{A}$ until $\tilde{t} < t_{\text{end}}$:

$$\forall t \in \left[0, \tilde{t}\right], \ \{y(t; y_0) \mid y_0 \in [y_0]\} \subseteq \text{int}(\mathcal{A})$$

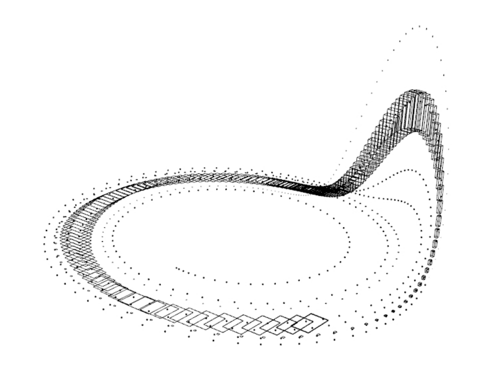
- Included in $\mathcal{A}$ inside $[0, t_{\text{end}}]$:

$$\exists t \in [0, t_{\text{end}}], \ \{y(t; y_0) \mid y_0 \in [y_0]\} \subseteq \text{int}(\mathcal{A}).$$

Example

System of Rossler: Initial states: $(0; -10.3; 0.03)$, some parameters: $a = 0.2, b = 0.2, c = 5.7$

$$\begin{cases} \dot{x} = -y - z \\ \dot{y} = x + ay \\ \dot{z} = b + z(x - c) \end{cases}$$



Computation of the cost function

For $\mathcal{U} \subseteq [u] \in$, $\dot{x} \in f(x(t), [u])$ and

$$\begin{aligned} J(u(.)) &= \int_0^T h(x(t), u(t))dt + g(x(T)) \\ &= \sum_{i=0}^{n} \int_{t_i}^{t_{i+1}} h(x(t), u(t))dt + g(x(T)) \\ &\in \sum_{i=0}^{n} (t_{i+1} - t_i) h([\tilde{x}_i], [u]) + g([x_T]) \end{aligned}$$

$$\left[\begin{array}{lll} \max\limits_{u(.)} & J(u(.)) = \int_0^T h(x(t),u(t))dt + g(x(T)) & \text{(cost function)} \\ \text{s. t.} & \dot{x} = f(x(t),u(t)),\ 0 < t \leqslant T & \text{(dynamical constraint)} \\ & x(0) = x_0,\ h(x(t)) \in \mathcal{H},\ 0 < t \leqslant T & \text{(boundary conditions)} \\ & u(t) \in \mathcal{U},\ \forall t & \text{(bounded control)} \end{array}\right]$$

Restriction

- Particular kind of dynamics:
  - the integral is provided by the dynamical constraints,
  - the set of possible control $u(t)$ is known and is discrete;
- the cost function is monotonic;
- the boundary conditions only occurs at a specific time τ.

n-mode hybrid system

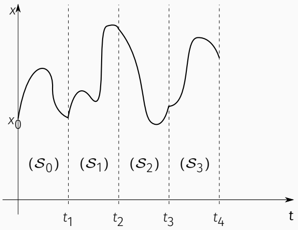
$$(\mathcal{S}_i)\begin{cases}\dot{x} = f_i(x) \\ x(t_i) = x_i\end{cases} \quad \text{in the time interval } [t_i, t_{i+1}]$$

- $f_i : \mathbb{R}^m \to \mathbb{R}^m$;
- $x_i \in \mathbb{R}^m$ is the initial condition for all modes $0 \leqslant i \leqslant n-1$.

A sequence $\{(\mathcal{S}_1), \ldots, (\mathcal{S}_k)\}$ corresponds to the switching of control law.

- x_0 is fixed;
- x_i is taken as the solution at time t_i of $(\mathcal{S}_{i-1})$.

$\Rightarrow$ not necessarily continuously differentiable.



Our problem can be modeled using the following optimization problem

$$\left[\begin{array}{ccc} \max\limits_{t_1,\dots,t_{n-1}} & g(x(\tau)) & \text{(cost function)} \\ \text{s. t.} & \forall 0 \leqslant i \leqslant n-1, (\mathcal{S}_i) & \text{(dynamical constraint)} \\ & h(x(\tau)) > 0 & \text{(reachability constraint)} \\ & \tau \in [t_{n-1}, t_n] & \end{array}\right]$$

with

- the decision variables $t_1, \dots, t_{n-1} \in \mathbb{R}^n_+$ the search space for the different times;
- the cost function $g : \mathbb{R}^m \to \mathbb{R}$ on the state variable at given time $\tau \in [t_{n-1}, t_n]$;
- some constraints defined by the dynamical systems $(\mathcal{S}_i)$ and the times t_i;
- a reachability constraint using $h : \mathbb{R}^m \to \mathbb{R}$.

Example: Goddard's Rocket

Model of the ascent of a rocket in the atmosphere:

$$\left[\begin{array}{ll} \max & m(T) \\ \text{s.t.} & \dot{r} = v \\ & \dot{v} = \frac{u - Av\exp(k(1-r))}{m} - \frac{1}{v^2} \\ & \dot{m} = -bu \\ & u(.) \in [0,1] \\ & r(0) = 1, v(0) = 0, m(0) = 1 \\ & r(T) \geqslant \mathcal{R}_T \end{array}\right]$$

with the parameters

- $b = 2$,
- $T_{\max} = 0.2$,
- $A = 310$,
- $k = 500$,
- $r_0 = 1$, $v_0 = 0$, $m_0 = 1$,
- $\mathcal{R}_T = 1.01$.

According to the time t:

$$u(t) = \begin{cases} 3.5 & \text{for } t \in [0, t_1] \ (\mathcal{S}_0) \\ 3.5\tanh(1+t) & \text{for } t \in [t_1, t_2] \ (\mathcal{S}_1) \\ 0 & \text{for } t \in [t_2, T] \ (\mathcal{S}_2) \end{cases}$$

Algorithm 1: simu(t_1, t_2, max) – simulates the system from 0 to T.

Input: time t_1, t_2 to switch dynamics; current maximum mass max

Output: the mass m or 0 if simulation will not produce a better solution

$([r_{t_1}], [v_{t_1}], [m_{t_1}]) \leftarrow$ simulation of x_0 using $(\mathcal{S}_0)$ from 0 to t_1;

if $\overline{m}_{t_1} \leq max$ **then**

  return 0;

$([r_{t_2}], [v_{t_2}], [m_{t_2}]) \leftarrow$ simulation using $(\mathcal{S}_1)$ from t_1 to t_2;

if $\overline{m}_{t_2} \leq max$ **then**

  return 0;

$([r_T], [v_T], [m_T]) \leftarrow$ simulation using $(\mathcal{S}_2)$ from t_2 to T;

if $\underline{r_T} \geqslant \mathcal{R}_T$ **then**

  return $[m_T]$;

else

  return 0;

Algorithm 2: finds the optimal switching times

```
Input: set of dynamics {(S_0), (S_1), (S_2)}
Output: switching times t_{1,max} and t_{2,max}
max ← 0;
for t_1 ← 0 to T − ε do
    for t_2 ← t_1 + ε to T do
        [m_, m̄] ← simu(t_1, t_2);
        if m_ > max then
            m_ ← max;
            t_{1,max} ← t_1;
            t_{2,max} ← t_2;
        else
            break;          // Due to monotonicity of the cost
             function
return t_{1,max} and t_{2,max};
```

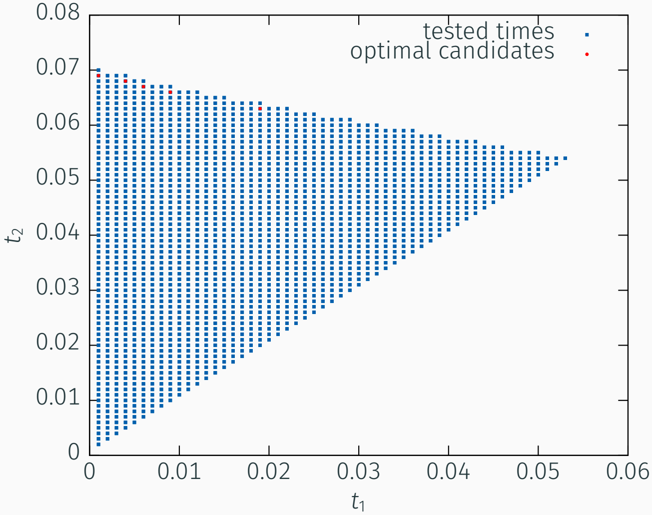


Figure 1: Mesh for t_2 w.r.t. t_1

Results

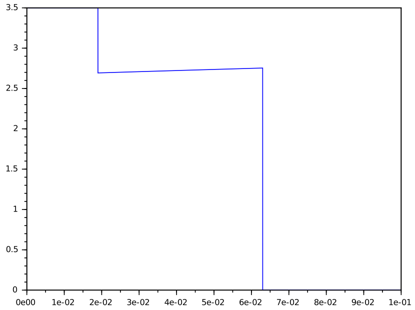


Figure 2: Optimal controller

Conclusion

- Promising results on the computation of optimal switching mode;
- easy-to-use tool development;
- benefits shown on an example.

Perspectives

To release restrictions to handle the problem in its generality

Thank you

Robert H Goddard.
A method of reaching extreme altitudes.
Nature, 105:809–811, 1920.

Knut Graichen and Nicolas Petit.
Solving the goddard problem with thrust and dynamic pressure constraints using saturation functions.
IFAC Proceedings Volumes, 41(2):14301–14306, 2008.