Set-based methods for property verification and control synthesis

Seminar in the Milan's office

Julien Alexandre dit Sandretto

Département U2IS

ENSTA Paris - IP Paris

Ecole Doctorale de Mathématiques Hadamard

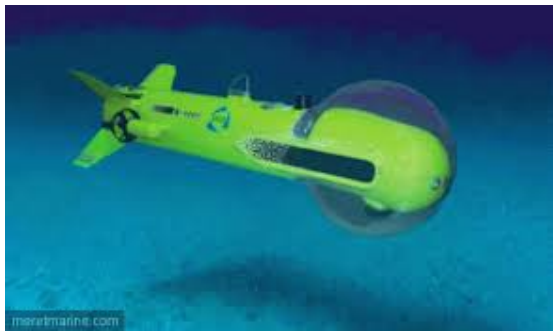
ENSTA

IP PARIS

Context

Cyber-physical systems

- Critical;
- Require safety;
- Need some planning and control solutions;
- Tainted with uncertainties (measures, inputs, etc) and approximations (modelling, algorithms).



Introduction

The chosen approach

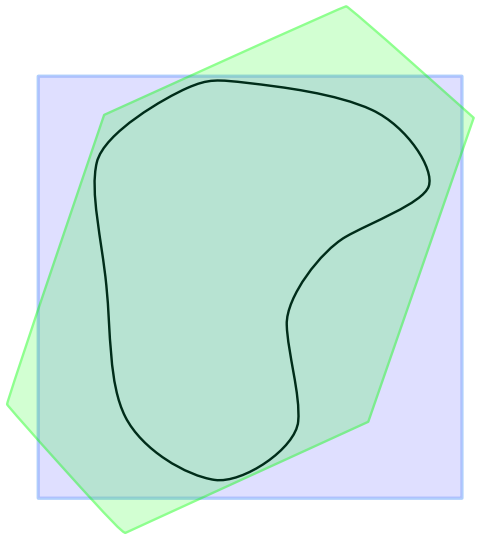
For uncertainties, set-based methods can help:

- Handle several values for measures and parameters;
- Provide some guarantees in computation with a correct-by-design abstraction.

For examples, Interval Analysis and Affine Arithmetic help to compute reliably abstractions of sets enclosed by boxes or zonotopes.

Helpful in Robotics

- Control synthesis,
- Motion planning,
- Kinematics,
- Parameter identification, etc.



Many publications: Jaulin, Rauh, Merlet, etc.

Introduction

Dynamical systems

Mainly interested by applications in robotics:
A robot is a dynamical system !

Model based approaches

Need a model for dynamical systems: differential equations are suitable.

Sets and Differential Equations

Differential Equations

Ordinary Differential Equation (ODE)

$$\dot{y}(t) = \frac{\partial y(t)}{\partial t} = f(y(t))$$

Solution is a function $y(t) : \mathbb{R} \mapsto \mathbb{R}^n$.

Initial Value Problem (IVP)

From an initial value $y(0) = y_0$, solution at time h can be computed by

$$y(h) = y_0 + \int_0^h f(y(s))ds$$

Hypotheses: f is Lipschitz and the associated vector field is sufficiently smooth.

In general, the equation above cannot be explicitly computed due to its implicit nature.

Differential Equations with sets

For a parametric IVP

$$\dot{y}(t) = f(y(t), p), \quad y(0) = y_0$$

Parameters

- Constant parameters not perfectly known but bounded;
- To consider several values at once, etc.

$$p \in \mathcal{P}$$

Initial value

- Uncertainty in measures;
- To estimate different trajectories, etc.

$$y(0) \in \mathcal{Y}_0$$

$$\dot{y}(t) \in f(y(t), \mathcal{P}), \quad y(0) \in \mathcal{Y}_0$$

Reachability

Compute the set of states at a given time $t > 0$.

Many names

Validated simulation, validated numerical integration, reachability analysis, guaranteed integration, etc..

Our goal

Compute a tube enclosing all the solutions:

$$\mathcal{Y} \supset \{y(t), t \in [0, T], \forall y(0) \in \mathcal{Y}_0, \forall p \in \mathcal{P}\}$$

Main tool: Interval analysis

$[x] = [\underline{x}, \overline{x}]$ stands for the set of reals x s.t. $\underline{x} \leq x \leq \overline{x}$

Arithmetic

Extension of operators $(+, -, *, /, sin, cos, ...)$, e.g. $[-1, 1] + [1, 3] = [0, 4]$
Rounding error handled $(1/3 \in 0.33333333[3, 4])$

Extension of function

$$[f]([x]) \supset f([x]) = \{f(y) | y \in [x]\}$$

Interval Integral

Rectangle rule: $\int_{[x]} f(x')dx' \in [f]([x]).w([x])$

Main tool: Interval analysis

$[x] = [\underline{x}, \overline{x}]$ stands for the set of reals x s.t. $\underline{x} \leq x \leq \overline{x}$

Arithmetic

Extension of operators $(+, -, *, /, sin, cos, ...)$, e.g. $[-1, 1] + [1, 3] = [0, 4]$
Rounding error handled $(1/3 \in 0.33333333[3, 4])$

Extension of function

$$[f]([x]) \supset f([x]) = \{f(y) | y \in [x]\}$$

Interval Integral

Rectangle rule: $\int_{[x]} f(x')dx' \in [f]([x]).w([x])$

Main tool: Interval analysis

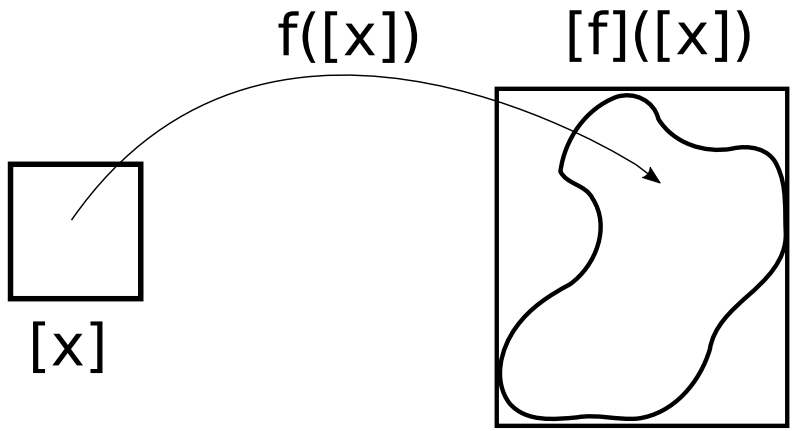
$[x] = [\underline{x}, \overline{x}]$ stands for the set of reals x s.t. $\underline{x} \leq x \leq \overline{x}$

Arithmetic

Extension of operators $(+, -, *, /, sin, cos, ...)$, e.g. $[-1, 1] + [1, 3] = [0, 4]$
Rounding error handled $(1/3 \in 0.33333333[3, 4])$

Extension of function

$$[f]([x]) \supset f([x]) = \{f(y) | y \in [x]\}$$



Interval Integral

Rectangle rule: $\int_{[x]} f(x')dx' \in [f]([x]).w([x])$

Main tool: Interval analysis

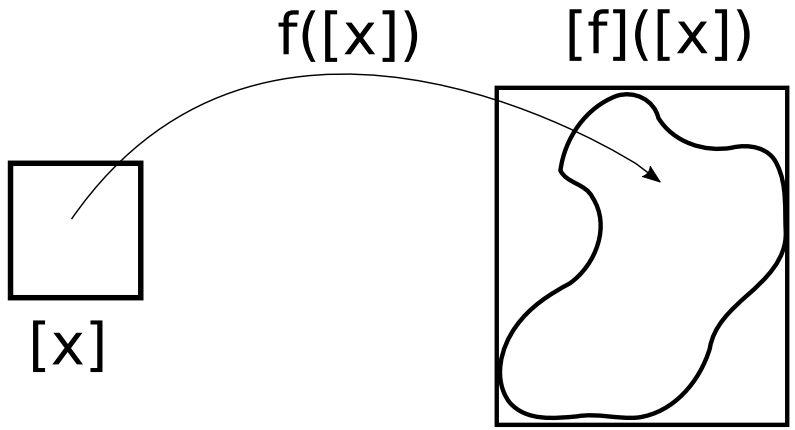
$[x] = [\underline{x}, \overline{x}]$ stands for the set of reals x s.t. $\underline{x} \leq x \leq \overline{x}$

Arithmetic

Extension of operators $(+, -, *, /, sin, cos, ...)$, e.g. $[-1, 1] + [1, 3] = [0, 4]$
Rounding error handled $(1/3 \in 0.33333333[3, 4])$

Extension of function

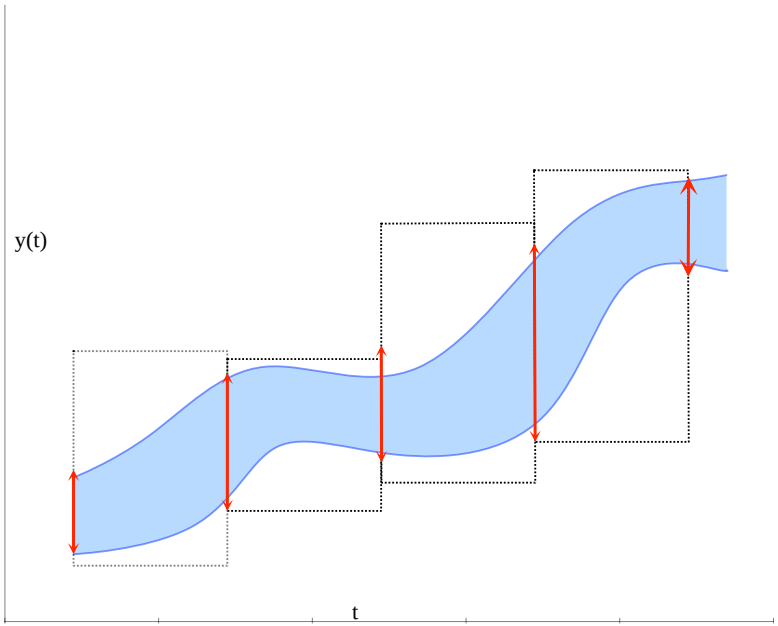
$$[f]([x]) \supset f([x]) = \{f(y) | y \in [x]\}$$



Interval Integral

Rectangle rule: $\int_{[x]} f(x')dx' \in [f]([x]).w([x])$

Validated simulation



$y(t)$ w.r.t. time. In blue, the theoretical set of trajectories enclosed by a tube.

Solution is a list of boxes:

- $t_i \times [y_i]$: validated integration scheme (mainly Taylor or Runge-Kutta)
- $[t_i, t_{i+1}] \times [\tilde{y}_i]$: Picard-Lindelöf algorithm (also called Cauchy-Lipschitz)

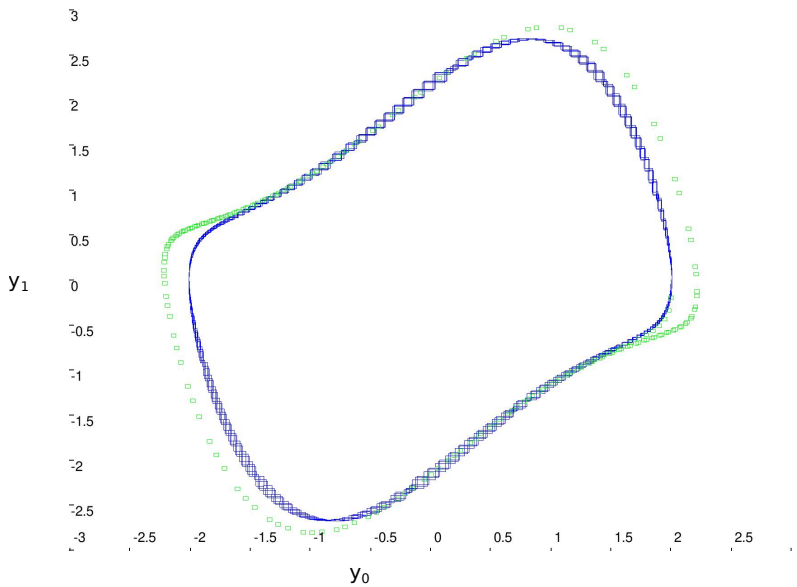
Set-based B-series and Runge-Kutta

Runge-Kutta and B-series

An example: Van der Pol

Initial states: $y(0)$
Parameter: μ

$$\dot{y} = \begin{cases} y_1 \\ \mu(1 - y_0^2)y_1 - y_0 \end{cases}$$



From the first scheme: Euler (1792)

$$y(k+1) = \begin{cases} y_0(k) + hy_1(k) \\ y_1(k) + h\mu(1 - y_0(k)^2)y_1(k) - y_0(k) \end{cases}$$

Runge-Kutta and B-series

To a family of schemes

$\approx$ 1900: Heun, Kutta, RK4, etc.
Then implicit RK, DIRK, SDIRK, etc.

A unified description: Butcher Tableau

Example of Heun's method:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$\begin{array}{c|ccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ \vdots & \vdots & \vdots & & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

$$\begin{aligned} \mathbf{k}_1 &= f(t_n, \mathbf{y}_n) \ , \\ \mathbf{k}_2 &= f(t_n + 1h, \mathbf{y}_n + h1\mathbf{k}_1) \\ \mathbf{y}_{n+1} &= \mathbf{y}_n + h\left(\frac{1}{2}\mathbf{k}_1 + \frac{1}{2}\mathbf{k}_2\right) \end{aligned}$$

To go further: introduction of rooted trees

Rooted tree = graph with a distinguished node, the root, in which every other node is connected to the root by a unique path.

Operations

- τ breaks up into rooted trees $\tau_1, \tau_2, ...$
- $\tau = [\tau_1, \tau_2, ...]$ by joining the roots of the trees to a new common root.

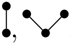
Order, symmetry, factorial

- Number of nodes in a tree is denoted by $|\tau|$ (order).
- Number of equivalence classes of heap-orderings $\alpha(\tau) = \frac{|\tau|!}{\tau! |S_\tau|}$
- S_τ denotes the symmetry group of τ
- Tree factorial $[\tau_1, \ldots, \tau_n]! = |[\tau_1, \ldots, \tau_n]| \cdot \tau_1! \cdots \tau_n!$, $\bullet! = 1$

Butcher's theory

Elementary differentials:

$$F(\bullet)(y) = f(y), \quad F(\tau)(y) = f^k(y)(F(\tau_1)(y), \dots, F(\tau_k)(y)),$$

with $\tau, \tau_1, \dots, \tau_k \in \mathcal{T}$, $\mathcal{T} = \{\bullet,$  $, \dots\}$ the set of trees and $\tau = [\tau_1, \dots, \tau_k]$ the result of k trees grafting.

From the chain rule:
$\dot{y} = f(y)$, $\ddot{y} = (f'f)(y)$, $\dddot{y} = (f''(f,f))(y) + (f'f'f)(y) \quad \Rightarrow$ tree structure

Butcher's theory

Theorem 1 (Butcher, 1963)

The qth derivative of the **exact solution** is given by

$$\mathbf{y}^{(q)} = \sum_{r(\tau)=q} \alpha(\tau) F(\tau)(\mathbf{y}_0) \quad \text{with} \quad \begin{array}{l} r(\tau) \text{ the order of the rooted tree } \tau \\ \alpha(\tau) \text{ a positive integer} \\ F(\tau)(.) \text{ elementary differential for } \tau \end{array}$$

We can do the same for the numerical solution:

Theorem 2 (Butcher, 1963)

The qth derivative of the **numerical solution** is given by

$$\mathbf{y}_1^{(q)} = \sum_{r(\tau)=q} \gamma(\tau)\phi(\tau)\alpha(\tau) F(\tau)(\mathbf{y}_0) \quad \text{with} \quad \begin{array}{l} \gamma(\tau) \text{ a positive integer} \\ \phi(\tau) \text{ depending on a Butcher tableau} \end{array}$$

Theorem 3, order condition (Butcher, 1963)

A Runge-Kutta method has order p iff $\phi(\tau) = \frac{1}{\gamma(\tau)} \quad \forall \tau, r(\tau) \leqslant p$

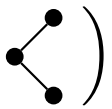
Truncation error of Runge-Kutta schemes

From Theorem 1 and Theorem 2, if a Runge-Kutta has order p then

$$\mathbf{y}(t_1;\mathbf{y}_0)-\mathbf{y}_1 = \frac{h^{p+1}}{(p+1)!}\sum_{r(\tau)=p+1}\alpha(\tau)\big[1-\gamma(\tau)\phi(\tau)\big]F(\tau)(\mathbf{y}(\xi)), \quad \xi\in[t_0,t_1]$$

- $\alpha(\tau)$, $\gamma(\tau)$ and $\phi(\tau)$ function of the coefficients of the RK method

Example

$\phi\Big($  $\Big)$ is associated to $\displaystyle\sum_{i,j=1}^{s} b_i a_{ij} c_j$ with $c_j = \displaystyle\sum_{k=1}^{s} a_{jk}$

Truncation error of Runge-Kutta schemes

This error can be bounded:

- for each method, the Butcher tableau and the order are available,
- ξ enclosed by $[t_0, t_1]$ and $\mathbf{y}([t_0, t_1])$ obtained by Picard-Lindelöf operator

Picard-Lindelöf operator

Formal solution of ODE: $\mathbf{y}_{n+1} = \mathbf{y}_n + \int_0^h f(s)ds$
Following the rectangle rule, based on Brouwer's theorem, the Picard-Lindelöf operator is defined such that:

$$P([\mathbf{x}]) = \mathbf{y_n} + [0, h][f]([\mathbf{x}]), \text{ with } [\mathbf{x}] \text{ a candidate}$$

- If $P([\mathbf{x}]) \subset Int([\mathbf{x}])$, then ODE admits one and only one solution, and $\mathbf{y}([t_0, t_1]) \subset [\mathbf{x}]$

In general, the rectangle rule is replaced by higher order Taylor series [1]

[1] Nedialkov et al., "Validated solutions of initial value problems for ordinary differential equations", Appl. Math. and Comp., 1999

Validated Runge-Kutta methods

A validated algorithm

$$[\mathbf{y}_{\ell+1}] = [\Phi]\,(h, [\mathbf{y}_\ell]) + \text{Error of } \Phi \quad .$$

$[\Phi]$

- Interval evaluation of explicit RKs
- Contraction based approach for implicit RKs

Error of $[\Phi]$

From Theorem 1 and Theorem 2 (Butcher 1963), if a Runge-Kutta has order p then

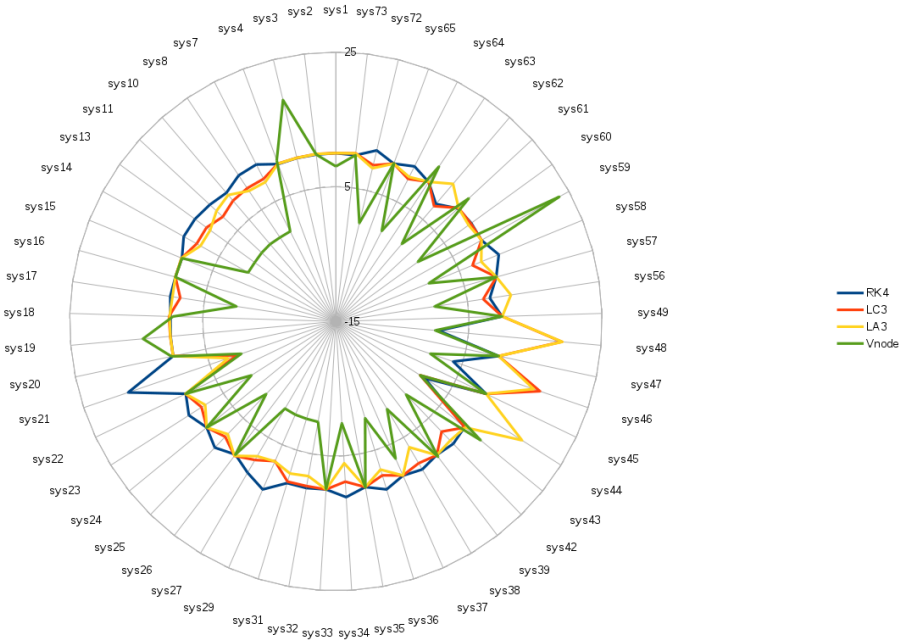
$$\text{Error of } \Phi = \frac{h^{p+1}}{(p+1)!} \sum_{r(\tau)=p+1} \alpha(\tau)\big[1 - \gamma(\tau)\phi(\tau)\big] F(\tau)([\mathbf{x}])$$

Validated RK: Results

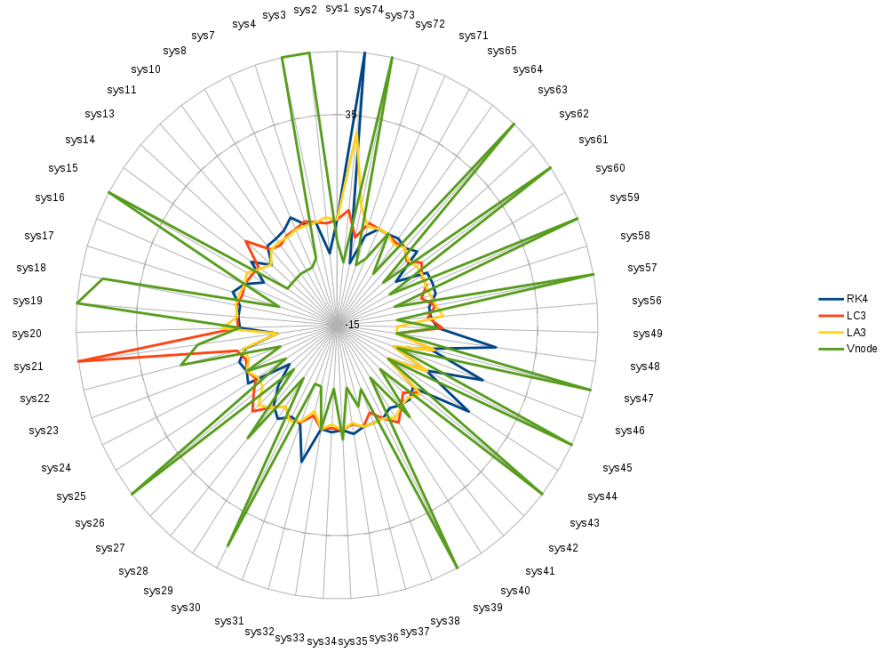
Implemented in DynIbex: Runge-Kutta explicit and implicit methods (≈ 10), affine arithmetic, Picard operator, truncation error with symbolic and AD approaches.

Good competitor

A report of 112 pages to compare DynIbex to the (available) state of the art (2015). Dimension > 50. Stiff. Chaos. "Large" initial conditions.



After 1s



After 10s

Validated RK: Results

Computation of truncation error

Too expensive for high order schemes...Vnode at order 20 is often faster than 4th order RK !

Some improvements in the code followed, but performances were sufficient for robotics!

B-series

About theoretical aspect, one formalism: the B-series (Cayley, Merson, Butcher, Hairer, Wanner, etc)
A B-series is a formal series of the form

$$B(a, hf, y) = a(\emptyset)y + \sum_{\tau \in \mathcal{T}} \frac{a(\tau)h^{|\tau|}}{\sigma(\tau)} F(\tau)(y),$$

with $F(.)(y)$ the elementary differentials as described previously.

$$B(a, hf, y) = a(\emptyset)y + ha(\bullet)f(y) + h^2 a(\overset{\bullet}{|}\bullet)(f'f)(y) + \frac{h^3}{2} a(\vee)(f''(f,f))(y) + \dots$$

Lemma (B-series and IVP)

The exact solution of an IVP is given by a B-series

$$y(h) = B(E, hf, y_0), \text{ with } E(\tau) = 1/\tau!$$

B-series

A Runge-Kutta scheme approximates the solution of an ODE with a B-series:

$$y(h) \approx y_1 = B(\phi(A, b), hf, y_0),$$

if the Runge-Kutta scheme is defined as follows

$$y_1 = y_0 + b^T hf(Y), \quad Y = ey_0 + Ahf(Y),$$

Order p if

$$E(\tau) = \phi(A, b)(\tau), \quad \forall \tau \in \mathcal{T}, |\tau| \leq p,$$

which means that $B(E, hf, y_0) - B(\phi(A, b), hf, y_0) \in \mathcal{O}(h^{p+1})$.

Remark: if $E(\tau) = 1/\tau!, \forall \tau \in \mathcal{T}, |\tau| \leq p$, truncated Taylor series!

Set-based B-series

Validated RK

$$y(h) = B(E, hf, y_0) \in B(\phi(A, b), hf, y_0) + B(\psi(A, b), hf, \mathcal{Y}^*)$$

with $\mathcal{Y}^* \ni y(t), \forall t \in [0, h]$, and

$$\psi(A, b)(\tau) = 1/\tau! - \phi(A, b)(\tau), \forall \tau \in \mathcal{T} \text{ if } |\tau| = p + 1,$$

Picard operator

If it exists a set $\mathcal{R} \subset \mathbb{R}^n$ such that

$$B(E, [0, h]f, \mathcal{R}) \subset \mathcal{R}$$

Brouwer theorem (also proposed/studied by Picard, Poincaré and Hadamard) and Picard-Lindelöf theorem: the initial value problem has a unique solution in the set $\mathcal{R}$ (then $\mathcal{Y}^* = \mathcal{R}$).

Set abstraction

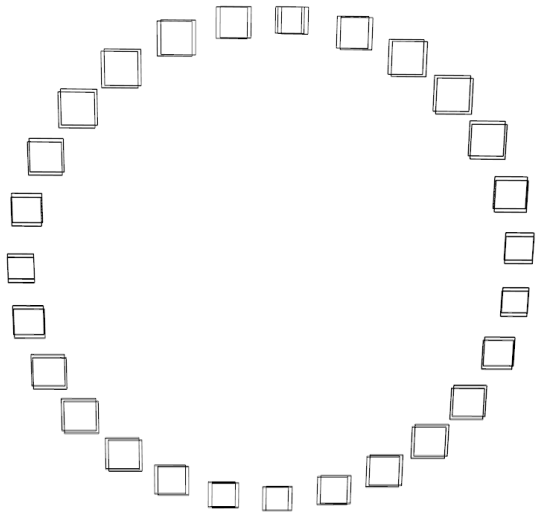
- Boxes with interval arithmetic:
  Fast but wrapping effect and pessimism
- Zonotopes with affine arithmetic:
  Expensive but “robust” to wrapping effect
- Polytopes with zonotopes and constraint programming:
  Very expensive but...

Validated RK: Results

Circle 100s with RK4

Initial states: $y(0) = ([0, 0.1]; [0.95, 1.05])$

The differential system: $\dot{y} = \begin{cases} -y_1 \\ y_0 \end{cases}$



Wrapping effect must be counteracted $\Rightarrow$ affine arithmetic

With zonotopes

Affine arithmetic extends numerical operations

- $\hat{x} = \alpha_0 + \sum_{i=1}^{n} \alpha_i \varepsilon_i$
- $\hat{y} = \beta_0 + \sum_{i=1}^{n} \beta_i \varepsilon_i$

with $\varepsilon_i \in [-1, 1]$, then

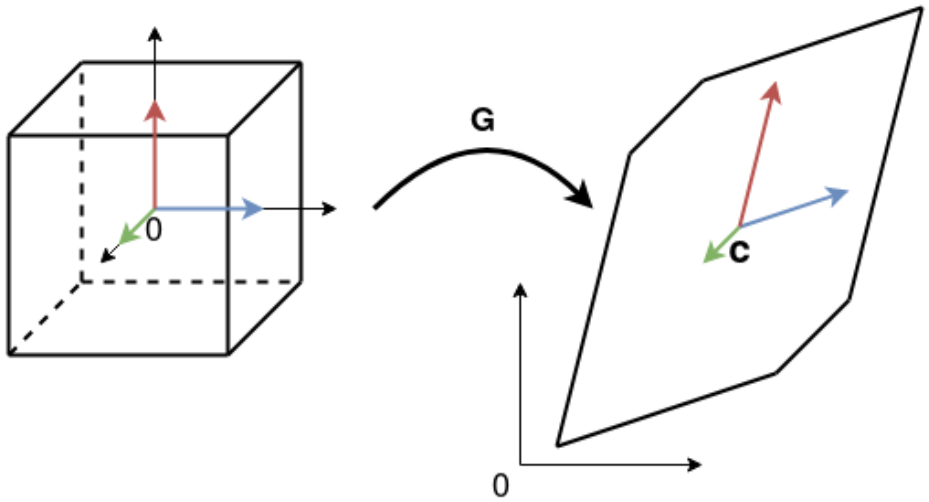
- $a\hat{x} + b\hat{y} + c = (a\alpha_0 + b\beta_0 + c) + \sum_{i=1}^{n}(a\alpha_i + b\beta_i)\varepsilon_i$

Most operations create new noise symbols, e.g. multiplication:
$\hat{x} \times \hat{y} = \alpha_0\alpha_1 + \sum_{i=1}^{n}(\alpha_i\beta_0 + \alpha_0\beta_i)\varepsilon_i + \nu\varepsilon_{n+1}$, where
$\nu = \left(\sum_{i=1}^{n} |\alpha_i|\right) \times \left(\sum_{i=1}^{n} |\beta_i|\right)$ over-approximates non linearities.

Arithmetic operations done with intervals to preserve guarantees!

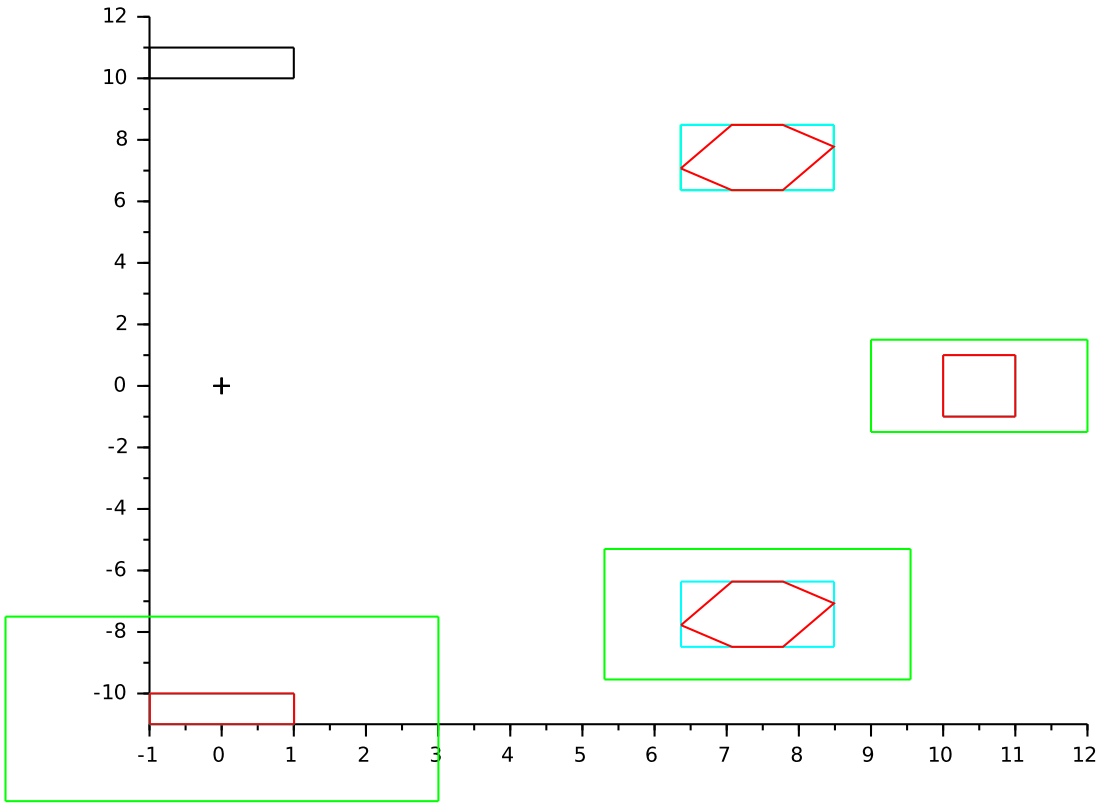
With zonotopes

Zonotopes are geometrical concretization of affine arithmetic



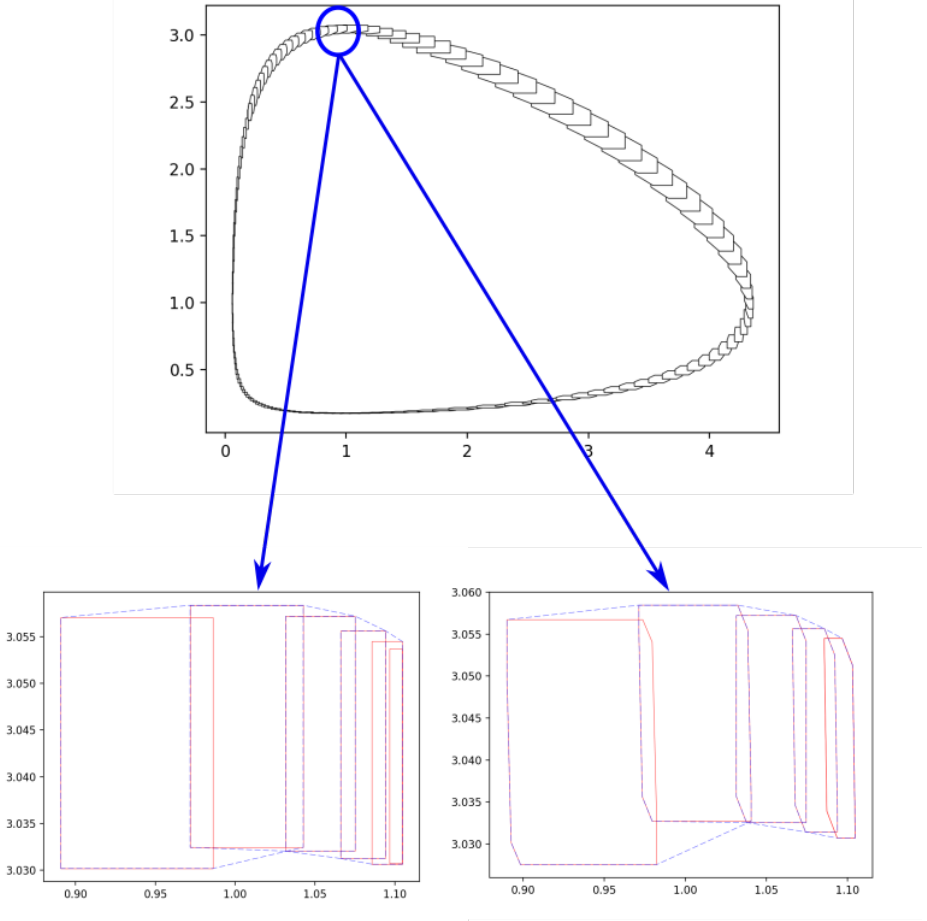
With zonotopes

Main property (that interests us): centrally symmetric polytope
$\Rightarrow$ robust to rotation.



With zonotopes

Volterra boxes vs zonotopes



With polytopes

- Zonotopes have a restrictive shape (need symmetry)
- Polytopes have better expressivity, come from linear constraints, etc
- But no associated arithmetic

Geometrical technique

A polytope $\mathcal{P}$ can be represented exactly by the intersection of a finite number of zonotopes: $\mathcal{P} = Z_1 \cap \cdots \cap Z_n$
Image of a function f on polytope can be computed as $f(\mathcal{P}) = f(Z_1 \cap \cdots \cap Z_n) \subseteq f(Z_1) \cap \cdots \cap f(Z_n)$ where $f(Z_1), \cdots, f(Z_n)$ can be computed using affine arithmetic.

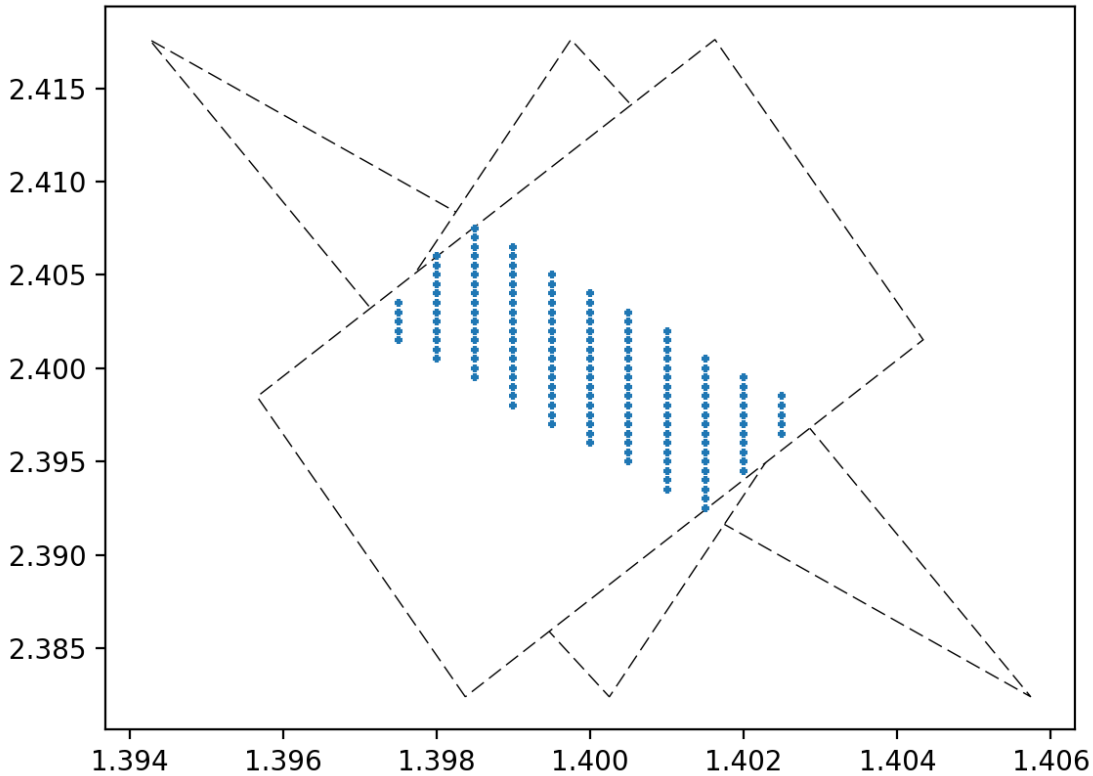
B-series with Polytopes

If $\mathcal{Y}_0$ enclosed by a polytope $\mathcal{P}_0$ and if $\mathcal{P}_0 = \mathcal{Z}^0 \cap \mathcal{Z}^1 \cap \cdots \cap \mathcal{Z}^z$, then $\mathcal{Y}(h) = B(E, h\mathcal{F}, \mathcal{Y}_0) \subset B(E, h\mathcal{F}, \mathcal{P}_0) =$
$B(E, h\mathcal{F}, \mathcal{Z}^0) \cap B(E, h\mathcal{F}, \mathcal{Z}^1) \cap \cdots \cap B(E, h\mathcal{F}, \mathcal{Z}^z)$

With polytopes

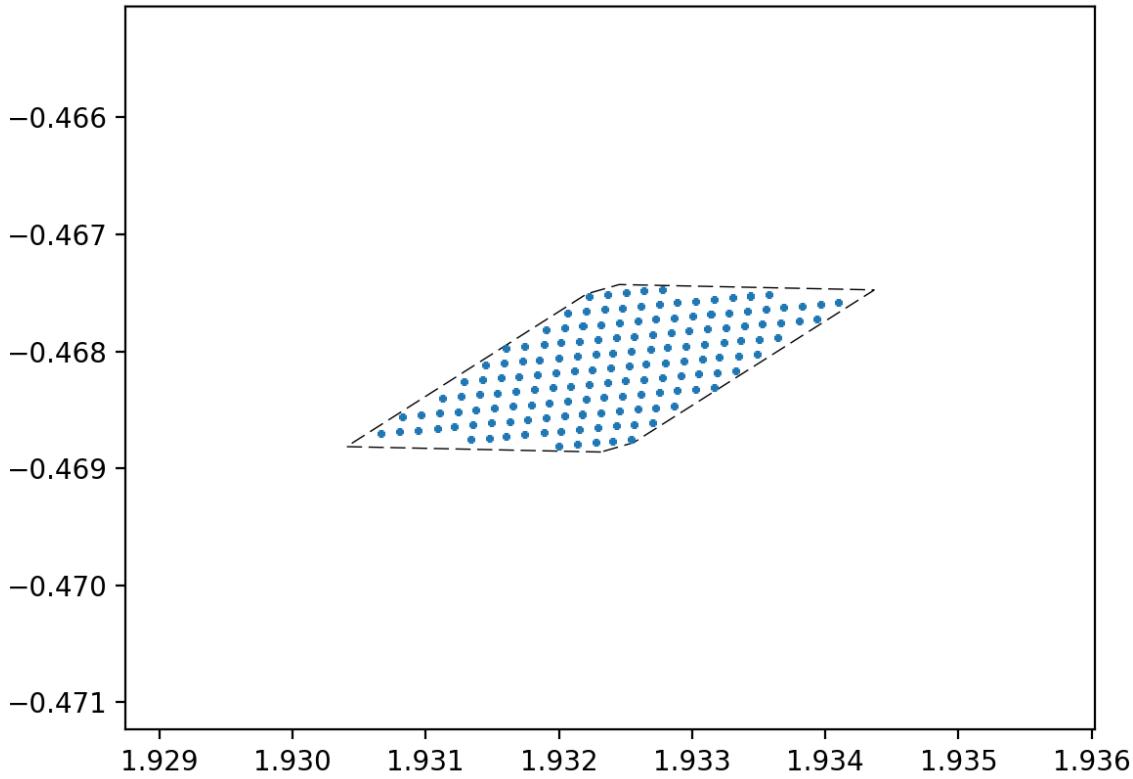
ENSTA IP PARIS

VanDerPol (initial condition)



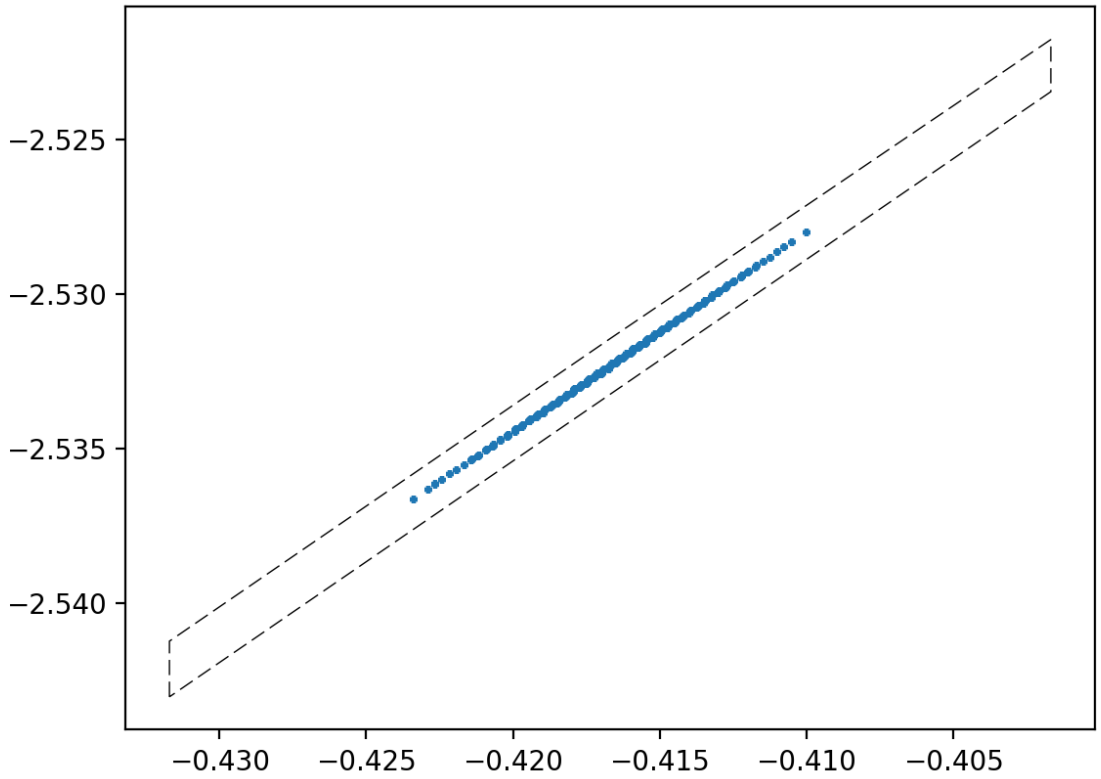
With polytopes

VanDerPol (after 1 seconds)



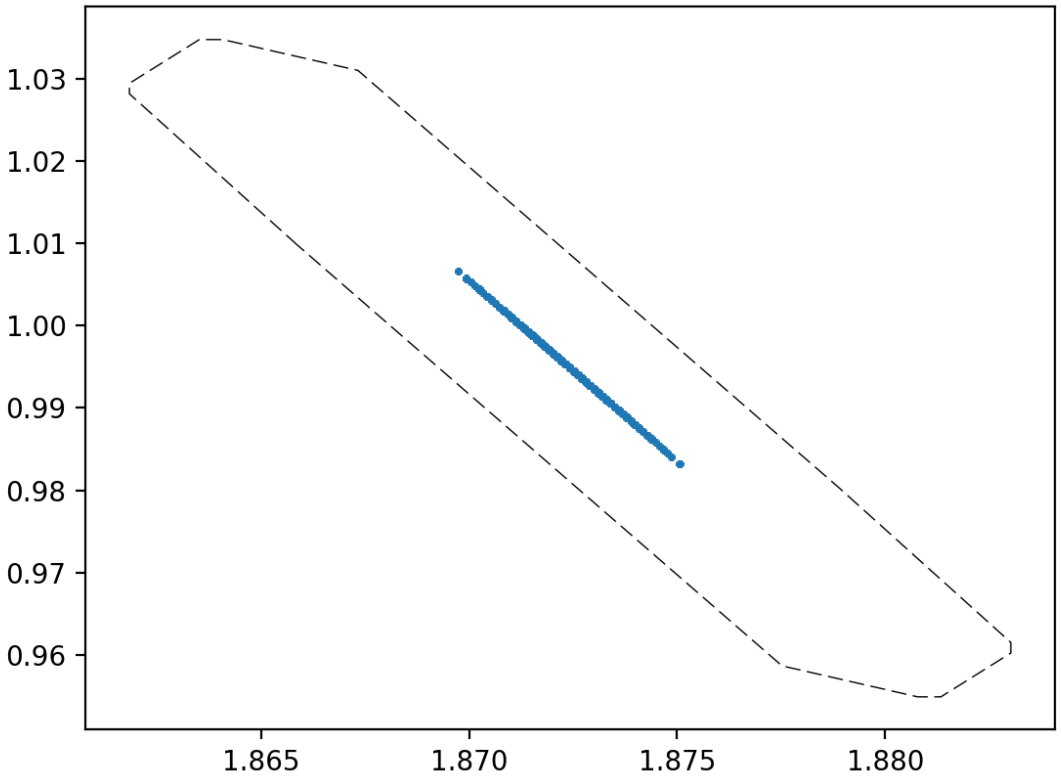
With polytopes

VanDerPol (after 3 seconds)



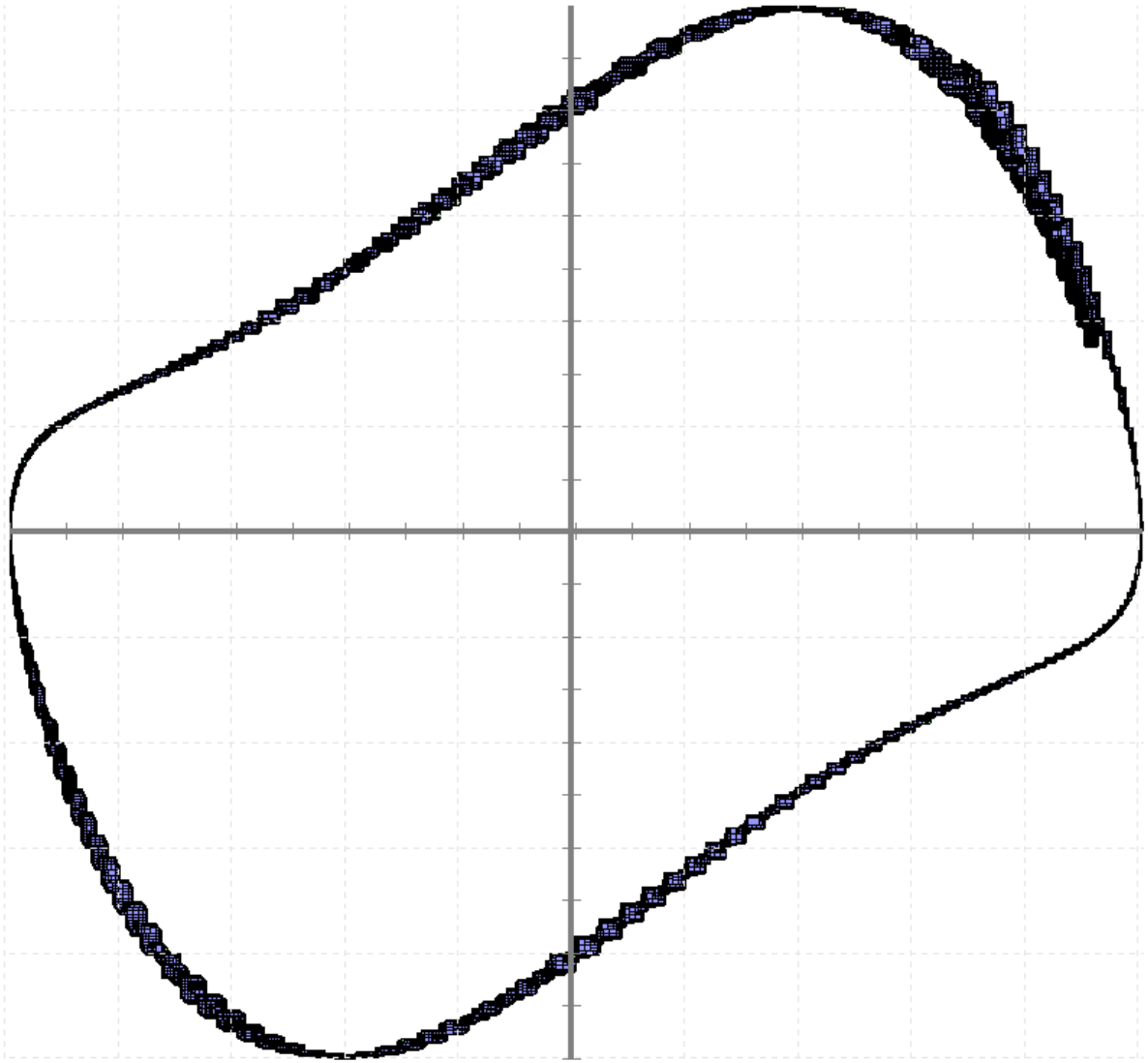
With polytopes

VanDerPol (after 7 seconds)



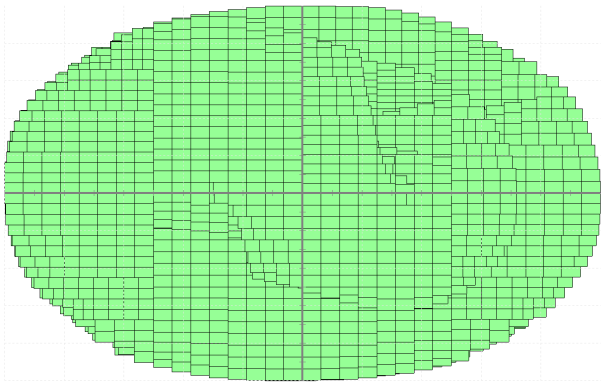
With polytopes

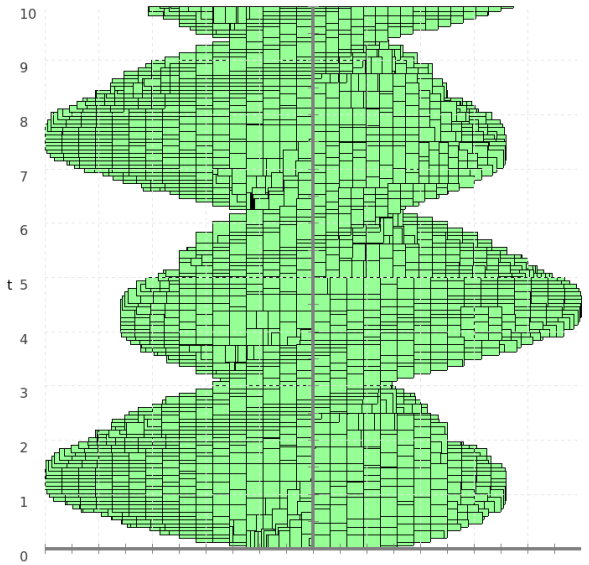
VanDerPol (tube)



With polytopes

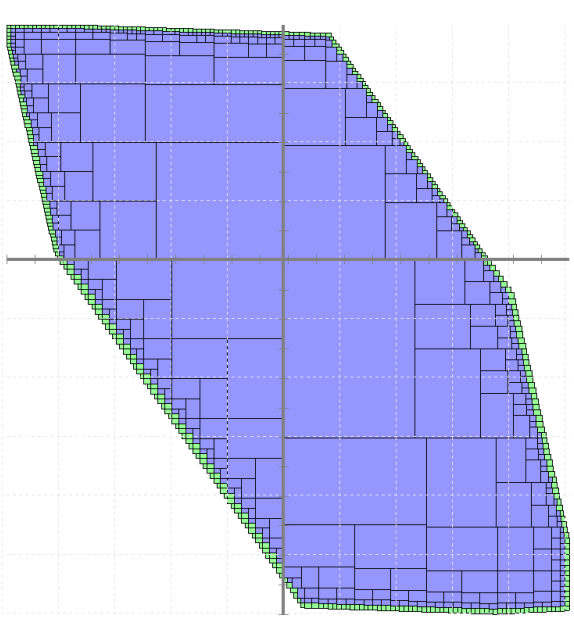
Another example, the circle

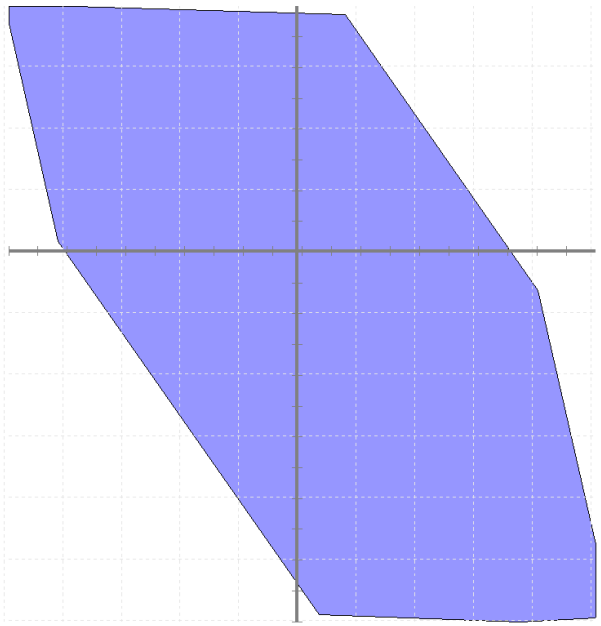




With polytopes

- Drawbacks: concretization very expensive, intersection of zonotopes computed by constraint programming (paving with boxes vs polytope obtained with Absolute)
- But perfect for collision or fault detection (we hope no solution)





Temporal Logic, Constraints and Reachability

Constraints and reachability

Properties on **dynamical systems** (mainly in robotics):

Verification

- Formal proof of safety
- Viability
- Reachability
- Limit cycle

Synthesis

- Parameter identification
- Control
- Optimization
- Motion Planning

Dynamical systems

- Dynamics $\Rightarrow$ Differential equations (ODEs or DAEs)
- Properties $\Rightarrow$ Constraints

Set-based constraints

Some points of attention

- While solving with guaranteed approches (ie intervals) hidden relaxations are added !
  $\Rightarrow$ Known problem of $a = 0$ (with $a \in [a]$) becoming $[a] \subset [-\epsilon, \epsilon]$
- $f(x) > 0$ translated to $f([x]) > 0$ can be undetermined
- Tube is an enclosure of trajectories: some are not feasible

Verification of dynamical properties

Interval Boolean

Interval Boolean set $\mathbb{IB} = \{\emptyset, [0,0], [1,1], [0,1]\}$

- $[0,0]$ = False and $[1,1]$ = True
- $\emptyset$ = impossible and $[0,1]$ = undetermined

Operations

$[a] \in \mathbb{IB}$ and $[b] \in \mathbb{IB}$:

- $[a] \wedge [b] = \{a \wedge b \mid a \in [a], b \in [b]\}$;
- $[a] \vee [b] = \{a \vee b \mid a \in [a], b \in [b]\}$;
- $\neg[a] = \{\neg a \mid a \in [a]\}$.

Verification of dynamical properties

Only some tests on tubes

If $[\mathbf{y}](t, \mathbf{p}, \mathbf{y_0})$ is the (interval, over-approximated) reachable tube, some temporal properties:

Verbal property	Associated test
Stay in $\mathcal{A}$	$\forall t \in [0, t_{\text{end}}],\ [\mathbf{y}](t, \mathbf{p}, \mathbf{y_0}) \subseteq \text{Int}(\mathcal{A})$
In $\mathcal{A}$ at τ	$[\mathbf{y}](\tau, \mathbf{p}, \mathbf{y_0}) \subseteq \text{Int}(\mathcal{A})$
Has crossed $\mathcal{A}$	$\exists t \in [0, t_{\text{end}}],\ [\mathbf{y}](t, \mathbf{p}, \mathbf{y_0}) \cap \text{Hull}(\mathcal{A}) \neq \emptyset$
Go out $\mathcal{A}$	$\exists t \in [0, t_{\text{end}}],\ [\mathbf{y}](t, \mathbf{p}, \mathbf{y_0}) \cap \text{Hull}(\mathcal{A}) = \emptyset$
Has reached $\mathcal{A}$	$[\mathbf{y}](t_{\text{end}}, \mathbf{p}, \mathbf{y_0}) \cap \text{Hull}(\mathcal{A}) \neq \emptyset$
Finished in $\mathcal{A}$	$[\mathbf{y}](t_{\text{end}}, \mathbf{p}, \mathbf{y_0}) \subseteq \text{Int}(\mathcal{A})$

Correct with respect to set abstraction, can be used in a branching algorithm (such as a SIVIA).

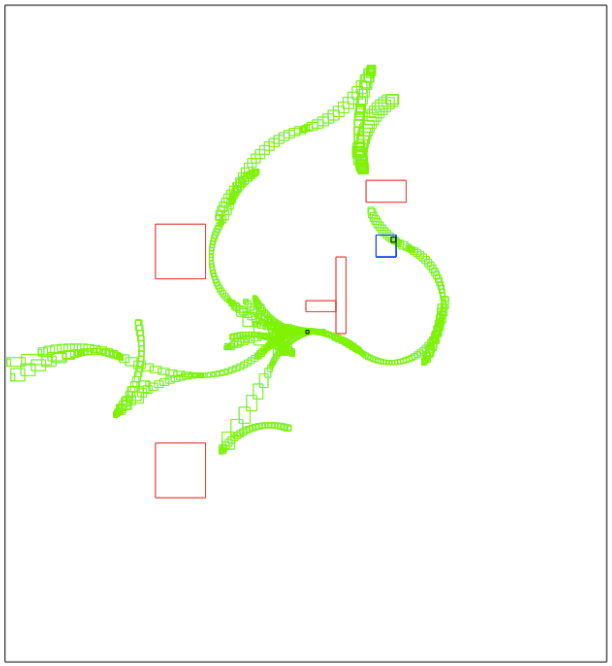
Applications in Robotics

Four results briefly recalled

A recent one in detail

Motion planning (1)

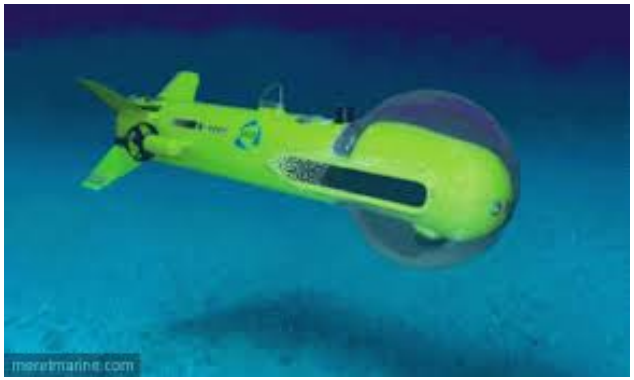
2D robot: find a path from A to B, avoiding obstacles and respecting dynamical constraints

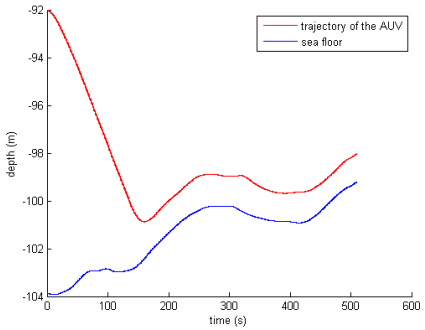


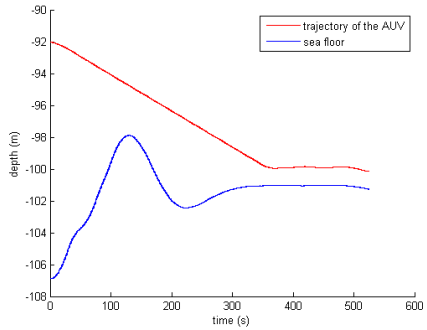
Formal Verification of Robotic Behaviors in Presence of Bounded Uncertainties, J. Alexandre dit Sandretto, A. Chapoutot, O. Mullier, 2017, IEEE International Conference on Robotic Computing (IRC)

Motion planning (2)

Submarine robot: follow the sea bed, avoiding
it but keeping short distance (sensor) and respecting dynamical constraints



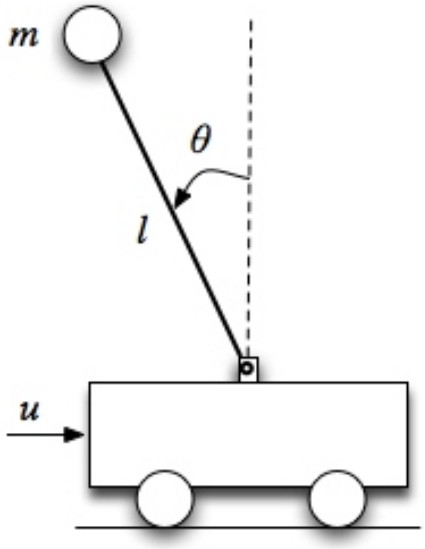


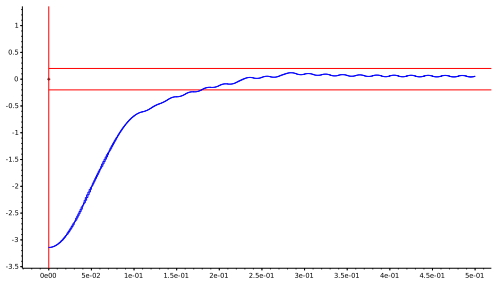


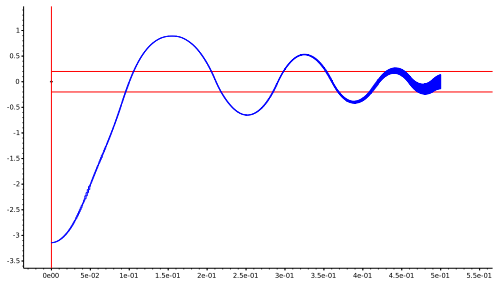
An Interval-based Sliding Horizon Motion Planning Method, E. Brendel, J. Alexandre dit Sandretto, A. Chapoutot, 2018, IFAC-PapersOnLine

Optimal control (1)

Inverse pendulum: model predictive control



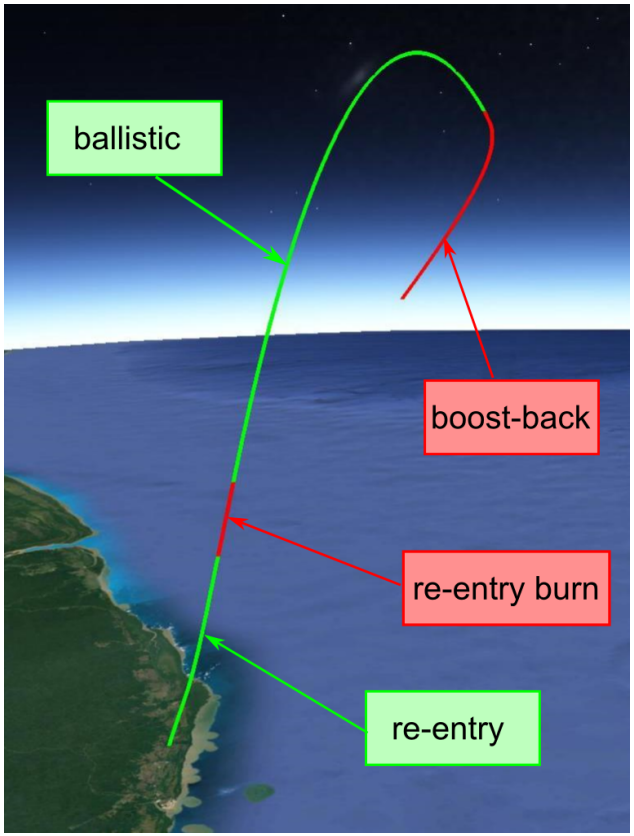


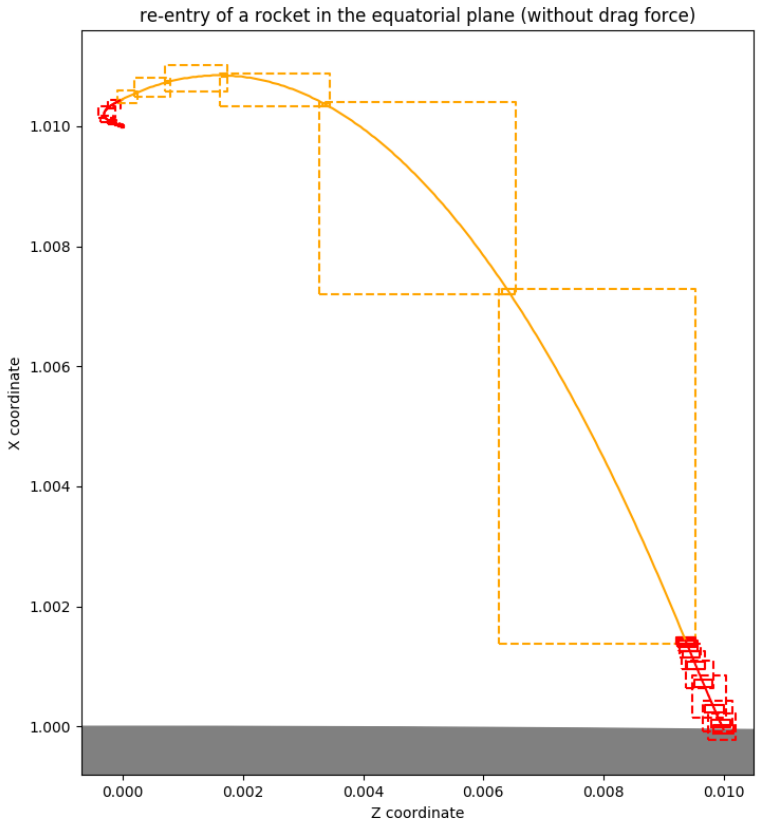


Reliable NonLinear Model-Predictive Control via Validated Simulation, J. Alexandre dit Sandretto, 2018, Annual American Control Conference (ACC)

Optimal control (2)

Rocket: re-entry by optimal control (Pontryagin)





Spatio-temporal constrained zonotopes for validation of optimal control problems, E. Bertin , B. Hérissé , J. Alexandre dit Sandretto , A. Chapoutot, 2021, Conference on Decision and Control (CDC)

One application in detail

Monitor the trajectory of a drone, based on sliding horizon and reachability analysis

A first result (offline): "Trajectory Monitoring For A Drone Using Interval Analysis", S. Largent, J. Alexandre Dit Sandretto, Workshop on Planning, PPNIV'22, Kyoto, Japan

Thesis of Antoine Besset

- Implement an online monitor with ROS (his Master project)
- Signal Temporal Logic (STL) based monitor to better express temporal properties (submitted to ICRA)
- Notion of risk and success during a mission (future)

STL based monitoring

Dynamics and properties

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t)), \mathbf{y}(\mathbf{0}) \in [\mathbf{y_0}], t \in [0, h],$$

where $\mathbf{y}$ represents the state vector of the system, $[\mathbf{y_0}]$ the estimation of state at the beginning of a new horizon.

$$\varphi := \mu \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \, U_{[a,b]} \, \varphi_2,$$

where φ is an STL formula and μ represents an atomic predicate. $U_{[a,b]}$ is the "until" operator bounded by the time interval $[a, b]$. Derived operators such as $\text{Always}_{[a,b]}$ and $\text{Eventually}_{[a,b]}$ are also defined to express additional temporal constraints.

STL based monitoring

Set-based predicates

μ is extended with: $[\mathbf{A}] \cap [\mathbf{b}] = \emptyset$ and $[\mathbf{A}] \subset [\mathbf{b}]$ resulting in Boolean intervals (to be safe with asbtraction).

Drone safety

$\varphi_{\text{flag}} \equiv \text{Always}_{[0,N\times\tau]} \neg p_{\text{collision}} \wedge \text{Always}_{[0,N\times\tau]} p_{\text{environment}}$.

Example

If $[\mathbf{Y}] \cap [\mathbf{o}] = \emptyset$, there is no collision (output: 1). If $[\mathbf{Y}] \subset [\mathbf{o}]$, a collision is confirmed (output: 0). Otherwise, if $[\mathbf{Y}] \cap [\mathbf{o}] \neq \emptyset$ and $[\mathbf{Y}] \not\subset [\mathbf{o}]$, it is unclear whether a collision will occur (output: $[0, 1]$).

STL based monitoring - in video

Thanks to Antoine!

Conclusion and Future Works

Conclusion

- Runge-Kutta with intervals, zonotopes, polytopes,
- Set-based B-series,
- Implementation in DynIbex,
- Constraints with differential equations.

Conclusion

Some results about theory and applications in robotics/safety

Theory

- **Validated Runge-Kutta schemes:**
  "Validated Explicit and Implicit Runge-Kutta Methods", J. Alexandre Dit Sandretto, A. Chapoutot, Reliable Computing, 2016
- **Set-based B-series:**
  "Validated B-series and Runge-Kutta pairs", J. Alexandre Dit Sandretto, Numerical Algorithms, 2023
- **Constraints with ODEs:**
  "Logical Differential Constraints Based on Interval Boolean Tests", J. Alexandre Dit Sandretto, A. Chapoutot, IFSA/NAFIPS 2019;
  "Constraint-Based Framework for Reasoning with Differential Equations", J. Alexandre Dit Sandretto, A. Chapoutot, O. Mullier, Cyber-Physical Systems Security, 2018
- **New RK schemes:**
  "Runge-Kutta Theory and Constraint Programming (Extended version with J. Butcher)", J. Alexandre Dit Sandretto, Reliable Computing, 2019

Conclusion

Applications

- **Motion planning:**
  "An Interval-based Sliding Horizon Motion Planning Method", J. Alexandre Dit Sandretto, E. Brendel, A. Chapoutot IFAC, 2018
- **Parameter identification:**
  "Tuning PI controller in non-linear uncertain closed-loop systems with interval analysis", J. Alexandre Dit Sandretto, A. Chapoutot, O. Mullier, Workshop on Synthesis of Complex Parameters, 2015
- **Appropriate design:**
  "Appropriate Design Guided by Simulation: An Hovercraft Application", J. Alexandre Dit Sandretto, D. Piccani de Souza, A. Chapoutot, Workshop on Model-Driven Robot Software Engineering, 2016
- **Optimal control:**
  "Prospects on Solving an Optimal Control Problem with Bounded Uncertainties on Parameters using Interval Arithmetics", E. Bertin, et al., Acta Cybernetica, 2021;
  "Reliable NonLinear Model-Predictive Control via Validated Simulation", J. Alexandre Dit Sandretto, ACC, 2018

Future Works

PhD supervision

- Lucas Si Larbi, Situation model and movement planning of a mobile robot in an unstructured environment, with CEA, 3rd year
- Joel Bourgon, Action planning for optimization of an engagement sequence with tactical mobility in a real-time framework, with KNDS and St Cyr Coetquidan, 2nd year
- Antoine Besset, STL based monitoring with notion of risk and success, 2nd year
- Hippolyte Watrelot, Synthesis of a controller integrating the notion of risk and temporal logic with security guarantees, 2nd year

Future Works

Personal researches (with external collaborations)

- Set-based B-series is an interesting object,
- New RK schemes dedicated to set-based B-series,
- New implementation based on error-free affine arithmetic,
- In Lie Group,
- STL with sets and probabilities.