A short introduction to Codac

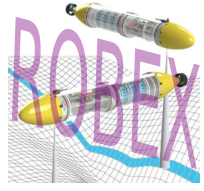
SWIM

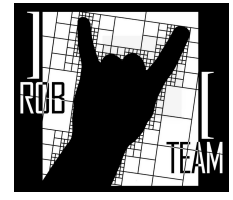
30th June 2025, Rennes, France

Tutorial proposed by Fabrice Le Bars, Maël Godard, Damien Massé, Simon Rohou

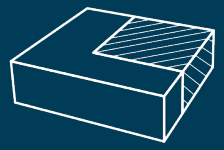
ENSTA, Lab-STICC, Brest, France

`https://codac.io/v2/tuto/cp_robotics`





Codac in a nutshell



codac
Catalog Of Domains And Contractors

Domains (wrappers)

- for reals $x \in \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^n$: intervals $[x]$ and boxes $[\mathbf{x}]$
- for trajectories $x(\cdot) : \mathbb{R} \to \mathbb{R}$: tubes $[x](\cdot)$

Domains (wrappers)

- for reals $x \in \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^n$: intervals $[x]$ and boxes $[\mathbf{x}]$
- for trajectories $x(\cdot) : \mathbb{R} \to \mathbb{R}$: tubes $[x](\cdot)$
- for subsets $\mathbb{X} \subset \mathbb{R}^n$: thicksets $\mathbb{X} \in [\mathbb{X}] = [\mathbb{X}^-, \mathbb{X}^+]$

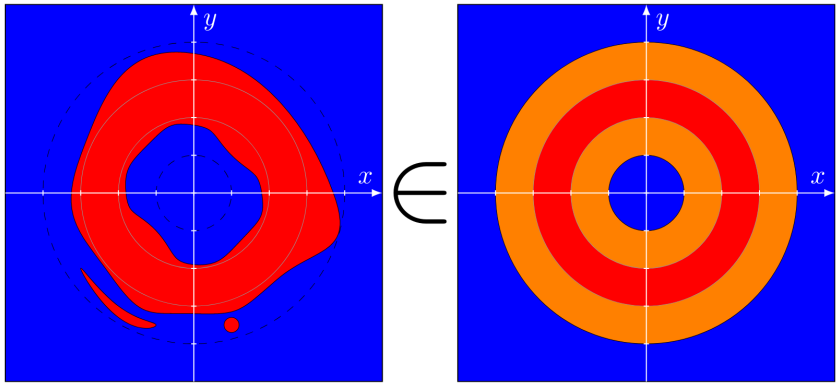


Illustration of a thickset (right-hand side)
for enclosing and uncertain red set (left-hand side)

■ Thick set inversion
Desrochers, Jaulin. *Artificial Intelligence. Volume 249, Issue C, Pages 1-18*, 2017

Domains (wrappers)

- for reals $x \in \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^n$: intervals $[x]$ and boxes $[\mathbf{x}]$
- for trajectories $x(\cdot) : \mathbb{R} \to \mathbb{R}$: tubes $[x](\cdot)$
- for subsets $\mathbb{X} \subset \mathbb{R}^n$: thicksets $\mathbb{X} \in [\mathbb{X}] = [\mathbb{X}^-, \mathbb{X}^+]$
- *etc.*

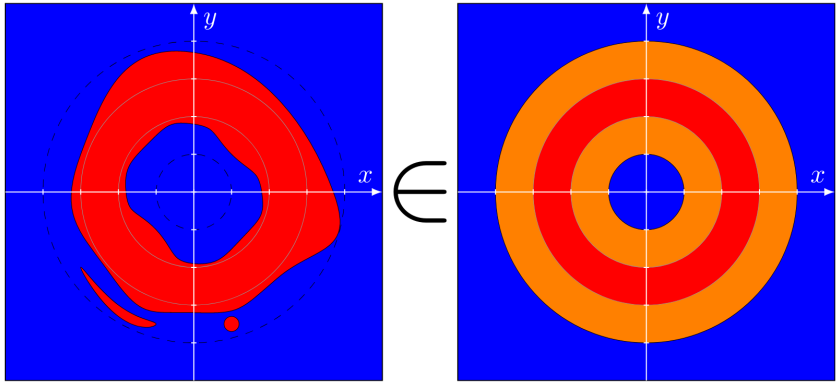
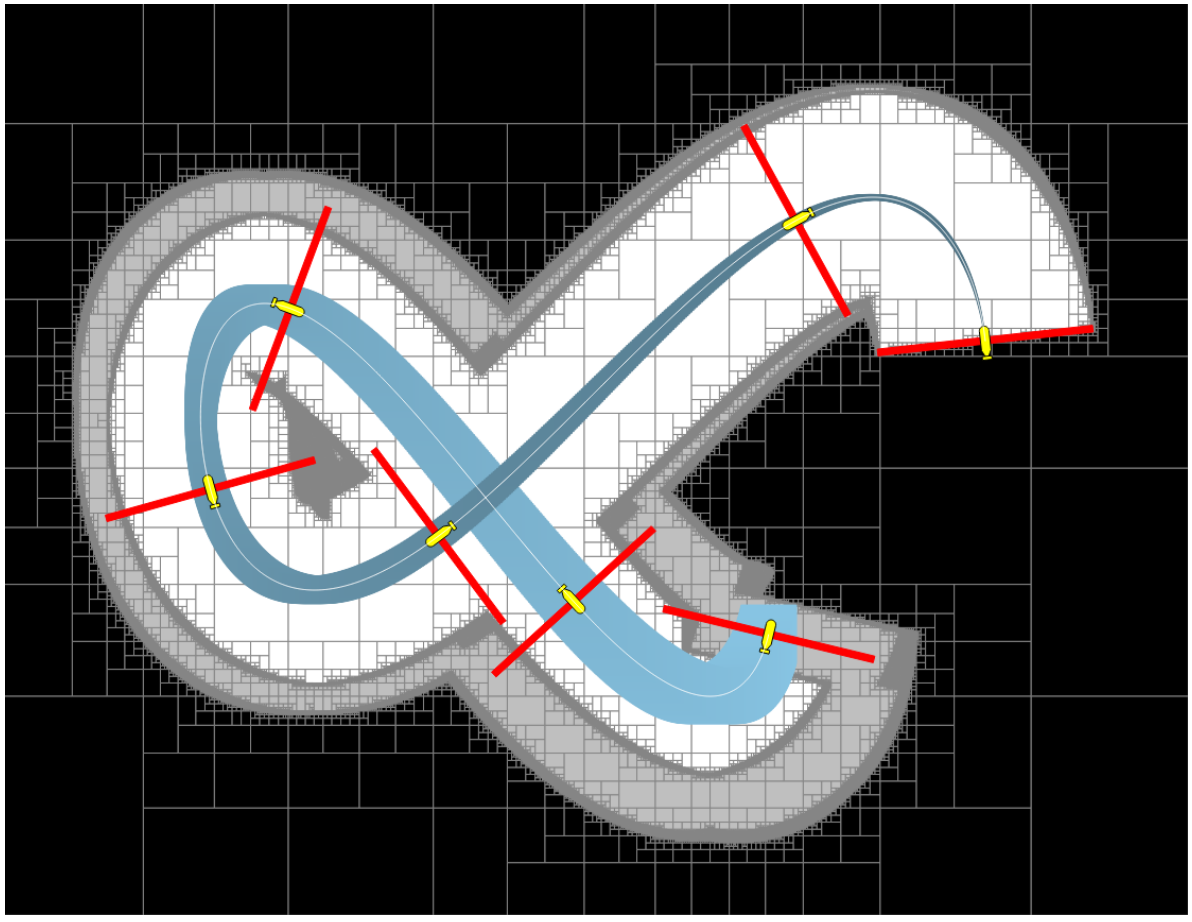


Illustration of a thickset (right-hand side)
for enclosing and uncertain red set (left-hand side)

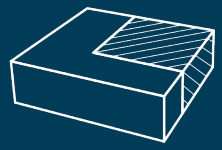
■ Thick set inversion
Desrochers, Jaulin. *Artificial Intelligence. Volume 249, Issue C, Pages 1-18*, 2017

codac
Catalog Of Domains And Contractors

Example of mixing tubes and thicksets



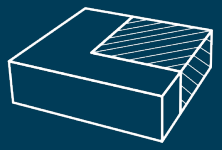
- Computing a Guaranteed Approximation of the Zone Explored by a Robot. Desrochers, Jaulin. *IEEE Trans. on Automatic Control. Volume 62, Issue 1.*, 2017



Catalog Of Domains And Contractors

Several types of **domains**:

- `Interval` (mainly from the GAOL library)
- `IntervalVector`, `IntervalMatrix` (using Eigen templates)
- `SlicedTube<T>...`, `Slice<T>...`
- `Ellipsoid`, `Paving`, `Thickset`, ...



Catalog Of Domains And Contractors

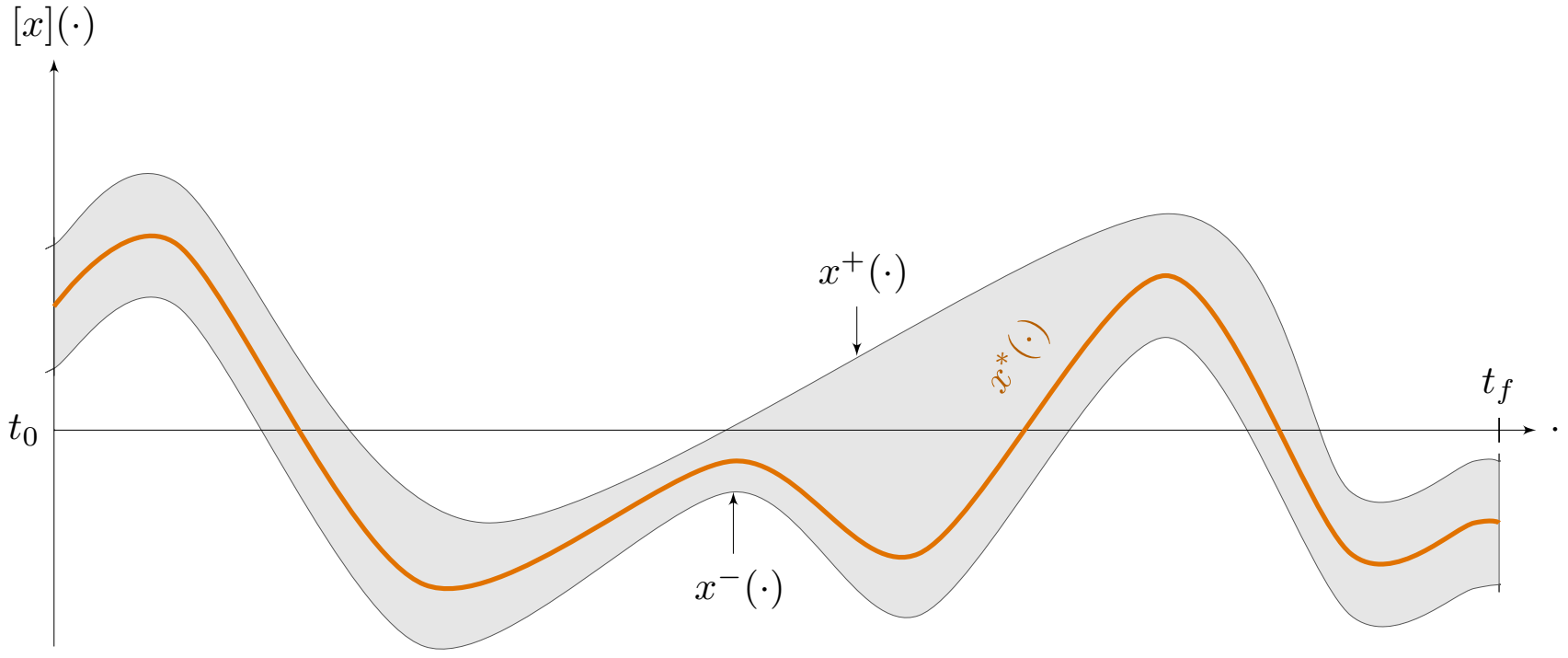
Several types of **domains**:

- `Interval` (mainly from the GAOL library)
- `IntervalVector`, `IntervalMatrix` (using Eigen templates)
- `SlicedTube<T>...`, `Slice<T>...`
- `Ellipsoid`, `Paving`, `Thickset`, ...

Contractors for various constraints:

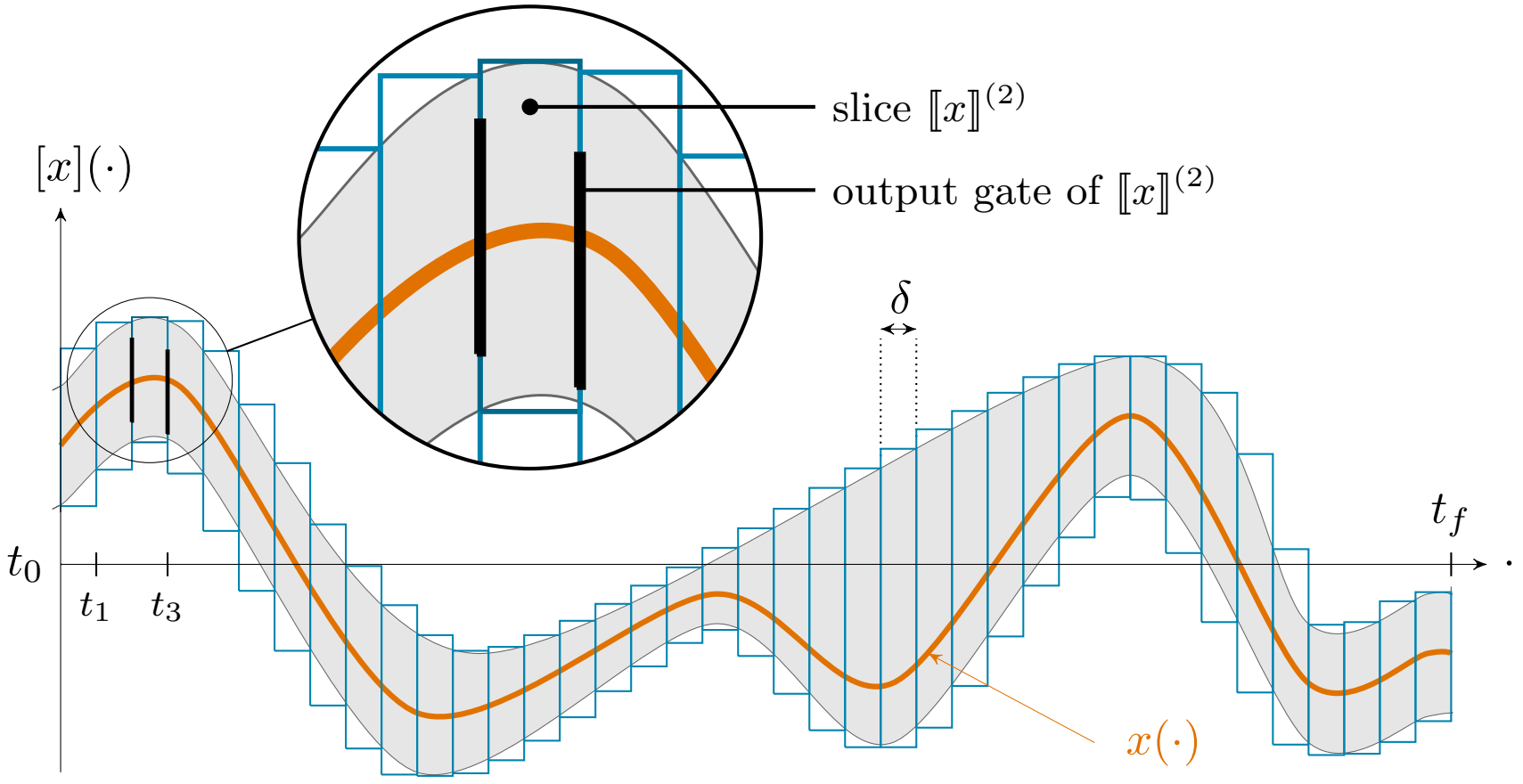
- non-linear constraints $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ (involving centered form)
- geometric constraints: distance, polar equation, no-cross, ...
- differential equations: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$, $\int\int$
- time uncertainties: $\mathbf{y} = \mathbf{x}(t), \quad \text{with } t \in [t]$
- delays: $x(t) = y(t - \tau)$
- ...

Domains for trajectories: tubes



Example of scalar tube: interval of two trajectories

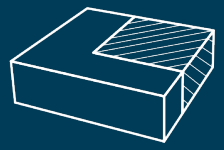
Domains for trajectories: tubes



The library is open source and available:

- in Python, C++, Matlab
- on Linux, Windows, MacOS systems
- from official packages:
  Python package: `pip install codac`
  Debian in progress..: `sudo apt install libcodac`

`http://www.codac.io`

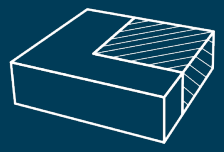


codac
Catalog Of Domains And Contractors

Current list of contributors

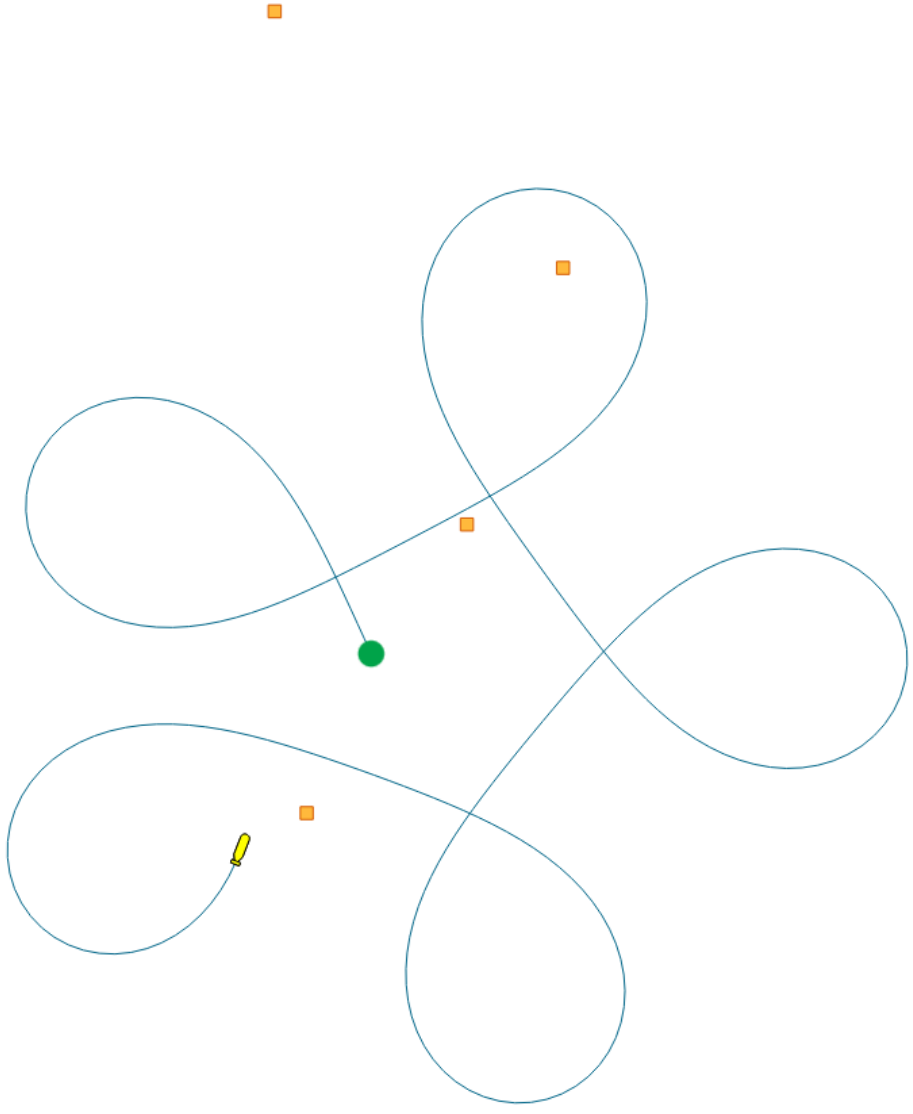
- Auguste Bourgois
- Cyril Bouvier
- Quentin Brateau
- Gilles Chabert
- Julien Damers
- Benoît Desrochers
- Peter Franek
- Maël Godard
- Nuwan Herath M.
- Luc Jaulin
- Fabrice Le Bars
- Morgan Louédec
- Damien Massé
- Bertrand Neveu
- Verlein Radwan
- Andreas Rauh
- Simon Rohou
- Joris Tillet
- Gilles Trombettoni
- Christophe Viel
- Raphael Voges

Constraint programming



codac
Catalog Of Domains And Contractors

SLAM: Simultaneous Localization And Mapping



Formalization

SLAM: Simultaneous Localization And Mapping.

Classically, we have:

$$\begin{cases} \mathbf{x}(0) = \mathbf{0} & (\text{initial state}) \\ \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) & (\text{evolution}) \end{cases}$$

With:

- $\mathbf{x}$: state vector (position, heading, ...)
- $\mathbf{u}$: input vector (command)
- $\mathbf{f}$: *evolution* function

Formalization

SLAM: Simultaneous Localization And Mapping.

Classically, we have:

$$\begin{cases} \mathbf{x}(0) = \mathbf{0} & (\text{initial state}) \\ \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) & (\text{evolution}) \\ y_i = g\big(\mathbf{x}(t_i), \mathbf{b}_j\big) & (\text{observations}) \end{cases}$$

With:

- $\mathbf{x}$: state vector (position, heading, ...)
- $\mathbf{u}$: input vector (command)
- $\mathbf{f}$: *evolution* function
- g: *observation* function (scalar, distance equation)
- y_i: scalar measurements (at t_i) (distance values)
- $\mathbf{b}_j$: unknown position of a landmark

Formalization

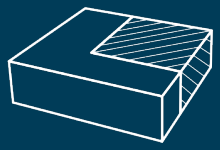
SLAM: Simultaneous Localization And Mapping.

Classically, we have:

$$\begin{cases} \mathbf{x}(0) = \mathbf{0} & \text{(initial state)} \\ \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) & \text{(evolution)} \\ y_i = g\big(\mathbf{x}(t_i), \mathbf{b}_j\big) & \text{(observations)} \end{cases}$$

With:

- $\mathbf{x}$: state vector (position, heading, ...)
- $\mathbf{u}$: input vector (command)
- $\mathbf{f}$: *evolution* function
- g: *observation* function (scalar, distance equation)
- y_i: scalar measurements (at t_i) (distance values)
- $\mathbf{b}_j$: unknown position of a landmark

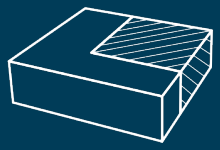


codac
Catalog Of Domains And Contractors

Involved variables and domains

Variables:

- reals: $y_i \in \mathbb{R}$
- vectors: $\mathbf{b}_j \in \mathbb{R}^2$
- trajectories: $\mathbf{x}(\cdot) : \mathbb{R} \to \mathbb{R}^n$



codac
Catalog Of Domains And Contractors

Involved variables and domains

Variables:

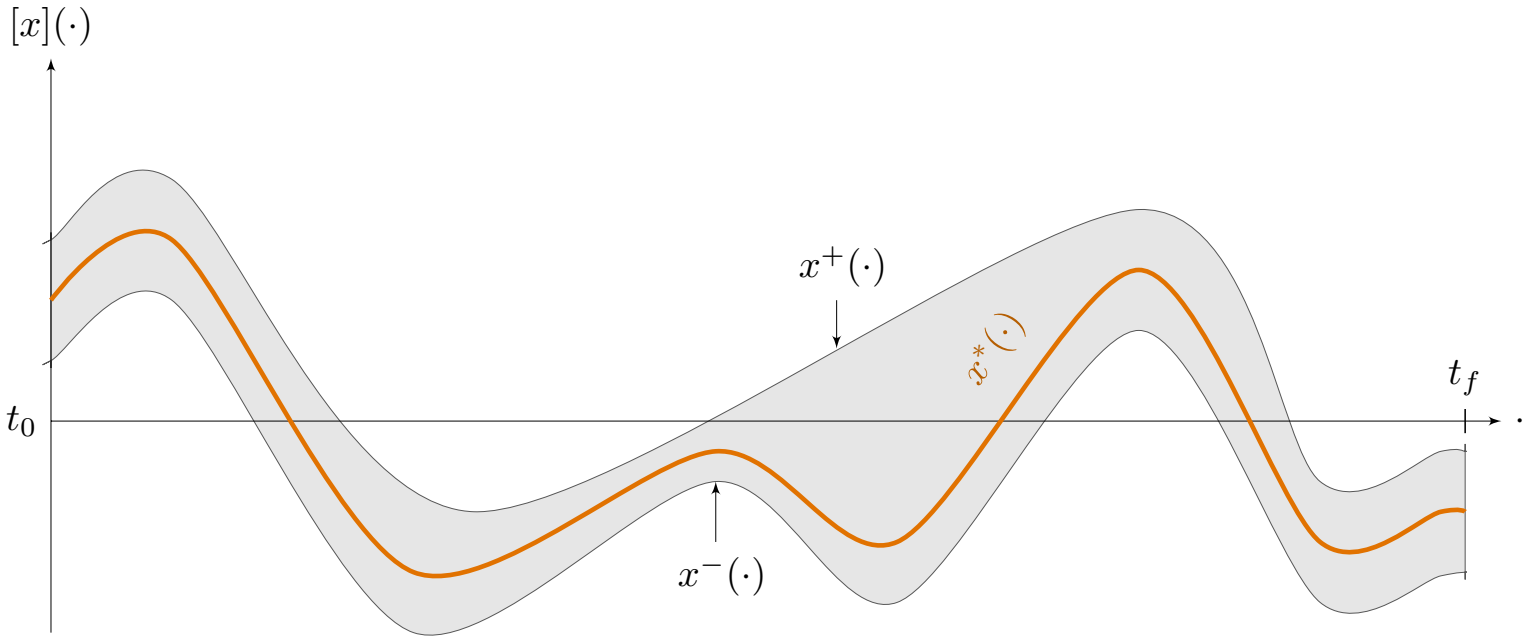
- reals: $y_i \in \mathbb{R}$
- vectors: $\mathbf{b}_j \in \mathbb{R}^2$
- trajectories: $\mathbf{x}(\cdot) : \mathbb{R} \to \mathbb{R}^n$

Domains (envelopes) of the variables:

- intervals: $[y_i] \in \mathbb{IR}$
- boxes: $[\mathbf{b}_j] \in \mathbb{IR}^2$
- tubes: $[\mathbf{x}](\cdot) : \mathbb{R} \to \mathbb{IR}^n$

Tubes: domains for trajectories

Tube $[x](\cdot)$**:** interval of trajectories $[x^-(\cdot), x^+(\cdot)]$
such that $\forall t \in \mathbb{R},\ x^-(t) \leqslant x^+(t)$



Tube $[x](\cdot)$ *enclosing an uncertain trajectory* $x(\cdot)$

Decomposition of the problem

System:

$$\begin{cases} \dot{\mathbf{x}}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot)\big) \\ y_i = g\big(\mathbf{x}_{1,2}(t_i), \mathbf{b}_j\big) \end{cases}$$

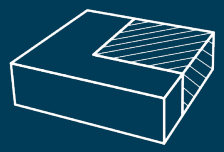
Decomposition of the problem

System:

$$\begin{cases} \dot{\mathbf{x}}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot)\big) \\ y_i = g\big(\mathbf{x}_{1,2}(t_i), \mathbf{b}_j\big) \end{cases}$$

Elementary constraints:

- $\mathbf{v}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot)\big) \rightarrow$ algebraic constraint $\rightarrow \mathcal{C}_{\mathbf{f}}\big([\mathbf{x}](\cdot), [\mathbf{v}](\cdot)\big)$



codac
Catalog Of Domains And Contractors

Decomposition of the problem

System:

$$\begin{cases} \dot{\mathbf{x}}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot)\big) \\ y_i = g\big(\mathbf{x}_{1,2}(t_i), \mathbf{b}_j\big) \end{cases}$$

Elementary constraints:

- $\mathbf{v}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot)\big) \rightarrow$ algebraic constraint $\rightarrow \mathcal{C}_{\mathbf{f}}\big([\mathbf{x}](\cdot), [\mathbf{v}](\cdot)\big)$
- $\dot{\mathbf{x}}(\cdot) = \mathbf{v}(\cdot) \rightarrow$ derivative constraint $\rightarrow \mathcal{C}_{\text{deriv}}\big([\mathbf{x}](\cdot), [\mathbf{v}](\cdot)\big)$

Decomposition of the problem

System:

$$\begin{cases} \dot{\mathbf{x}}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot)\big) \\ y_i = g\big(\mathbf{x}_{1,2}(t_i), \mathbf{b}_j\big) \end{cases}$$

Elementary constraints:

- $\mathbf{v}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot)\big) \rightarrow$ algebraic constraint $\rightarrow \mathcal{C}_{\mathbf{f}}\big([\mathbf{x}](\cdot), [\mathbf{v}](\cdot)\big)$
- $\dot{\mathbf{x}}(\cdot) = \mathbf{v}(\cdot) \rightarrow$ derivative constraint $\rightarrow \mathcal{C}_{\text{deriv}}\big([\mathbf{x}](\cdot), [\mathbf{v}](\cdot)\big)$
- $\mathbf{p}_i = \mathbf{x}_{1,2}(t_i) \rightarrow$ evaluation constraint $\rightarrow \mathcal{C}_{\text{eval}}\big([t_i], [\mathbf{p}_i], [\mathbf{x}_{1,2}](\cdot)\big)$

Decomposition of the problem

System:

$$\begin{cases} \dot{\mathbf{x}}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot)\big) \\ y_i = g\big(\mathbf{x}_{1,2}(t_i), \mathbf{b}_j\big) \end{cases}$$

Elementary constraints:

- $\mathbf{v}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot)\big) \rightarrow$ algebraic constraint $\rightarrow \mathcal{C}_{\mathbf{f}}\big([\mathbf{x}](\cdot), [\mathbf{v}](\cdot)\big)$
- $\dot{\mathbf{x}}(\cdot) = \mathbf{v}(\cdot) \rightarrow$ derivative constraint $\rightarrow \mathcal{C}_{\text{deriv}}\big([\mathbf{x}](\cdot), [\mathbf{v}](\cdot)\big)$
- $\mathbf{p}_i = \mathbf{x}_{1,2}(t_i) \rightarrow$ evaluation constraint $\rightarrow \mathcal{C}_{\text{eval}}\big([t_i], [\mathbf{p}_i], [\mathbf{x}_{1,2}](\cdot)\big)$
- $y_i = g\big(\mathbf{p}_i, \mathbf{b}_j\big) \rightarrow$ distance constraint $\rightarrow \mathcal{C}_{\text{dist}}\big([\mathbf{p}_i], [\mathbf{b}_j], [y_i]\big)$

Decomposition of the problem

System:

$$\begin{cases} \dot{\mathbf{x}}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot)\big) \\ y_i = g\big(\mathbf{x}_{1,2}(t_i), \mathbf{b}_j\big) \end{cases}$$

$\mathbf{v}(\cdot)$ and $\mathbf{p}_i$ are intermediate variables

---

Elementary constraints:

- $\mathbf{v}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot)\big) \rightarrow$ algebraic constraint $\rightarrow \mathcal{C}_{\mathbf{f}}\big([\mathbf{x}](\cdot), [\mathbf{v}](\cdot)\big)$
- $\dot{\mathbf{x}}(\cdot) = \mathbf{v}(\cdot) \rightarrow$ derivative constraint $\rightarrow \mathcal{C}_{\text{deriv}}\big([\mathbf{x}](\cdot), [\mathbf{v}](\cdot)\big)$
- $\mathbf{p}_i = \mathbf{x}_{1,2}(t_i) \rightarrow$ evaluation constraint $\rightarrow \mathcal{C}_{\text{eval}}\big([t_i], [\mathbf{p}_i], [\mathbf{x}_{1,2}](\cdot)\big)$
- $y_i = g\big(\mathbf{p}_i, \mathbf{b}_j\big) \rightarrow$ distance constraint $\rightarrow \mathcal{C}_{\text{dist}}\big([\mathbf{p}_i], [\mathbf{b}_j], [y_i]\big)$

Decomposition of the problem

System:

$$\begin{cases} \dot{\mathbf{x}}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot)\big) \\ y_i = g\big(\mathbf{x}_{1,2}(t_i), \mathbf{b}_j\big) \end{cases}$$

$\mathbf{v}(\cdot)$ and $\mathbf{p}_i$ are intermediate variables

Note: some symbolic solver could break down such problem automatically.

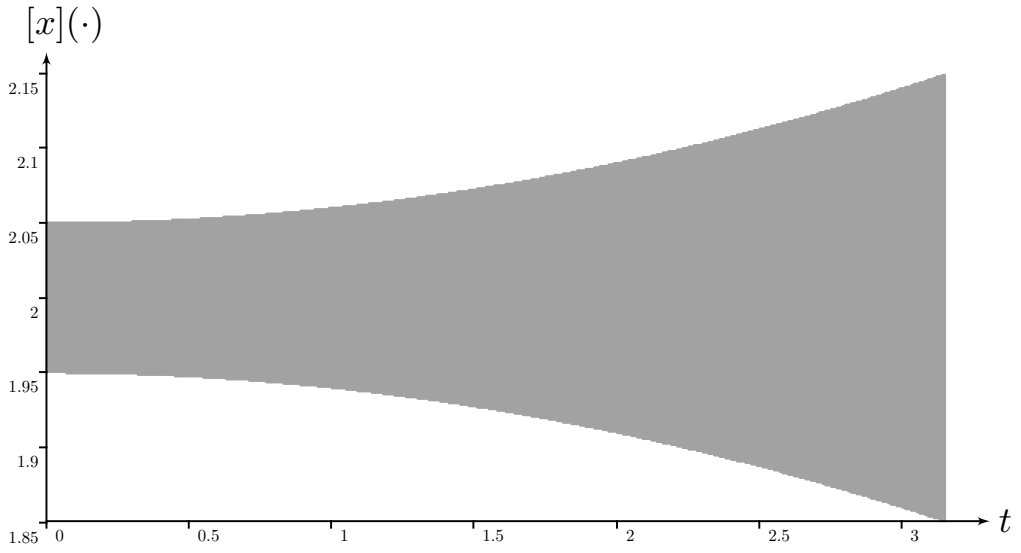
Elementary constraints:

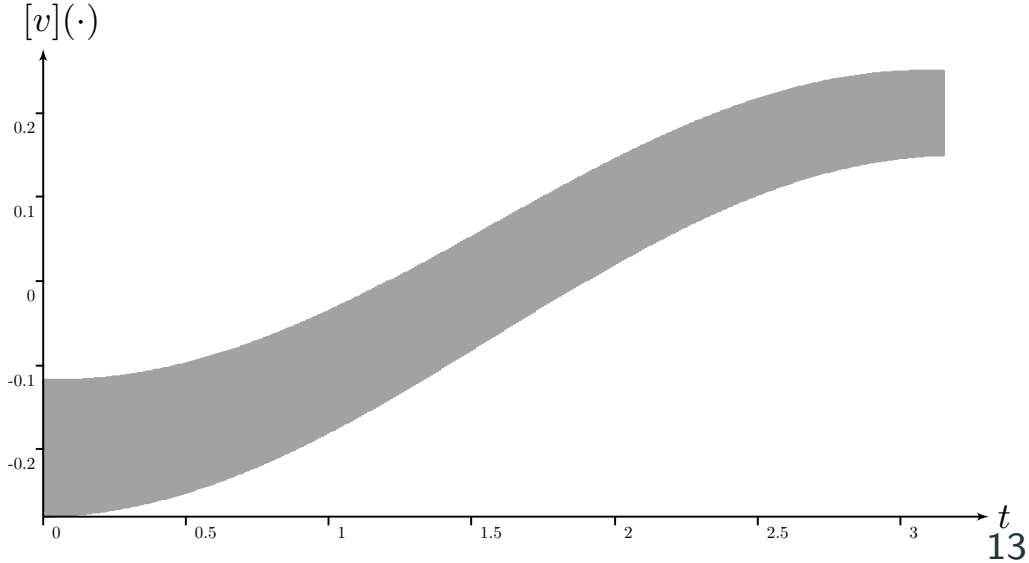
- $\mathbf{v}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot)\big) \rightarrow$ algebraic constraint $\rightarrow \mathcal{C}_{\mathbf{f}}\big([\mathbf{x}](\cdot), [\mathbf{v}](\cdot)\big)$
- $\dot{\mathbf{x}}(\cdot) = \mathbf{v}(\cdot) \rightarrow$ derivative constraint $\rightarrow \mathcal{C}_{\text{deriv}}\big([\mathbf{x}](\cdot), [\mathbf{v}](\cdot)\big)$
- $\mathbf{p}_i = \mathbf{x}_{1,2}(t_i) \rightarrow$ evaluation constraint $\rightarrow \mathcal{C}_{\text{eval}}\big([t_i], [\mathbf{p}_i], [\mathbf{x}_{1,2}](\cdot)\big)$
- $y_i = g\big(\mathbf{p}_i, \mathbf{b}_j\big) \rightarrow$ distance constraint $\rightarrow \mathcal{C}_{\text{dist}}\big([\mathbf{p}_i], [\mathbf{b}_j], [y_i]\big)$

Derivative constraint

Differential constraint:

- $\dot{\mathbf{x}}(\cdot) = \mathbf{v}(\cdot)$
- one trajectory and its derivative





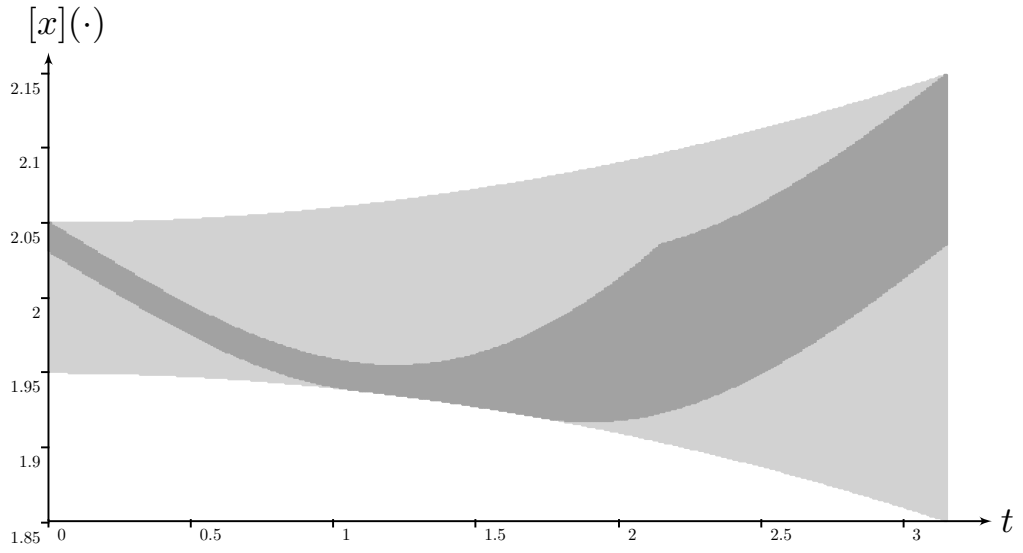
Derivative constraint

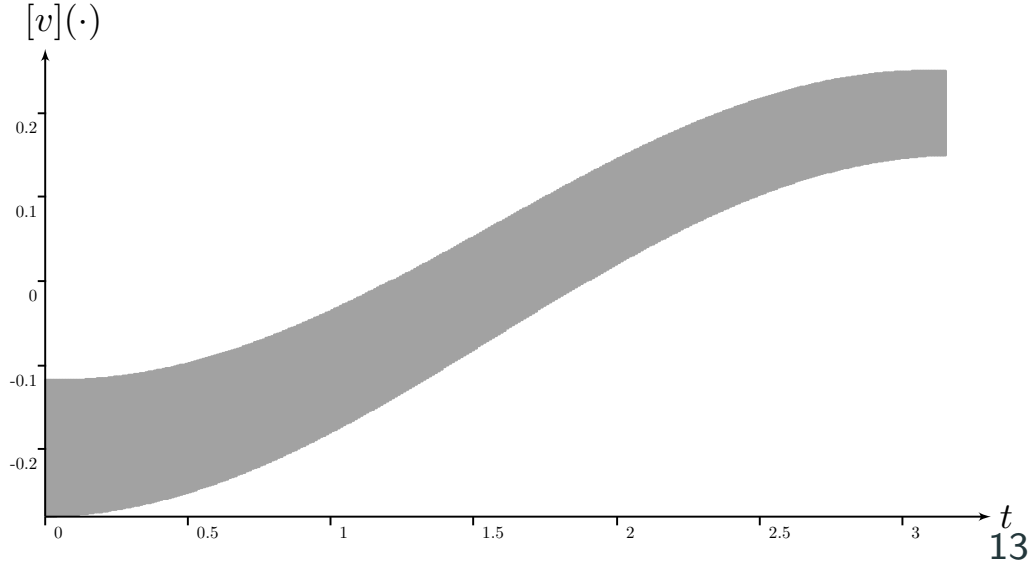
Differential constraint:

- $\dot{\mathbf{x}}(\cdot) = \mathbf{v}(\cdot)$
- one trajectory and its derivative

Contractor programming:

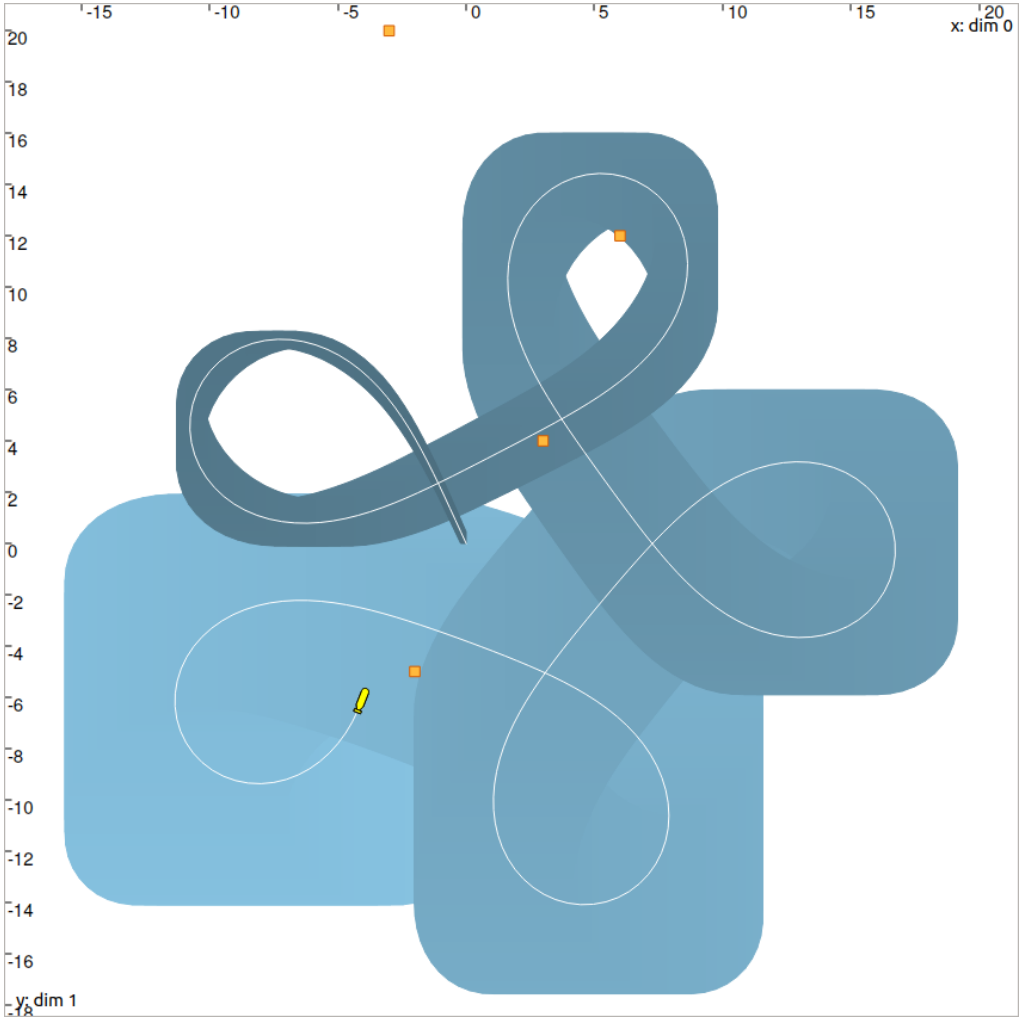
- $\mathcal{C}_{\text{deriv}}\left([\mathbf{x}](\cdot), [\mathbf{v}](\cdot)\right)$





codac
Catalog Of Domains And Contractors

SLAM results



codac
Catalog Of Domains And Contractors

SLAM results