Reliable loop-based localization method in very poor environments

Simon Rohou[1], Luc Jaulin[1], Lyudmila Mihaylova[2], Fabrice Le Bars[1], Sandor M. Veres[2]

[1] ENSTA Bretagne, Lab-STICC, UMR CNRS 6285, France
[2] University of Sheffield, Western Bank Sheffield S10 2TN, UK
simon.rohou@ensta-bretagne.org

IMT Atlantique visit
June 2017

Section 1

Introduction

Introduction

Motivations, robot localization: $\mathbf{p}(t) = ?$

Underwater exploration **without surfacing**:

- case of very deep environments (airplanes search)
- reasons of discretion and security (military mission)

Motivations, robot localization: $\mathbf{p}(t) = ?$

Underwater exploration **without surfacing**:

- case of very deep environments (airplanes search)
- reasons of discretion and security (military mission)

Need for **localization methods** based on the following constraints:

- no underwater GNSS receiver
- unstructured environment: no landmark
- opacities: limited observations

Introduction

Motivations, robot localization: $\mathbf{p}(t) = ?$

Underwater exploration **without surfacing**:

- case of very deep environments (airplanes search)
- reasons of discretion and security (military mission)

Need for **localization methods** based on the following constraints:

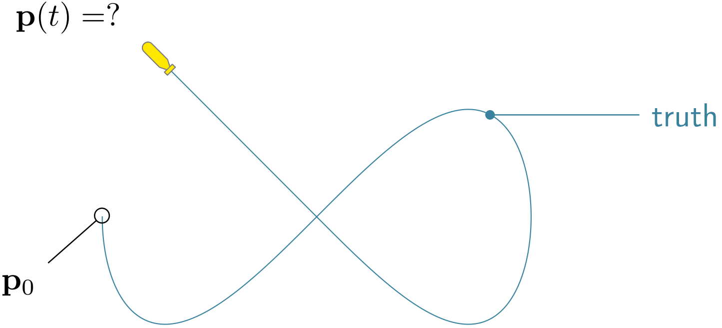
- no underwater GNSS receiver
- unstructured environment: no landmark
- opacities: limited observations

Steady solution, **dead-reckoning**:

- navigation based on *proprioceptive* measurements
- fast drift on position estimation: strong errors

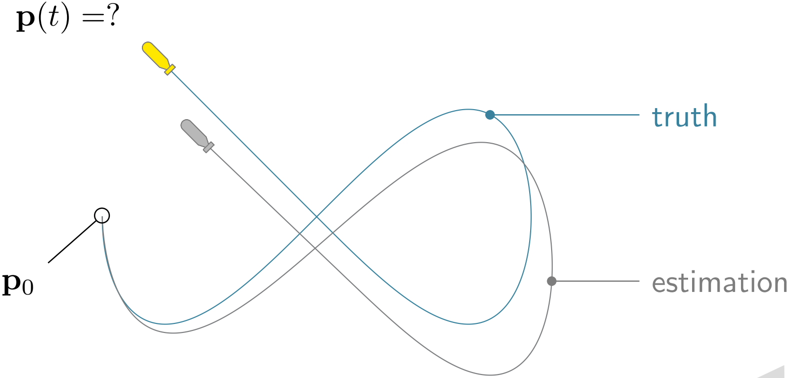
Introduction

Motivations, robot localization: $\mathbf{p}(t) = ?$



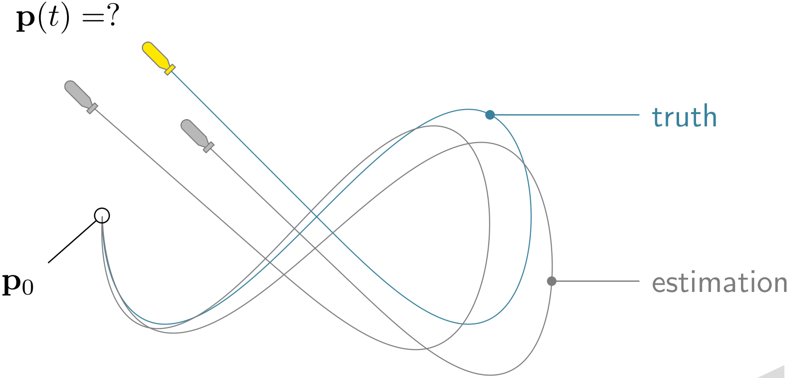
Introduction

Motivations, robot localization: $\mathbf{p}(t) = ?$



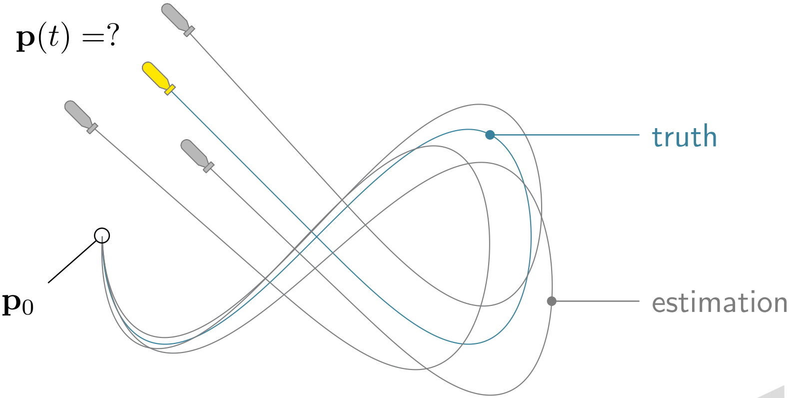
Introduction

Motivations, robot localization: $\mathbf{p}(t) = ?$



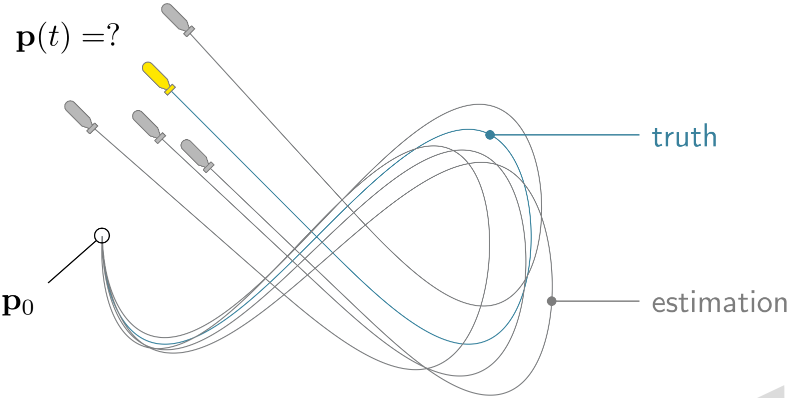
Introduction

Motivations, robot localization: $\mathbf{p}(t) = ?$



Introduction

Motivations, robot localization: $\mathbf{p}(t) = ?$



Motivations, robot localization: $\mathbf{p}(t) = ?$

Exploration solution, **SLAM**:

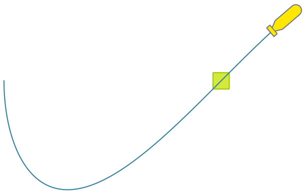
- *Simultaneous Localization and Mapping*
- **come back** to a previous pose and **recognize** the environment

Introduction

Motivations, robot localization: $\mathbf{p}(t) = ?$

Exploration solution, **SLAM**:

- *Simultaneous Localization and Mapping*
- **come back** to a previous pose and **recognize** the environment

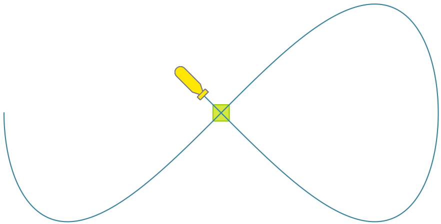


Introduction

Motivations, robot localization: $\mathbf{p}(t) = ?$

Exploration solution, **SLAM**:

- *Simultaneous Localization and Mapping*
- **come back** to a previous pose and **recognize** the environment

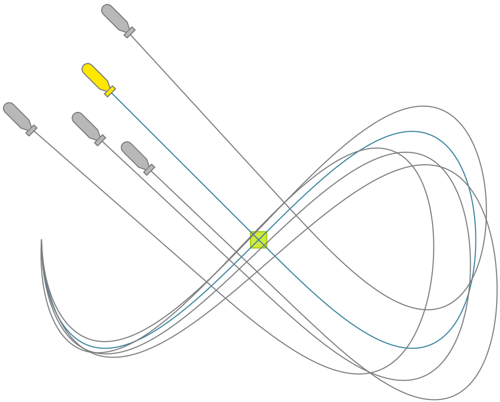


Introduction

Motivations, robot localization: $\mathbf{p}(t) = ?$

Exploration solution, **SLAM**:

- *Simultaneous Localization and Mapping*
- **come back** to a previous pose and **recognize** the environment

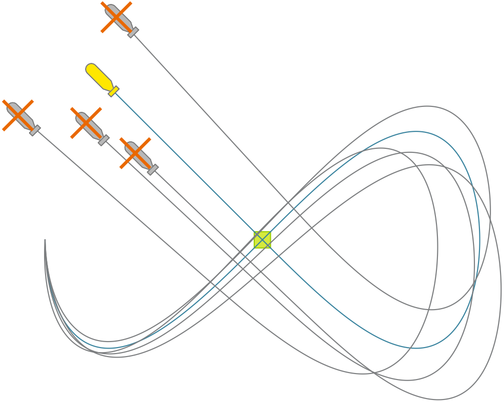


Introduction

Motivations, robot localization: $\mathbf{p}(t) = ?$

Exploration solution, **SLAM**:

- *Simultaneous Localization and Mapping*
- **come back** to a previous pose and **recognize** the environment
- eliminate trajectories **not consistent** with the observation



Introduction

Problem: similar environments (singularities)

What if we recognize a **wrong scene**?

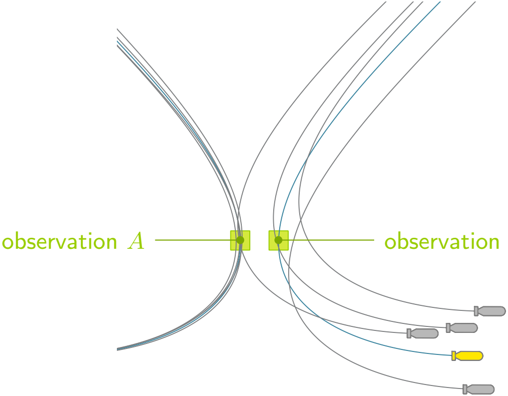
- homogeneous environments $\implies$ similar observations
- strong positioning drift $\implies$ false loop detections

Introduction

Problem: similar environments (singularities)

What if we recognize a **wrong scene**?

- homogeneous environments $\Longrightarrow$ similar observations
- strong positioning drift $\Longrightarrow$ false loop detections

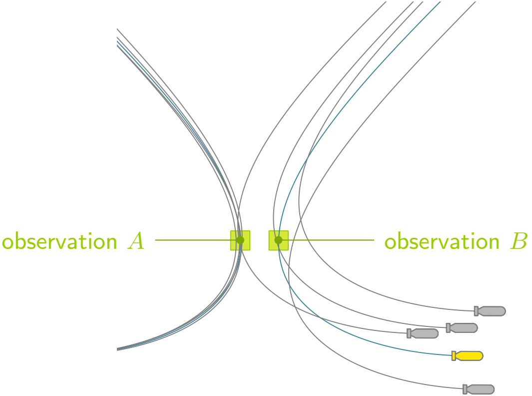


Introduction

Problem: similar environments (singularities)

Need for **loop proof**:

- verify that a trajectory crosses itself at some point
- ..whatever the uncertainties describing this trajectory



Section 2

Formalization

Formalization

Mobile robotics

Robot localization = state estimation problem.
Classically, we have:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) & \text{(navigation)} \\ \end{cases}$$

Where:

- $\mathbf{x} \in \mathbb{R}^n$ is the state vector (position, bearing, ...)
- $\mathbf{u} \in \mathbb{R}^m$ is the input vector (command)
- $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the *evolution* function

Formalization

Mobile robotics

Robot localization = state estimation problem.
Classically, we have:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) & \text{(navigation)} \\ \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) & \text{(measurements)} \end{cases}$$

Where:

- $\mathbf{x} \in \mathbb{R}^n$ is the state vector (position, bearing, ...)
- $\mathbf{u} \in \mathbb{R}^m$ is the input vector (command)
- $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the *evolution* function
- $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^p$ is the *observation* function
- $\mathbf{y} \in \mathbb{R}^p$ is some exteroceptive measurement (camera, sonar...)

Formalization

Mobile robotics

Robot localization = state estimation problem.
Classically, we have:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) & \text{(navigation)} \\ y(t) = g(\mathbf{x}(t)) & \text{(measurements)} \end{cases}$$

Where:

- $\mathbf{x} \in \mathbb{R}^n$ is the state vector (position, bearing, ...)
- $\mathbf{u} \in \mathbb{R}^m$ is the input vector (command)
- $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the *evolution* function
- $g : \mathbb{R}^n \to \mathbb{R}$ is the *observation* function
- $y \in \mathbb{R}$ is some **scalar** measurement (temperature, radioactivity)

Formalization

Set-membership approach

A problem involving constraints is classically presented with a **Constraint Network**:

Variables:

Constraints:

Domains:

Formalization

Set-membership approach

A problem involving constraints is classically presented with a **Constraint Network**:

$$\begin{cases} \textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{u}(\cdot),\ \mathbf{y}(\cdot) \\ \textbf{Constraints:} \\ \\ \textbf{Domains:} \end{cases}$$

Formalization

Set-membership approach

A problem involving constraints is classically presented with a **Constraint Network**:

$$\begin{cases} \textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{u}(\cdot),\ \mathbf{y}(\cdot) \\ \textbf{Constraints:} \\ \\ \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbf{u}](\cdot),\ [\mathbf{y}](\cdot) \end{cases}$$

Formalization

Set-membership approach

A problem involving constraints is classically presented with a **Constraint Network**:

$$\begin{cases} \textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{u}(\cdot),\ \mathbf{y}(\cdot) \\ \textbf{Constraints:} \\ \quad - \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \qquad \text{(evolution equation)} \\ \quad - \quad \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) \qquad \text{(observation equation)} \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbf{u}](\cdot),\ [\mathbf{y}](\cdot) \end{cases}$$

Lab-STICC

Formalization

Set-membership approach

A problem involving constraints is classically presented with a **Constraint Network**:

$$\begin{cases} \textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{u}(\cdot),\ \mathbf{y}(\cdot) \\ \textbf{Constraints:} \\ \quad - \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)) \qquad \text{(evolution equation)} \\ \quad - \quad \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) \qquad \text{(observation equation)} \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbf{u}](\cdot),\ [\mathbf{y}](\cdot) \end{cases}$$

- **domains** can be intervals, boxes, polytopes, tubes...
- **constraints** can be applied over the domains thanks to *contractors*

Formalization

Loop-based localization method

Variables: $\mathbf{x}(\cdot)$, $\mathbf{y}(\cdot)$

Constraints:

- $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$ (evolution)
- $\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t))$ (observation)

Domains: $[\mathbf{x}](\cdot)$, $[\mathbf{y}](\cdot)$

Formalization

Loop-based localization method

$$\begin{cases} \textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{y}(\cdot) \\ \textbf{Constraints:} \\ \quad - \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \qquad \text{(evolution)} \\ \quad - \quad \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) \qquad \text{(observation)} \\ \quad - \quad h(\mathbf{x}(t_1), \mathbf{x}(t_2)) = 0 \implies \mathbf{y}(t_1) = \mathbf{y}(t_2) \qquad \text{(intertemporality)} \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbf{y}](\cdot) \end{cases}$$

Introducing $h : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$:

- **inter-temporal** function
- depicts if two states $\mathbf{x}_1$, $\mathbf{x}_2$ lead to identical observations $\mathbf{y}_1$, $\mathbf{y}_2$

Formalization

Loop-based localization method

$$\begin{cases} \textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{y}(\cdot),\ \mathbb{T}^*_{\mathbf{p}} = \{(t_1,t_2)\} \\ \textbf{Constraints:} \\ \quad - \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \qquad \text{(evolution)} \\ \quad - \quad \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) \qquad \text{(observation)} \\ \quad - \quad h(\mathbf{x}(t_1),\mathbf{x}(t_2)) = 0 \implies \mathbf{y}(t_1) = \mathbf{y}(t_2) \qquad \text{(intertemporality)} \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbf{y}](\cdot),\ \mathbb{T}_{\mathbf{p}} \end{cases}$$

Introducing $h : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$:

- **inter-temporal** function
- depicts if two states $\mathbf{x}_1$, $\mathbf{x}_2$ lead to identical observations $\mathbf{y}_1$, $\mathbf{y}_2$

Temporal resolution:

- a pair (t_1, t_2) becomes variable
- $\{(t_1,t_2)_1,\dots,(t_1,t_2)_q\} = \mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}_{\mathbf{p}} \subset [t_0,t_f]^2$

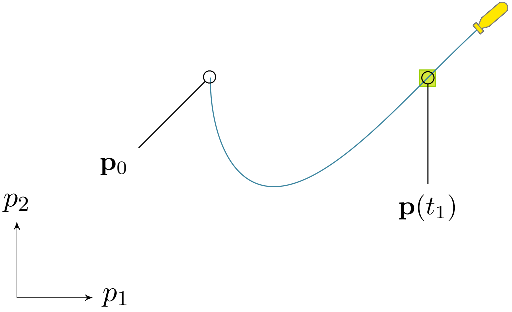
Formalization

Loop-based localization method

$$\begin{cases} \textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{y}(\cdot),\ \mathbb{T}^*_{\mathbf{p}} = \{(t_1,t_2)\} \\ \textbf{Constraints:} \\ \quad - \ \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \\ \quad - \ \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) \\ \quad - \ h(\mathbf{x}(t_1), \mathbf{x}(t_2)) = 0 \implies \mathbf{y}(t_1) = \mathbf{y}(t_2) \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbf{y}](\cdot),\ \mathbb{T}_{\mathbf{p}} \end{cases}$$

Example:

- $\mathbf{x} = (p_1, p_2, \theta)^\intercal \in \mathbb{R}^3$



Formalization

Loop-based localization method

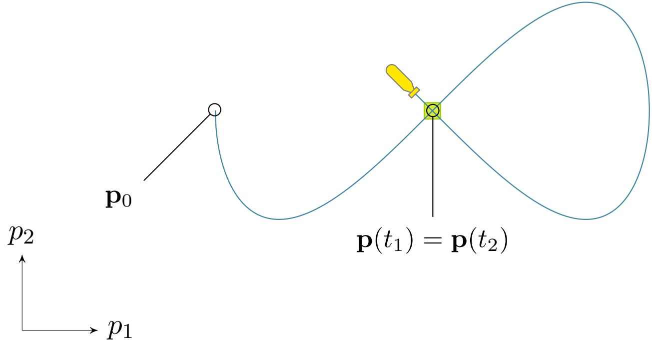
Variables: $\mathbf{x}(\cdot)$, $\mathbf{y}(\cdot)$, $\mathbb{T}^*_{\mathbf{p}} = \{(t_1, t_2)\}$

Constraints:
- $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$
- $\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t))$
- $h(\mathbf{x}(t_1), \mathbf{x}(t_2)) = 0 \implies \mathbf{y}(t_1) = \mathbf{y}(t_2)$

Domains: $[\mathbf{x}](\cdot)$, $[\mathbf{y}](\cdot)$, $\mathbb{T}_{\mathbf{p}}$

Example:

- $\mathbf{x} = (p_1, p_2, \theta)^\intercal \in \mathbb{R}^3$



Formalization

Loop-based localization method

Variables: $\mathbf{x}(\cdot)$, $\mathbf{y}(\cdot)$, $\mathbb{T}^*_{\mathbf{p}} = \{(t_1, t_2)\}$

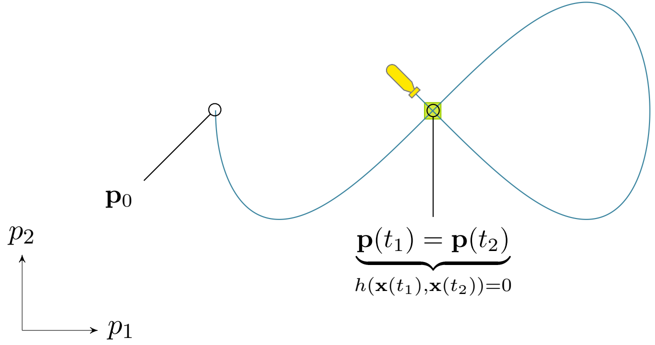
Constraints:

- $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$
- $\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t))$
- $h(\mathbf{x}(t_1), \mathbf{x}(t_2)) = 0 \implies \mathbf{y}(t_1) = \mathbf{y}(t_2)$

Domains: $[\mathbf{x}](\cdot)$, $[\mathbf{y}](\cdot)$, $\mathbb{T}_{\mathbf{p}}$

Example:

- $\mathbf{x} = (p_1, p_2, \theta)^\intercal \in \mathbb{R}^3$
- $h(\mathbf{x}(t_1), \mathbf{x}(t_2)) = \left\| \begin{pmatrix} x_1(t_2) - x_1(t_1) \\ x_2(t_2) - x_2(t_1) \end{pmatrix} \right\|$



Section 3

Loop detections

Loop detections

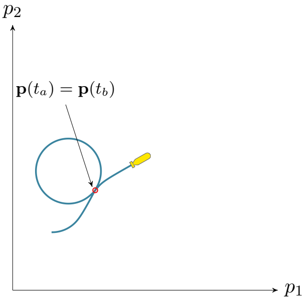
Definitions (Aubry, 2013)

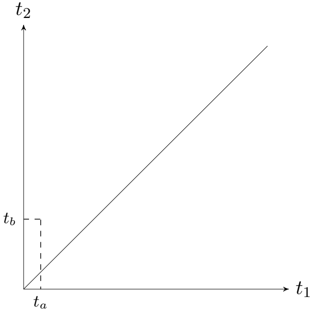
- robot position: $\mathbf{p} = (x, y)^\intercal \in \mathbb{R}^2$
- 2D robot trajectory: $\mathbf{p}(t) : \mathbb{R} \to \mathbb{R}^2$, $t \in [t_0, t_f]$
- looped trajectory $\Leftrightarrow$ trajectory that crosses itself
  - $\mathbf{p}(t_1) = \mathbf{p}(t_2)$, $t_1 \neq t_2$
  - 1 loop $\Leftrightarrow$ 1 t-pair (t_1, t_2)

Loop detections

Definitions (Aubry, 2013)

- robot position: $\mathbf{p} = (x, y)^\intercal \in \mathbb{R}^2$
- 2D robot trajectory: $\mathbf{p}(t) : \mathbb{R} \rightarrow \mathbb{R}^2$, $t \in [t_0, t_f]$
- looped trajectory $\Leftrightarrow$ trajectory that crosses itself
  - $\mathbf{p}(t_1) = \mathbf{p}(t_2)$, $t_1 \neq t_2$
  - 1 loop $\Leftrightarrow$ 1 t-pair (t_1, t_2)

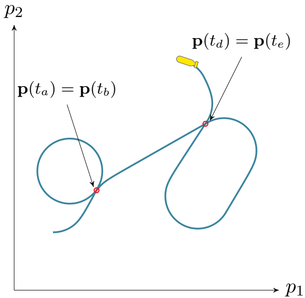


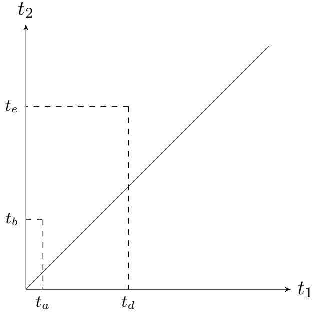


Loop detections

Definitions (Aubry, 2013)

- robot position: $\mathbf{p} = (x, y)^\intercal \in \mathbb{R}^2$
- 2D robot trajectory: $\mathbf{p}(t) : \mathbb{R} \to \mathbb{R}^2$, $t \in [t_0, t_f]$
- looped trajectory $\Leftrightarrow$ trajectory that crosses itself
  - $\mathbf{p}(t_1) = \mathbf{p}(t_2)$, $t_1 \neq t_2$
  - 1 loop $\Leftrightarrow$ 1 t-pair (t_1, t_2)

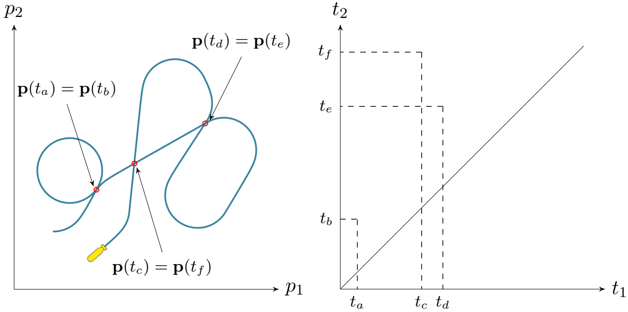




Loop detections

Definitions (Aubry, 2013)

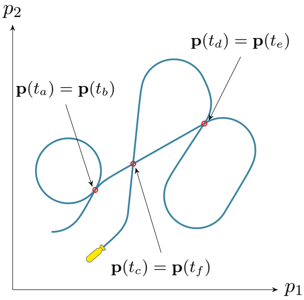
- robot position: $\mathbf{p} = (x, y)^\intercal \in \mathbb{R}^2$
- 2D robot trajectory: $\mathbf{p}(t) : \mathbb{R} \to \mathbb{R}^2$, $t \in [t_0, t_f]$
- looped trajectory $\Leftrightarrow$ trajectory that crosses itself
  - $\mathbf{p}(t_1) = \mathbf{p}(t_2)$, $t_1 \neq t_2$
  - 1 loop $\Leftrightarrow$ 1 t-pair (t_1, t_2)

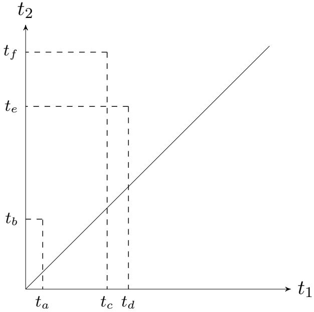


Loop detections

Definitions (Aubry, 2013)

- t-plane $\Leftrightarrow$ all feasible t-pairs $= [t_0, t_f]^2$
- *loop set* $\mathbb{T}^*_{\mathbf{p}}$:
  - $\mathbb{T}^*_{\mathbf{p}} = \left\{(t_1, t_2) \in [t_0, t_f]^2 \mid \mathbf{p}(t_1) = \mathbf{p}(t_2), t_1 < t_2\right\}$
- *loop set* of below example:
  - $\mathbb{T}^*_{\mathbf{p}} = \{(t_a, t_b), (t_c, t_f), (t_d, t_e)\}$





Loop detections

Computing loops from proprioceptive sensors

Context: robot trajectory $\mathbf{p}(t)$ cannot be directly sensed.
Computation from speed measurements:

$$\mathbf{p}(t) = \int_{t_0}^{t} \mathbf{v}(\tau) d\tau + \mathbf{p}_0, \tag{1}$$

with $\mathbf{v}(t) \in \mathbb{R}^2$: robot velocity vector at time $t \in [t_0, t_f]$.

Loop detections

Computing loops from proprioceptive sensors

Context: robot trajectory $\mathbf{p}(t)$ cannot be directly sensed.
Computation from speed measurements:

$$\mathbf{p}(t) = \int_{t_0}^{t} \mathbf{v}(\tau)d\tau + \mathbf{p}_0, \qquad (1)$$

with $\mathbf{v}(t) \in \mathbb{R}^2$: robot velocity vector at time $t \in [t_0, t_f]$.

Loop-set from velocity:

$$\begin{aligned} \mathbb{T}^*_{\mathbf{p}} &= \left\{(t_1,t_2) \in [t_0,t_f]^2 \mid \mathbf{p}(t_1) = \mathbf{p}(t_2), t_1 < t_2\right\} && (2) \\ &= \left\{(t_1,t_2) \in [t_0,t_f]^2 \mid \int_{t_1}^{t_2} \mathbf{v}(\tau)d\tau = \mathbf{0}, t_1 < t_2\right\} && (3) \end{aligned}$$

Loop detection of mobile robots using interval analysis
C. Aubry, R. Desmare, L. Jaulin. *Automatica*, 2013

Loop detections

Dealing with inter-temporalities

1 t-pair $(t_1, t_2) \implies 1$ inter-temporality

$$\left\{\begin{array}{l}
\textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{y}(\cdot),\ \mathbb{T}^*_{\mathbf{p}} = \{(t_1, t_2)\} \\
\textbf{Constraints:} \\
\quad - \ \ \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \\
\quad - \ \ \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) \\
\quad - \ \ h(\mathbf{x}(t_1), \mathbf{x}(t_2)) = 0 \implies \mathbf{y}(t_1) = \mathbf{y}(t_2) \\
\textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbf{y}](\cdot),\ \mathbb{T}_{\mathbf{p}}
\end{array}\right.$$

Loop detections

Dealing with inter-temporalities

$$1 \ t\text{-pair } (t_1, t_2) \implies 1 \text{ inter-temporality}$$

$$\begin{cases} \textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{y}(\cdot),\ \mathbb{T}^*_{\mathbf{p}} = \{(t_1, t_2)\} \\ \textbf{Constraints:} \\ \quad - \ \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \\ \quad - \ \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) \\ \quad - \ h(\mathbf{x}(t_1), \mathbf{x}(t_2)) = 0 \implies \mathbf{y}(t_1) = \mathbf{y}(t_2) \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbf{y}](\cdot),\ \mathbb{T}_{\mathbf{p}} \end{cases}$$

Loop-set from trajectory:

$$\mathbb{T}^*_{\mathbf{p}} = \left\{(t_1, t_2) \in [t_0, t_f]^2 \mid \mathbf{p}(t_1) = \mathbf{p}(t_2)\right\} \tag{4}$$

Loop detections

Dealing with inter-temporalities

1 t-pair $(t_1, t_2) \implies$ 1 inter-temporality

$$\begin{cases} \textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{y}(\cdot),\ \mathbb{T}^*_{\mathbf{p}} = \{(t_1, t_2)\} \\ \textbf{Constraints:} \\ \quad - \ \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \\ \quad - \ \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) \\ \quad - \ h(\mathbf{x}(t_1), \mathbf{x}(t_2)) = 0 \implies \mathbf{y}(t_1) = \mathbf{y}(t_2) \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbf{y}](\cdot),\ \mathbb{T}_{\mathbf{p}} \end{cases}$$

Loop-set from trajectory:

$$\mathbb{T}^*_{\mathbf{p}} = \left\{(t_1, t_2) \in [t_0, t_f]^2 \mid \mathbf{p}(t_1) = \mathbf{p}(t_2)\right\} \tag{4}$$

Inter-temporalities from observations:

$$\mathbb{T}^*_{\mathbf{y}} = \left\{(t_1, t_2) \in [t_0, t_f]^2 \mid \mathbf{y}(t_1) = \mathbf{y}(t_2)\right\} \tag{5}$$

Loop detections

Dealing with inter-temporalities

1 t-pair $(t_1, t_2) \implies$ 1 inter-temporality

$$\left\{\begin{array}{l}\textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{y}(\cdot),\ \mathbb{T}^*_{\mathbf{p}} = \{(t_1,t_2)\} \\ \textbf{Constraints:} \\ \quad - \ \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \\ \quad - \ \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) \\ \quad - \ h(\mathbf{x}(t_1), \mathbf{x}(t_2)) = 0 \implies \mathbf{y}(t_1) = \mathbf{y}(t_2) \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbf{y}](\cdot),\ \mathbb{T}_{\mathbf{p}}\end{array}\right.$$

Loop-set from trajectory:

$$\mathbb{T}^*_{\mathbf{p}} = \left\{(t_1, t_2) \in [t_0, t_f]^2 \mid \mathbf{p}(t_1) = \mathbf{p}(t_2)\right\} \tag{4}$$

Inter-temporalities from observations:

$$\mathbb{T}^*_{\mathbf{y}} = \left\{(t_1, t_2) \in [t_0, t_f]^2 \mid \mathbf{y}(t_1) = \mathbf{y}(t_2)\right\} \tag{5}$$

From constraint (3): $h(\mathbf{x}(t_1), \mathbf{x}(t_2)) = 0 \implies \mathbf{y}(t_1) = \mathbf{y}(t_2)$

$$\mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}^*_{\mathbf{y}} \tag{6}$$

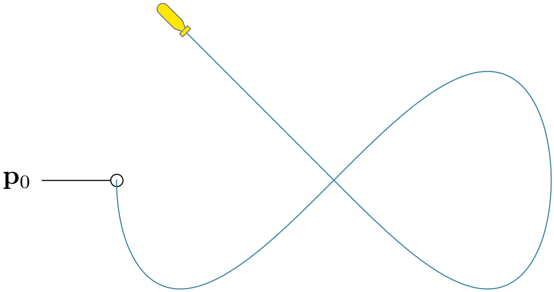
Section 4

Dealing with uncertainties

Dealing with uncertainties

Uncertain trajectories

Actual trajectory: $\mathbf{p}(t) = \int_{t_0}^{t} \mathbf{v}(\tau)d\tau + \mathbf{p}_0$

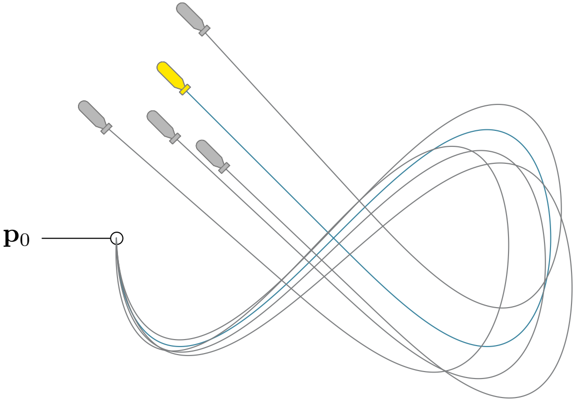


Dealing with uncertainties

Uncertain trajectories

Actual trajectory: $\mathbf{p}(t) = \int_{t_0}^{t} \mathbf{v}(\tau) d\tau + \mathbf{p}_0$

Drifting trajectory: $\mathbf{p}_e(t) = \int_{t_0}^{t} \big(\mathbf{v}(\tau) + \epsilon(\tau)\big) d\tau + \mathbf{p}_0$

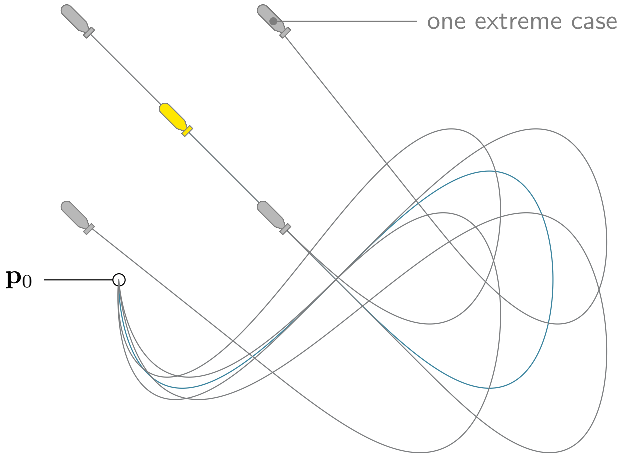


Dealing with uncertainties

Uncertain trajectories

Actual trajectory: $\mathbf{p}(t) = \int_{t_0}^{t} \mathbf{v}(\tau)d\tau + \mathbf{p}_0$

Approach: consider worst cases by defining bounded solutions

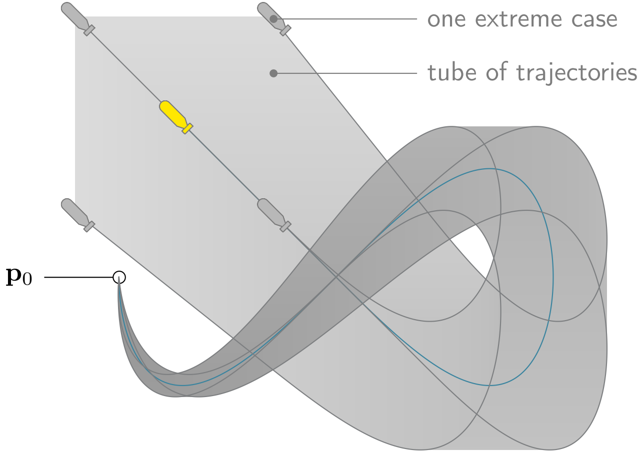


Dealing with uncertainties

Uncertain trajectories

Actual trajectory: $\mathbf{p}(t) = \int_{t_0}^{t} \mathbf{v}(\tau)d\tau + \mathbf{p}_0$

Approach: consider worst cases by defining bounded solutions



Dealing with uncertainties

Interval Analysis

An interval $[x]$:

- a closed and connected subset of $\mathbb{R}$ delimited by two bounds
- $[x^-, x^+] = \{x \in \mathbb{R} \mid x^- \leqslant x \leqslant x^+\}$
- $[x] \in \mathbb{IR}$

A box $[\mathbf{x}]$:

- a cartesian product of n intervals
- $[\mathbf{x}] \in \mathbb{IR}^n$

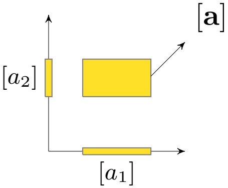


Figure: a box $[\mathbf{a}] \in \mathbb{IR}^2$

Dealing with uncertainties

Interval Analysis

An interval $[x]$:

- a closed and connected subset of $\mathbb{R}$ delimited by two bounds
- $[x^-, x^+] = \{x \in \mathbb{R} \mid x^- \leqslant x \leqslant x^+\}$
- $[x] \in \mathbb{IR}$

A box $[\mathbf{x}]$:

- a cartesian product of n intervals
- $[\mathbf{x}] \in \mathbb{IR}^n$

Notation: actual value denoted x^*, $\mathbf{x}^*$, ...

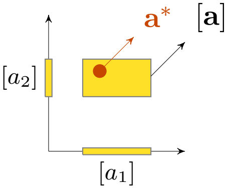


Figure: a box $[\mathbf{a}] \in \mathbb{IR}^2$

Dealing with uncertainties

Interval Analysis

Based on the extension of all classical **real arithmetic operators**:

- $+, -, \times, \div$
- *ex:* $[x]+[y]=[x^-+y^-,x^++y^+]$
- *ex:* $[x]-[y]=[x^--y^+,x^+-y^-]$

Adaptation of **elementary functions** such as:

- cos, exp, tan, etc.
- output is the smallest interval containing all the images of all defined inputs through the function

Dealing with uncertainties

Tubes

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

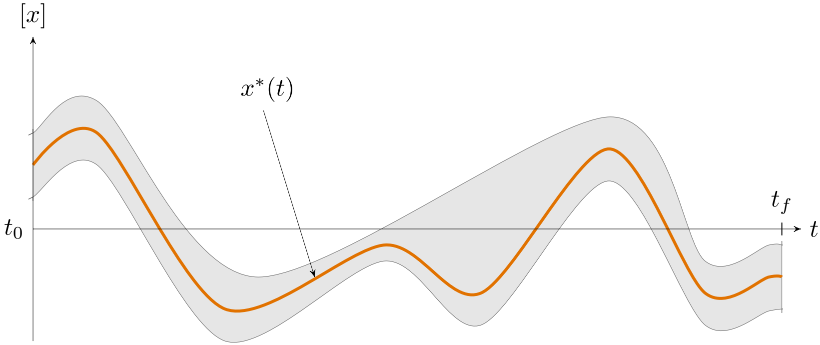


Figure: tube $[x](\cdot)$ enclosing an uncertain trajectory $x^*(\cdot)$

Dealing with uncertainties

Tubes arithmetic

Example:
Tube arithmetic makes it possible to compute the following tubes:

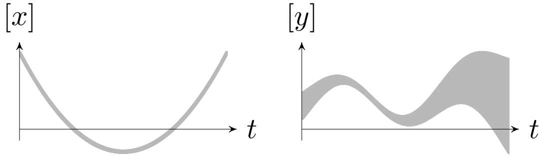
$$\begin{aligned} [a](\cdot) &= [x](\cdot) + [y](\cdot) \\ [b](\cdot) &= \sin\big([x](\cdot)\big) \\ [c](\cdot) &= \int_0 [x](\tau)d\tau \end{aligned}$$

Definition:
If f is an elementary function such as $\sin$, $\cos$, $\dots$, $f\big([x](\cdot)\big)$ is the smallest tube containing all feasible values for $f\big(x(\cdot)\big)$, $x(\cdot) \in [x](\cdot)$.

Lab-STICC

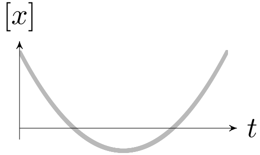
Dealing with uncertainties

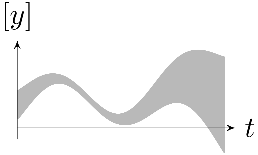
Tubes arithmetic: example

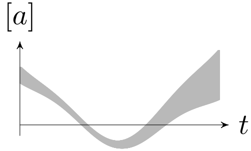


Dealing with uncertainties

Tubes arithmetic: example



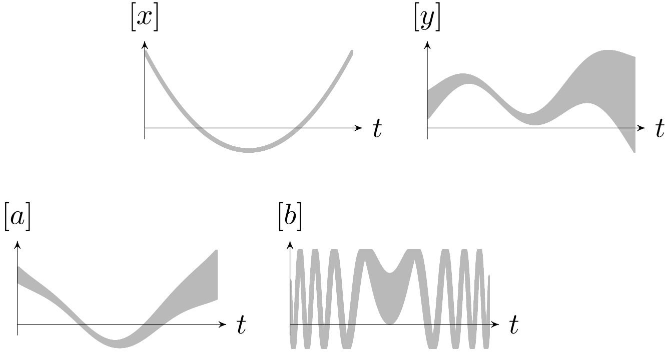




$$a(\cdot) = x(\cdot) + y(\cdot)$$

Dealing with uncertainties

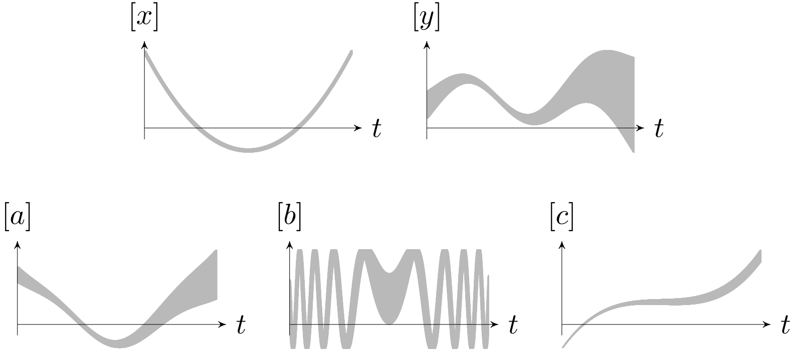
Tubes arithmetic: example



$a(\cdot) = x(\cdot) + y(\cdot)$ $\qquad$ $b(\cdot) = \sin\big(x(\cdot)\big)$

Dealing with uncertainties

Tubes arithmetic: example



$a(\cdot) = x(\cdot) + y(\cdot)$ $\quad$ $b(\cdot) = \sin\big(x(\cdot)\big)$ $\quad$ $c(\cdot) = \int_0 x(\tau)d\tau$

Dealing with uncertainties

Integral of tubes

Definition: the integral of a tube $[x](\cdot) = [x^-, x^+]$ is an interval:

$$\int_a^b [x](\tau)d\tau = \left\{ \int_a^b x(\tau)d\tau \mid x \in [x] \right\} = \left[\int_a^b x^-(\tau)d\tau, \int_a^b x^+(\tau)d\tau \right]$$

[Aubry2013]

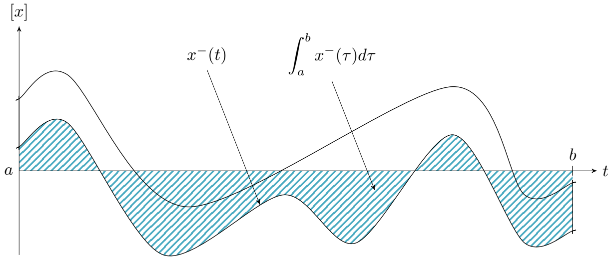


Figure: blue area: lower bound of the tube's integral

Dealing with uncertainties

Integral of tubes

Definition: the integral of a tube $[x](\cdot) = [x^-, x^+]$ is an interval:

$$\int_a^b [x](\tau)d\tau = \left\{ \int_a^b x(\tau)d\tau \mid x \in [x] \right\} = \left[\int_a^b x^-(\tau)d\tau, \int_a^b x^+(\tau)d\tau \right]$$

[Aubry2013]

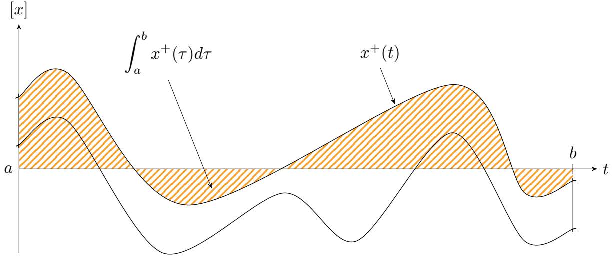


Figure: orange area: upper bound of the tube's integral

Section 5

Reliable localization method

Reliable localization method

Problem statement

$$\begin{cases} \textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{y}(\cdot),\ \mathbb{T}^*_{\mathbf{p}} = \{(t_1,t_2)\} \\ \textbf{Constraints:} \\ \quad - \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \qquad \text{(evolution)} \\ \quad - \quad \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) \qquad \text{(observation)} \\ \quad - \quad h(\mathbf{x}(t_1), \mathbf{x}(t_2)) = 0 \implies \mathbf{y}(t_1) = \mathbf{y}(t_2) \qquad \text{(intertemporality)} \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbf{y}](\cdot),\ \mathbb{T}_{\mathbf{p}} \end{cases}$$

Reliable localization method

Problem statement

$$
\begin{cases}
\textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{y}(\cdot),\ \mathbb{T}^*_{\mathbf{p}} = \{(t_1, t_2)\} \\
\textbf{Constraints:} \\
\quad - \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \qquad \text{(evolution)} \\
\quad - \quad \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) \qquad \text{(observation)} \\
\quad - \quad h(\mathbf{x}(t_1), \mathbf{x}(t_2)) = 0 \implies \mathbf{y}(t_1) = \mathbf{y}(t_2) \qquad \text{(intertemporality)} \\
\textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbf{y}](\cdot),\ \mathbb{T}_{\mathbf{p}}
\end{cases}
$$

- h depicts when a robot performed a loop (t_1, t_2)

Lab-STICC

Reliable localization method

Problem statement

$$\begin{cases} \textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{y}(\cdot),\ \mathbb{T}^*_{\mathbf{p}} = \{(t_1,t_2)\} \\ \textbf{Constraints:} \\ \quad - \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) & \text{(evolution)} \\ \quad - \quad \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) & \text{(observation)} \\ \quad - \quad h(\mathbf{x}(t_1),\mathbf{x}(t_2)) = 0 \implies \mathbf{y}(t_1) = \mathbf{y}(t_2) & \text{(intertemporality)} \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbf{y}](\cdot),\ \mathbb{T}_{\mathbf{p}} \end{cases}$$

- h depicts when a robot performed a loop (t_1, t_2)
- $[\mathbf{x}](\cdot)$, $[\mathbf{y}](\cdot)$ are tubes from measurements

Reliable localization method

Inter-temporalities: reliable approximation

Available information:

- 1 t-plane to detect loops (set $\mathbb{T}^*_{\mathbf{p}}$)
- 1 t-plane to detect identical observations (set $\mathbb{T}^*_{\mathbf{y}}$)

Reliable localization method

Inter-temporalities: contraction

Temporal **contraction**:

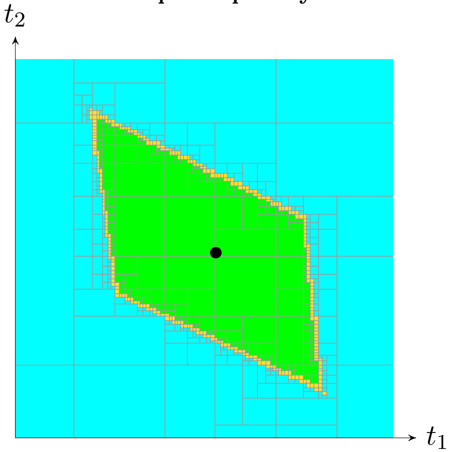
$$\mathbb{T}_{\mathbf{p}} = \mathbb{T}_{\mathbf{p}} \cap \mathbb{T}_{\mathbf{y}} \tag{7}$$

Reliable localization method

Inter-temporalities: contraction

Temporal **contraction**:

$$\mathbb{T}_{\mathbf{p}} = \mathbb{T}_{\mathbf{p}} \cap \mathbb{T}_{\mathbf{y}} \quad (7)$$

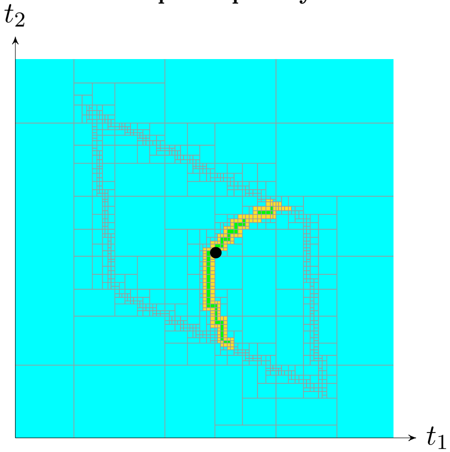


Reliable localization method

Inter-temporalities: contraction

Temporal **contraction**:

$$\mathbb{T}_{\mathbf{p}} = \mathbb{T}_{\mathbf{p}} \cap \mathbb{T}_{\mathbf{y}} \tag{7}$$



Reliable localization method

Back to spatial space

$$\begin{cases} \textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{y}(\cdot),\ \mathbb{T}^*_{\mathbf{p}} = \{(t_1,t_2)\} \\ \textbf{Constraints:} \\ \quad - \ \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \\ \quad - \ \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) \\ \quad - \ h(\mathbf{x}(t_1), \mathbf{x}(t_2)) = 0 \implies \mathbf{y}(t_1) = \mathbf{y}(t_2) \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbf{y}](\cdot),\ \mathbb{T}_{\mathbf{p}} \end{cases}$$

$$\mathbb{T}^*_{\mathbf{p}} = \left\{(t_1,t_2) \mid \underbrace{h(\mathbf{x}(t_1),\mathbf{x}(t_2)) = 0}_{\mathbf{p}(t_1)=\mathbf{p}(t_2)}\right\} \tag{8}$$

Reliable localization method

Back to spatial space

$$\left\{\begin{array}{l}\textbf{Variables: } \mathbf{x}(\cdot),\ \mathbf{y}(\cdot),\ \mathbb{T}^*_{\mathbf{p}} = \{(t_1,t_2)\} \\ \textbf{Constraints:} \\ \quad - \ \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \\ \quad - \ \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) \\ \quad - \ h(\mathbf{x}(t_1), \mathbf{x}(t_2)) = 0 \implies \mathbf{y}(t_1) = \mathbf{y}(t_2) \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbf{y}](\cdot),\ \mathbb{T}_{\mathbf{p}}\end{array}\right.$$

$$\mathbb{T}^*_{\mathbf{p}} = \left\{(t_1, t_2) \mid \underbrace{h(\mathbf{x}(t_1), \mathbf{x}(t_2)) = 0}_{\mathbf{p}(t_1)=\mathbf{p}(t_2)}\right\} \tag{8}$$

Inter-temporal **constraint** with bounded uncertainties:

$$\exists t_1 \in [t_1], \exists t_2 \in [t_2], \exists \mathbf{p}(\cdot) \in [\mathbf{p}](\cdot) \mid \mathbf{p}(t_1) = \mathbf{p}(t_2) \tag{9}$$

Reliable localization method

Dedicated differential tube contractors

$\mathcal{C}_{\text{obs}}$: contraction based on the observation $[\mathbf{p}]([t_2])$ made at time $[t_1]$.

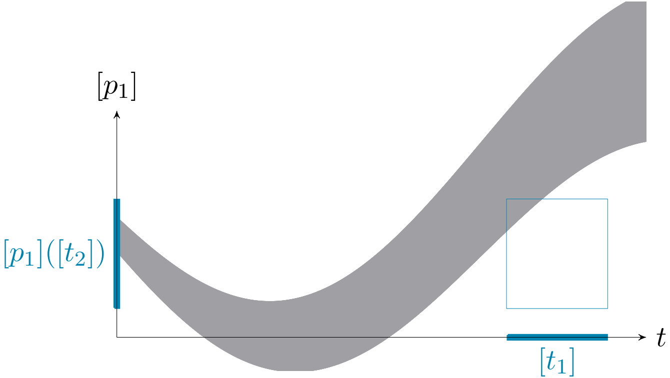


Figure: tube $[p_1](\cdot)$ before contraction

Reliable non-linear state estimation involving time uncertainties
S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, S. M. Veres. *Submitted to Automatica*, 2017

Reliable localization method

Dedicated differential tube contractors

$\mathcal{C}_{\text{obs}}$: contraction based on the observation $[\mathbf{p}]([t_2])$ made at time $[t_1]$.

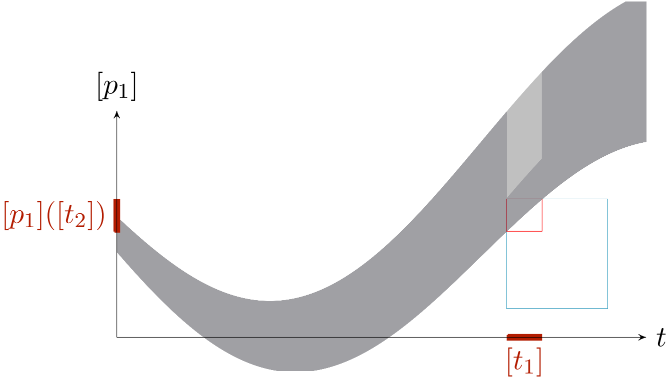


Figure: contraction of tube $[p_1](\cdot)$ and both $[p_1]([t_2])$ and $[t_1]$

Reliable non-linear state estimation involving time uncertainties
S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, S. M. Veres. *Submitted to Automatica*, 2017

Reliable localization method

Dedicated differential tube contractors

$\mathcal{C}_{\text{obs}}$: contraction based on the observation $[\mathbf{p}]([t_2])$ made at time $[t_1]$.

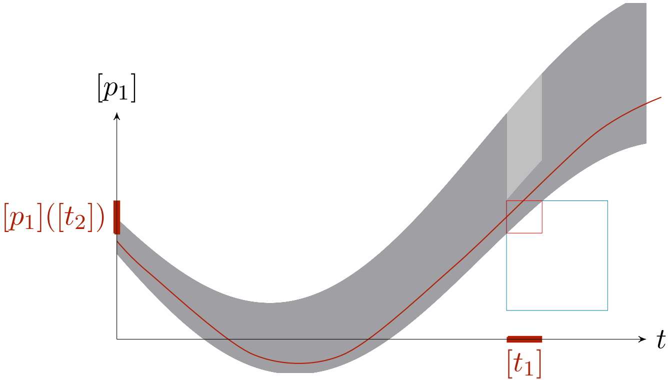


Figure: contraction of tube $[p_1](\cdot)$ and both $[p_1]([t_2])$ and $[t_1]$

Reliable non-linear state estimation involving time uncertainties
S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, S. M. Veres. *Submitted to Automatica*, 2017

Reliable localization method

Dedicated differential tube contractors

$\mathcal{C}_{\frac{d}{dt}}$: contraction based on the differential constraint:

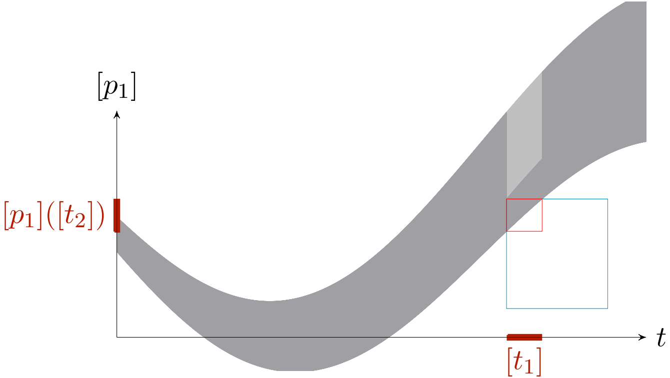


Figure: contraction of tube $[p_1](\cdot)$ and both $[p_1]([t_2])$ and $[t_1]$

Reliable non-linear state estimation involving time uncertainties
S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, S. M. Veres. *Submitted to Automatica*, 2017

Reliable localization method

Dedicated differential tube contractors

$\mathcal{C}_{\frac{d}{dt}}$: contraction based on the differential constraint:

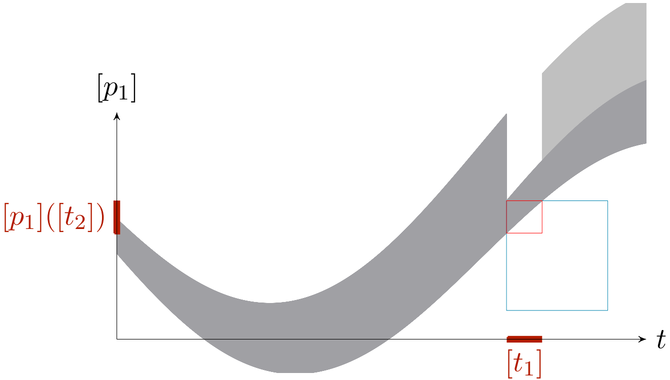


Figure: tube contraction in forward

Reliable non-linear state estimation involving time uncertainties
S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, S. M. Veres. *Submitted to Automatica*, 2017

Reliable localization method

Dedicated differential tube contractors

$\mathcal{C}_{\frac{d}{dt}}$: contraction based on the differential constraint:

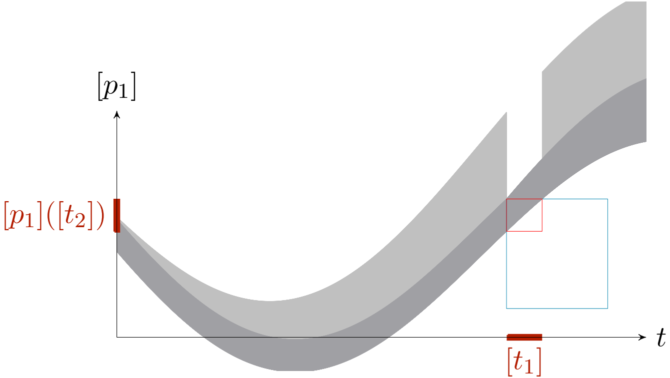


Figure: tube contraction in forward/backward

Reliable non-linear state estimation involving time uncertainties
S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, S. M. Veres. *Submitted to Automatica*, 2017

Section 6

Realistic application

Realistic application

Daurade mission

2 hours experimental mission in Brittany (France)

Figure: The *Daurade* Autonomous Underwater Vehicle (AUV)

Realistic application

Reliable approximation of absolute speed $\mathbf{v}^*(\cdot)$

Robot sensors for **absolute speed computation:**

- velocity sensor (DVL)
- inertial measurement unit

Realistic application

Reliable approximation of absolute speed $\mathbf{v}^*(\cdot)$

Robot sensors for **absolute speed computation:**

- velocity sensor (DVL)
- inertial measurement unit

Uncertainties:

- datasheets $\Longrightarrow$ standard deviation σ for each sensor

Realistic application

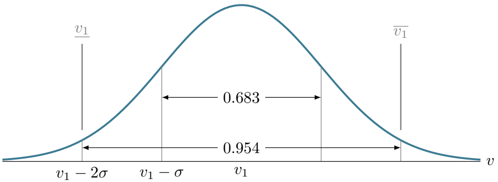
Reliable approximation of absolute speed $\mathbf{v}^*(\cdot)$

Robot sensors for **absolute speed computation:**

- velocity sensor (DVL)
- inertial measurement unit

Uncertainties:

- datasheets $\Longrightarrow$ standard deviation σ for each sensor
- 95% confidence rate: $v_1^* \in [v_1] = [v_1 - 2\sigma, v_1 + 2\sigma]$



Realistic application

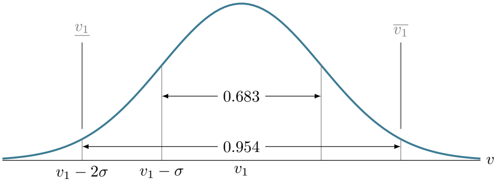
Reliable approximation of absolute speed $\mathbf{v}^*(\cdot)$

Robot sensors for **absolute speed computation:**

- velocity sensor (DVL)
- inertial measurement unit

Uncertainties:

- datasheets $\Longrightarrow$ standard deviation σ for each sensor
- 95% confidence rate: $v_1^* \in [v_1] = [v_1 - 2\sigma, v_1 + 2\sigma]$



- uncertainties propagated thanks to interval arithmetic

Realistic application

Reliable approximation of absolute speed $\mathbf{v}^*(\cdot)$

Obtained tube $[\mathbf{v}](\cdot)$:

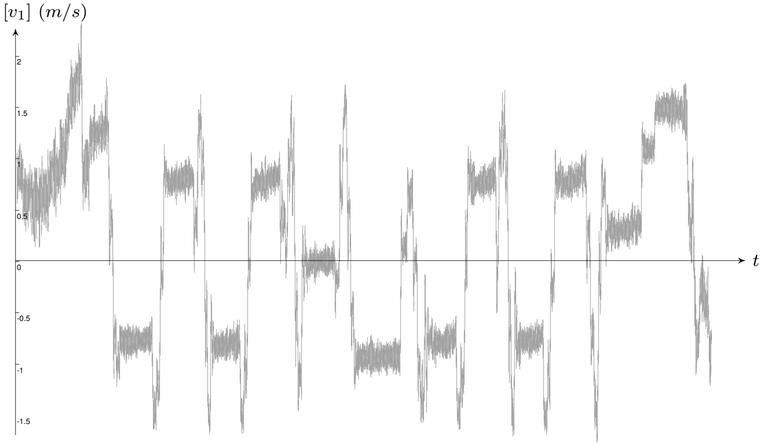


Figure: East speed velocity tube $[v_1](\cdot)$

Realistic application

Reliable approximation of absolute speed $\mathbf{v}^*(\cdot)$

Results...

Support:

DGA
Direction Générale de l'Armement

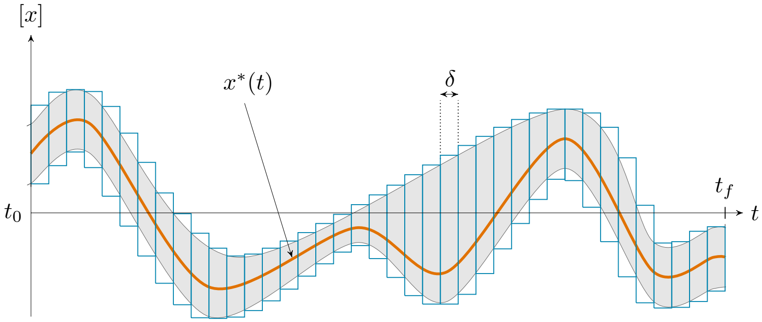
Tools:

IBEX library
used for interval arithmetic, contractor programming

VIBES
used for rendering

Tubex library

An open-source C++ library providing tools to guarantee computations over sets of trajectories.



http://www.simon-rohou.fr/research/tubex-lib/

References:

■ Loop detection of mobile robots using interval analysis
C. Aubry, R. Desmare, L. Jaulin. *Automatica*, 2013

■ Guaranteed computation of robot trajectories
S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, S. M. Veres. *Robotics and Autonomous Systems*, 2017

■ Reliable non-linear state estimation involving time uncertainties
S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, S. M. Veres. *Submitted to Automatica*, 2017

■ Proving the existence of loops in robot trajectories
S. Rohou, P. Franek, C. Aubry, L. Jaulin. *Submitted to IEEE Transactions on Robotics*, 2017