Reliable robot localization: a constraint programming approach over dynamical systems

Simon Rohou

PhD advisors: Luc Jaulin, Lyudmila Mihaylova, Fabrice Le Bars, Sandor M. Veres

ENSTA Bretagne, Lab-STICC, UMR CNRS 6285, Brest, France
The University of Sheffield, Sheffield, UK

French robotics workshop
22^{nd} November 2018

A temporal approach for the SLAM problem

Simon Rohou

PhD advisors: Luc Jaulin, Lyudmila Mihaylova, Fabrice Le Bars, Sandor M. Veres

ENSTA Bretagne, Lab-STICC, UMR CNRS 6285, Brest, France
The University of Sheffield, Sheffield, UK

French robotics workshop
22nd November 2018

Section 1

Motivations

Motivations

Robot localization: $\mathbf{p}(t) = ?$

Underwater exploration **without surfacing**:

- reasons of discretion and security (military missions)
- case of very deep environments (wrecks search)

Titanic wreck: 3821m deep

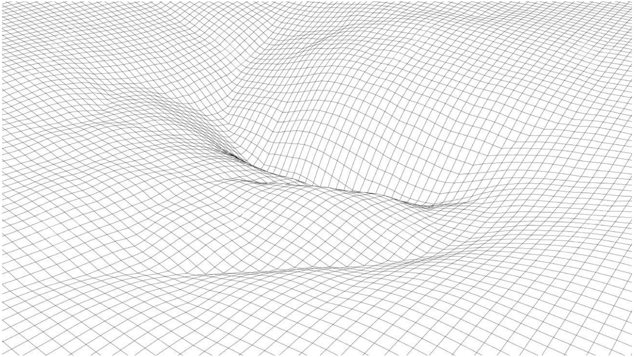
Lost MH370 aircraft: up to 6000m deep

Motivations

Problem: homogeneous environments

Under the surface:

- **no seamarks** or points of interest
- usual SLAM methods do not apply

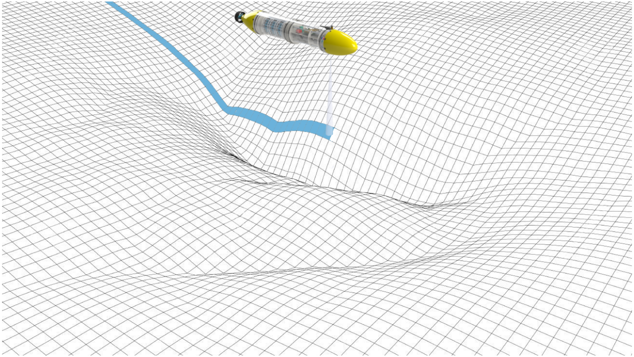


Motivations

Problem: homogeneous environments

Available data:

- **bathymetric** measurements (scalar values)
- using **raw-data SLAM** methods? poor measurements...

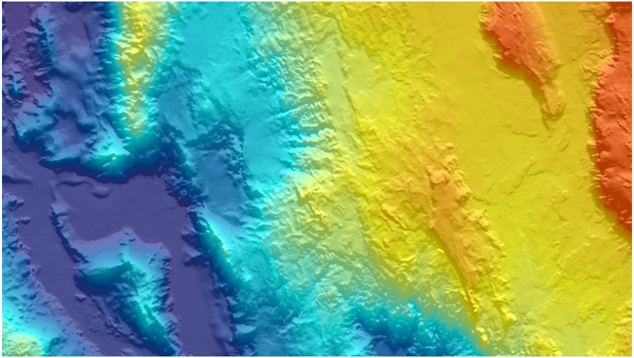


Motivations

Problem: homogeneous environments

Available data:

- **bathymetric** measurements (scalar values)
- using **raw-data SLAM** methods? poor measurements...



Looking for MH370 – © 2014, Commonwealth of Australia

Section 2

Formalization

Formalization

Mobile robotics

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

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

With:

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

Formalization

Mobile robotics

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

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

With:

- $\mathbf{x}$: state vector (position, bearing, ...)
- $\mathbf{u}$: input vector (command)
- $\mathbf{f}$: *evolution* function
- $\mathbf{g}$: *observation* function
- $\mathbf{z}$: some exteroceptive measurement (camera, sonar...)

Formalization

Mobile robotics

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

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

With:

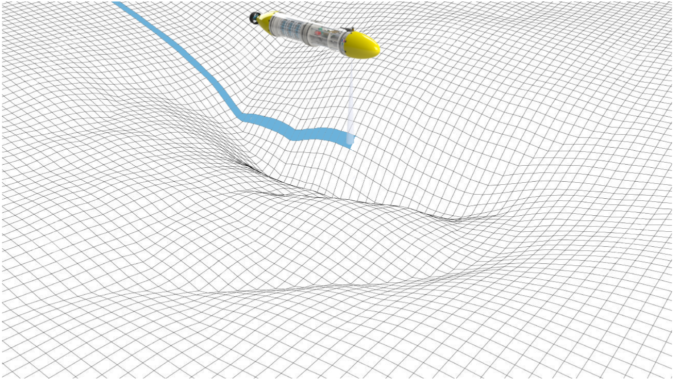
- $\mathbf{x}$: state vector (position, bearing, ...)
- $\mathbf{u}$: input vector (command)
- $\mathbf{f}$: *evolution* function
- $\mathbf{g}$: *observation* function
- $\mathbf{z}$: some exteroceptive measurement (camera, sonar...)

Formalization

Bathymetric localization: observation function g not at hand

Observation equation:

- $z(t) = g(\mathbf{x}(t))$
- expression of g unknown $\implies$ no relation between z and $\mathbf{x}$

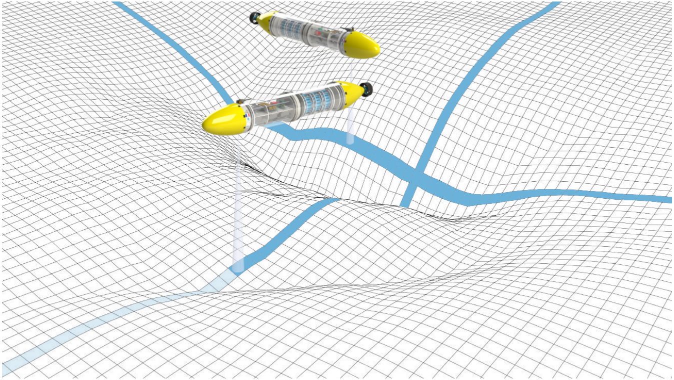


Formalization

Bathymetric localization: observation function g not at hand

Observation equation:

- $z(t) = g(\mathbf{x}(t))$
- expression of g unknown $\implies$ no relation between z and $\mathbf{x}$
- main approach: **inter-temporal measurements**



Formalization

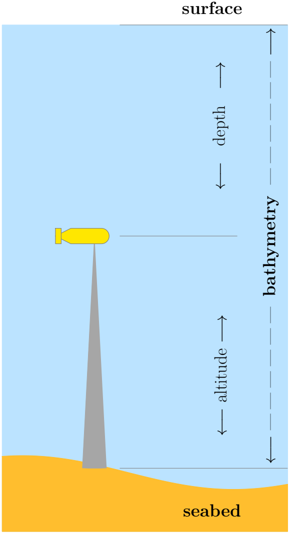
New SLAM formalism: inter-temporal measurements

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \\ \cancel{z(t) = g(\mathbf{x}(t))} \\ \\ \\ \end{cases}$$

Formalization

New SLAM formalism: inter-temporal measurements

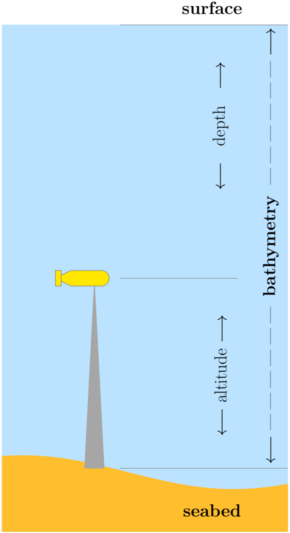
$$\left\{\begin{array}{l}\dot{\mathbf{x}}(t)=\mathbf{f}(\mathbf{x}(t))\\ \cancel{z(t)=g(\mathbf{x}(t))}\\ \\ \\ \end{array}\right.$$



Formalization

New SLAM formalism: inter-temporal measurements

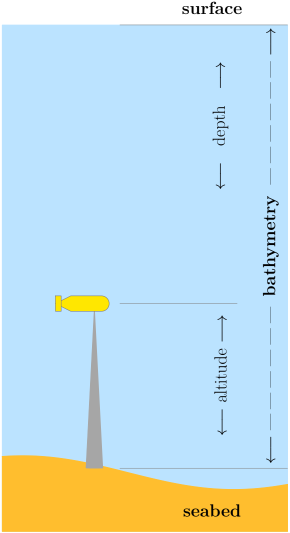
$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \\ \cancel{z(t) = g(\mathbf{x}(t))} \end{cases}$$



Formalization

New SLAM formalism: inter-temporal measurements

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \\ \cancel{z(t) = g(\mathbf{x}(t))} \\ \\ \end{cases}$$

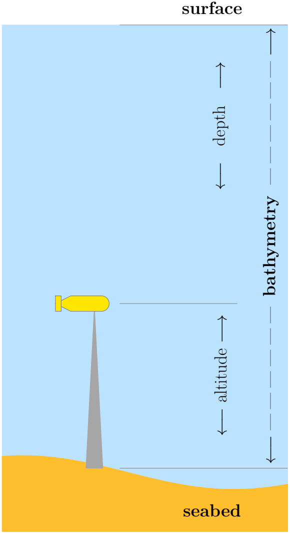


Formalization

New SLAM formalism: inter-temporal measurements

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \\ \cancel{z(t) = g(\mathbf{x}(t))} \\ \underbrace{\mathbf{p}(t_1) = \mathbf{p}(t_2)}_{\text{same } \textbf{h-positions}} \implies \underbrace{z(t_1) = z(t_2)}_{\text{same } \textbf{bathymetry}} \end{cases}$$

Horizontal position vector $\mathbf{p} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$



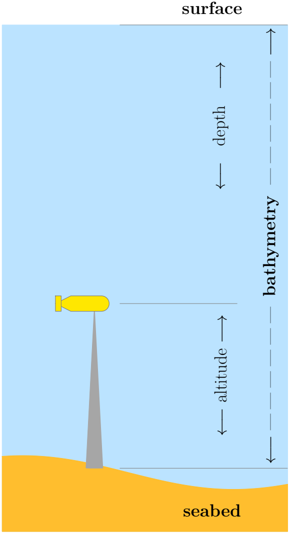
Formalization

New SLAM formalism: inter-temporal measurements

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \\ \cancel{z(t) = g(\mathbf{x}(t))} \\ \underbrace{\mathbf{p}(t_1) = \mathbf{p}(t_2)}_{\text{same } \textbf{h-positions}} \implies \underbrace{z(t_1) = z(t_2)}_{\text{same } \textbf{bathymetry}} \end{cases}$$

Horizontal position vector $\mathbf{p} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

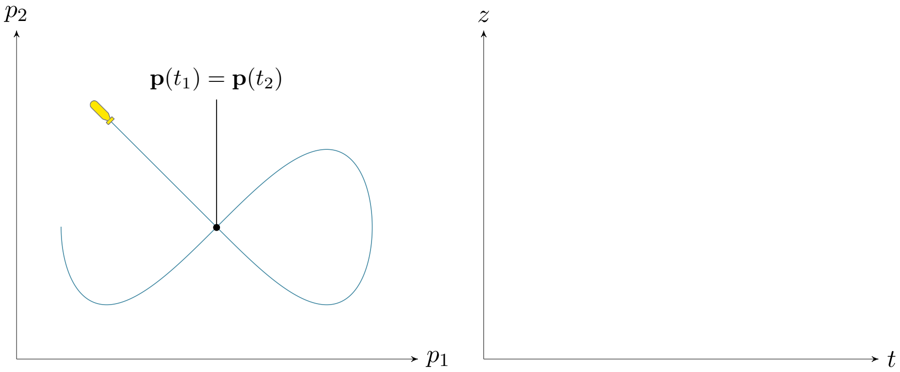
How to deal with these **inter-temporal** equations?



Formalization

Naive inter-temporal resolution

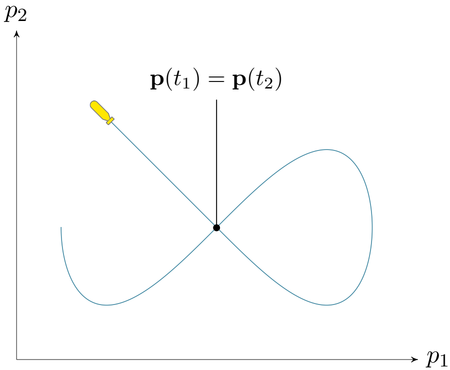
A robot coming back to a previous position $\mathbf{p}$ should sense the same observation z.

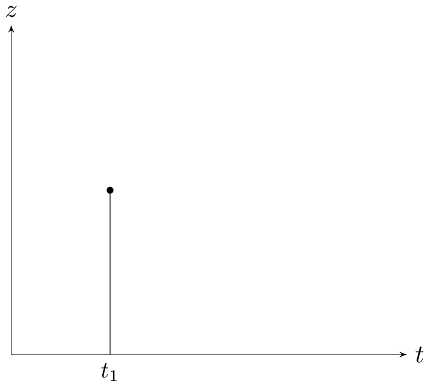


Formalization

Naive inter-temporal resolution

A robot coming back to a previous position $\mathbf{p}$ should sense the same observation z.

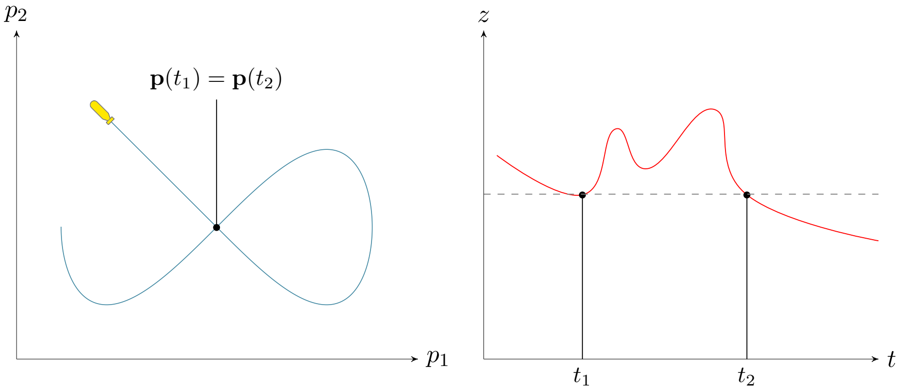




Formalization

Naive inter-temporal resolution

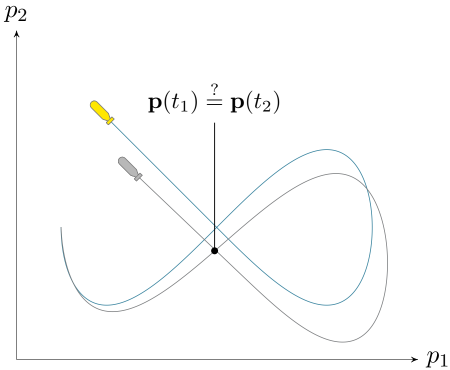
A robot coming back to a previous position $\mathbf{p}$ should sense the same observation z.

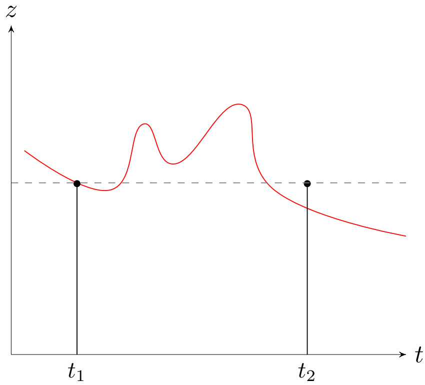


Formalization

Naive inter-temporal resolution

A robot coming back to a previous position $\mathbf{p}$
should sense the same observation z.

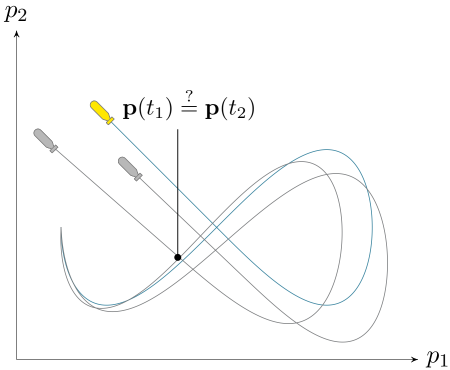


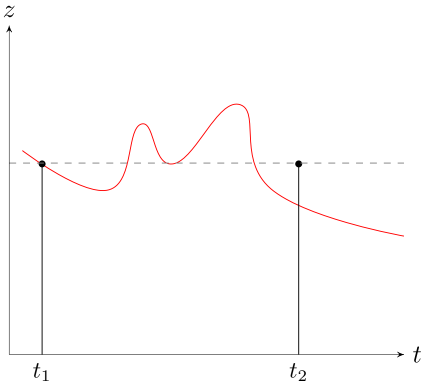


Formalization

Naive inter-temporal resolution

A robot coming back to a previous position $\mathbf{p}$ should sense the same observation z.

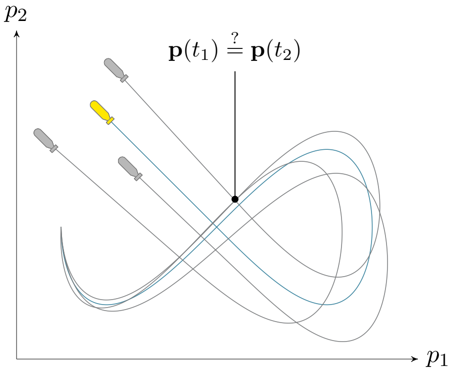


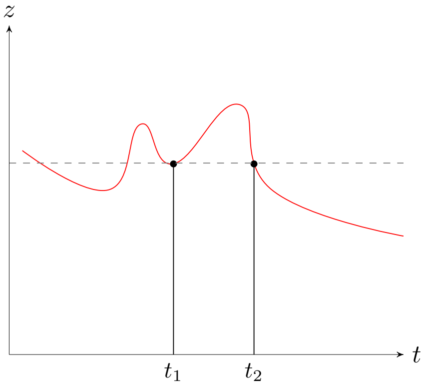


Formalization

Naive inter-temporal resolution

A robot coming back to a previous position $\mathbf{p}$ should sense the same observation z.

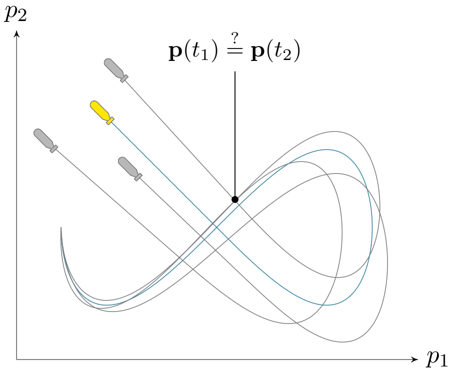


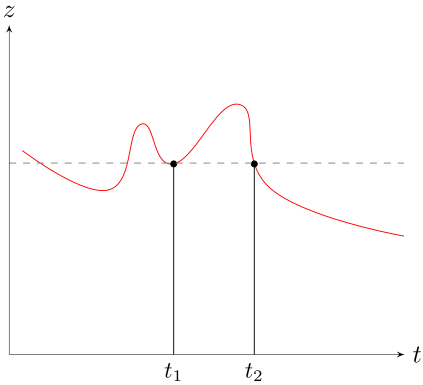


Formalization

Naive inter-temporal resolution

A robot coming back to a previous position $\mathbf{p}$ should sense the same observation z.





Method: temporal resolution, estimation of feasible pairs (t_1, t_2)

Section 3

Temporal resolution

Temporal resolution

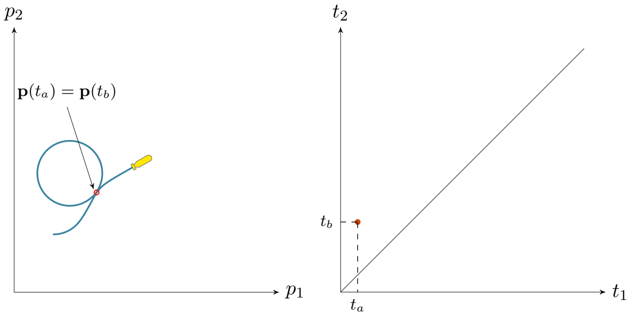
Loops: definitions (Aubry, 2013)

- 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)

Temporal resolution

Loops: definitions (Aubry, 2013)

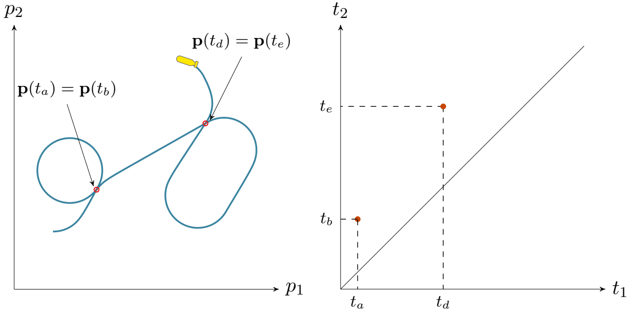
- 2D robot trajectory: $\mathbf{p}(t) : \mathbb{R} \to \mathbb{R}^2$, $t \in [t_0, t_f]$
- looped trajectory $\Leftrightarrow$ trajectory that crosses itself
  - $\mathbf{p}(t_1) = \mathbf{p}(t_2)$, $t_1 \neq t_2$
  - 1 loop $\Leftrightarrow$ 1 t-pair (t_1, t_2)



Temporal resolution

Loops: definitions (Aubry, 2013)

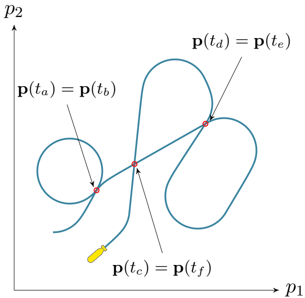
- 2D robot trajectory: $\mathbf{p}(t) : \mathbb{R} \rightarrow \mathbb{R}^2$, $t \in [t_0, t_f]$
- looped trajectory $\Leftrightarrow$ trajectory that crosses itself
  - $\mathbf{p}(t_1) = \mathbf{p}(t_2)$, $t_1 \neq t_2$
  - 1 loop $\Leftrightarrow$ 1 t-pair (t_1, t_2)

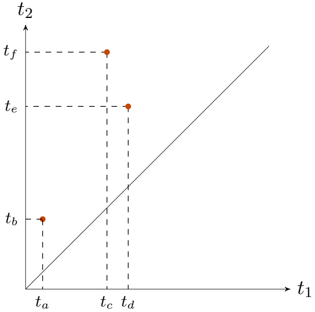


Temporal resolution

Loops: definitions (Aubry, 2013)

- 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)

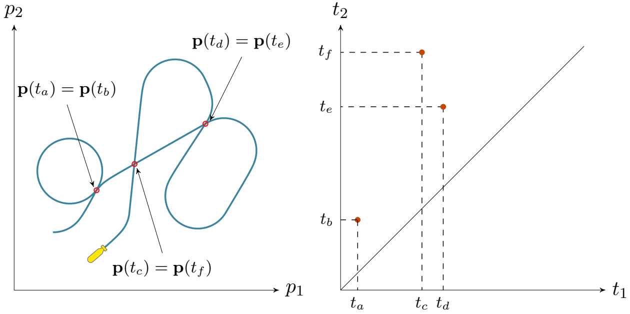




Temporal resolution

Loops: definitions (Aubry, 2013)

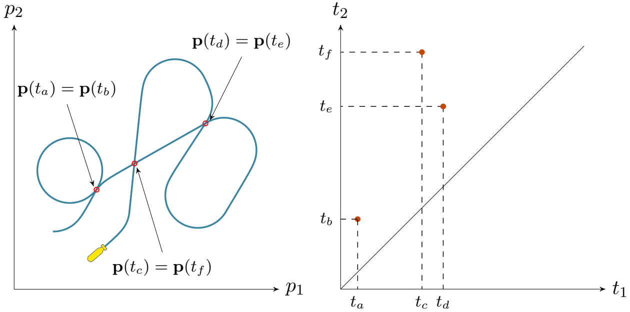
- 2D robot trajectory: $\mathbf{p}(t) : \mathbb{R} \to \mathbb{R}^2$, $t \in [t_0, t_f]$
- looped trajectory $\Leftrightarrow$ trajectory that crosses itself
  - $\mathbf{p}(t_1) = \mathbf{p}(t_2)$, $t_1 \neq t_2$
  - 1 loop $\Leftrightarrow$ 1 t-pair (t_1, t_2)
- t-plane $\Leftrightarrow$ all feasible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



Temporal resolution

Loops: definitions (Aubry, 2013)

- *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:
  - $\mathbb{T}^*_{\mathbf{p}} = \{(t_a, t_b), (t_c, t_f), (t_d, t_e)\}$



Temporal resolution

Loop set: approximation from sensors

$$\mathbb{T}^*_{\mathbf{p}} = \{(t_1, t_2) \mid \mathbf{p}(t_1) = \mathbf{p}(t_2)\}$$

Trajectory $\mathbf{p}(\cdot)$ unknown, but measurements $\mathbf{v}(\cdot)$, $z(\cdot)$ available:

Temporal resolution

Loop set: approximation from sensors

$$\mathbb{T}^*_{\mathbf{p}} = \{(t_1, t_2) \mid \mathbf{p}(t_1) = \mathbf{p}(t_2)\}$$

Trajectory $\mathbf{p}(\cdot)$ unknown, but measurements $\mathbf{v}(\cdot)$, $z(\cdot)$ available:

Proprioceptive sensors
(velocities $\mathbf{v} \in \mathbb{R}^2$)

$$\mathbb{T}^*_{\mathbf{p}} = \left\{(t_1, t_2) \mid \int_{t_1}^{t_2} \mathbf{v}(\tau) d\tau = \mathbf{0}\right\}$$

Exteroceptive sensors
(bathymetry $z \in \mathbb{R}$)

$$\mathbb{T}^*_{z} = \left\{(t_1, t_2) \mid z(t_1) = z(t_2)\right\}$$

Temporal resolution

Loop set: approximation from sensors

$$\mathbb{T}^*_{\mathbf{p}} = \{(t_1, t_2) \mid \mathbf{p}(t_1) = \mathbf{p}(t_2)\}$$

Trajectory $\mathbf{p}(\cdot)$ unknown, but measurements $\mathbf{v}(\cdot)$, $z(\cdot)$ available:

Proprioceptive sensors
(velocities $\mathbf{v} \in \mathbb{R}^2$)

$$\mathbb{T}^*_{\mathbf{p}} = \left\{(t_1, t_2) \mid \int_{t_1}^{t_2} \mathbf{v}(\tau)d\tau = \mathbf{0}\right\}$$

Exteroceptive sensors
(bathymetry $z \in \mathbb{R}$)

$$\mathbb{T}^*_{z} = \left\{(t_1, t_2) \mid z(t_1) = z(t_2)\right\}$$

Inter-temporal implication:

$$\Big(\mathbf{p}(t_1) = \mathbf{p}(t_2) \implies z(t_1) = z(t_2)\Big) \implies \Big(\mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}^*_{z}\Big)$$

Temporal resolution

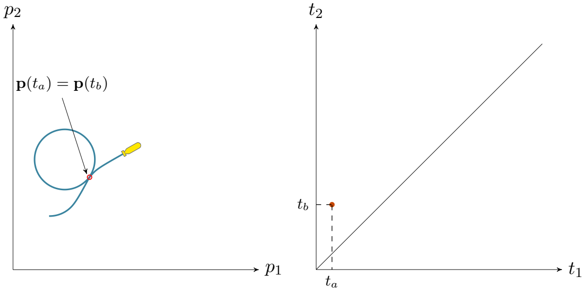
Graphical interpretation

$$\begin{cases} \mathbb{T}^*_{\mathbf{p}} = \left\{ (t_1, t_2) \mid \int_{t_1}^{t_2} \mathbf{v}(\tau) d\tau = \mathbf{0} \right\} \\ \mathbb{T}^*_{z} = \left\{ (t_1, t_2) \mid z(t_1) = z(t_2) \right\} \\ \mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}^*_{z} \end{cases}$$

Temporal resolution

Graphical interpretation

$$
\begin{cases}
\mathbb{T}^*_{\mathbf{p}} = \left\{ (t_1, t_2) \mid \int_{t_1}^{t_2} \mathbf{v}(\tau) d\tau = \mathbf{0} \right\} \\
\mathbb{T}^*_{z} = \left\{ (t_1, t_2) \mid z(t_1) = z(t_2) \right\} \\
\mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}^*_{z}
\end{cases}
$$

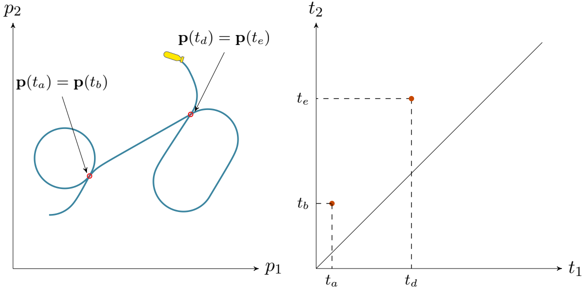


$\mathbb{T}^*_{\mathbf{p}}$

Temporal resolution

Graphical interpretation

$$\begin{cases} \mathbb{T}^*_{\mathbf{p}} = \left\{ (t_1, t_2) \mid \int_{t_1}^{t_2} \mathbf{v}(\tau) d\tau = \mathbf{0} \right\} \\ \mathbb{T}^*_{z} = \left\{ (t_1, t_2) \mid z(t_1) = z(t_2) \right\} \\ \mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}^*_{z} \end{cases}$$

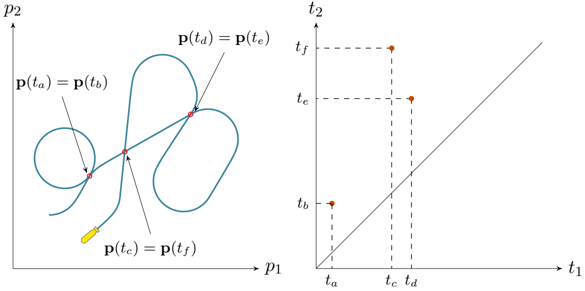


$\mathbb{T}^*_{\mathbf{p}}$

Temporal resolution

Graphical interpretation

$$\begin{cases} \mathbb{T}^*_{\mathbf{p}} = \left\{(t_1,t_2) \mid \int_{t_1}^{t_2} \mathbf{v}(\tau)d\tau = \mathbf{0}\right\} \\ \mathbb{T}^*_{z} = \left\{(t_1,t_2) \mid z(t_1) = z(t_2)\right\} \\ \mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}^*_{z} \end{cases}$$

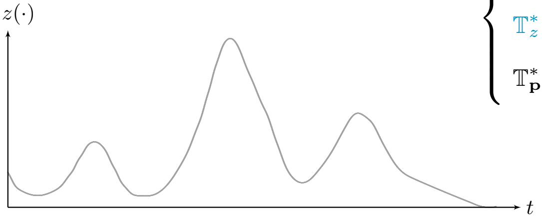


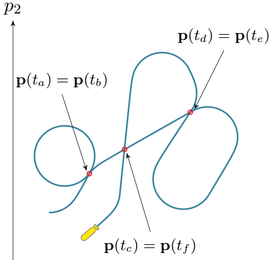
$\mathbb{T}^*_{\mathbf{p}}$

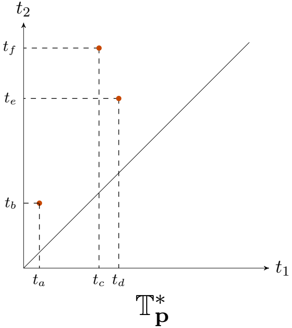
Temporal resolution

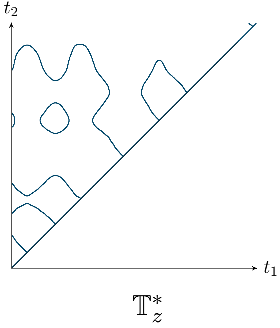
Graphical interpretation

$$
\begin{cases}
\mathbb{T}^*_{\mathbf{p}} = \left\{ (t_1, t_2) \mid \int_{t_1}^{t_2} \mathbf{v}(\tau) d\tau = \mathbf{0} \right\} \\
\mathbb{T}^*_{z} = \left\{ (t_1, t_2) \mid z(t_1) = z(t_2) \right\} \\
\mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}^*_{z}
\end{cases}
$$





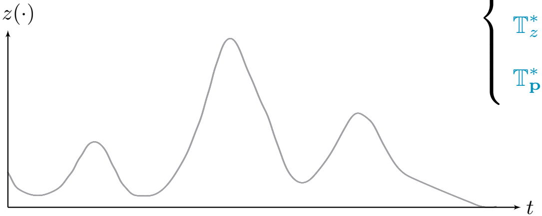


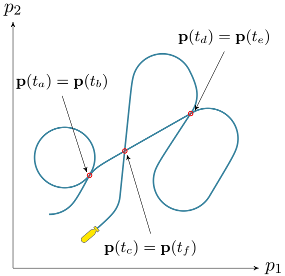


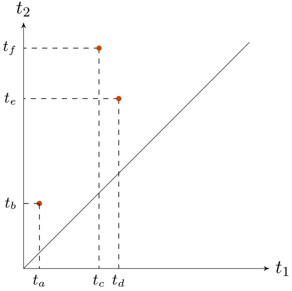
Temporal resolution

Graphical interpretation

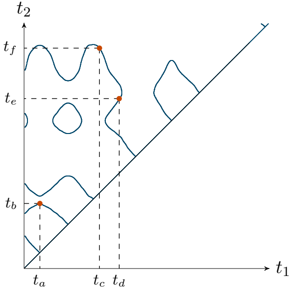
$$\begin{cases} \mathbb{T}^*_{\mathbf{p}} = \left\{ (t_1, t_2) \mid \int_{t_1}^{t_2} \mathbf{v}(\tau) d\tau = \mathbf{0} \right\} \\ \mathbb{T}^*_{z} = \left\{ (t_1, t_2) \mid z(t_1) = z(t_2) \right\} \\ \mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}^*_{z} \end{cases}$$







$\mathbb{T}^*_{\mathbf{p}}$

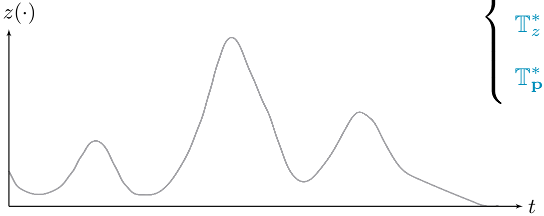


$\mathbb{T}^*_{z}$

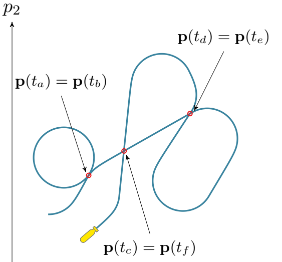
Temporal resolution

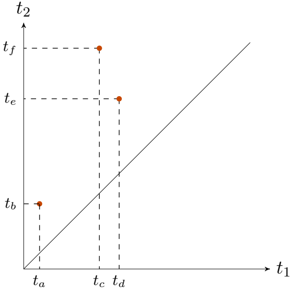
Graphical interpretation

$$
\begin{cases}
\mathbb{T}^*_{\mathbf{p}} = \left\{ (t_1, t_2) \mid \int_{t_1}^{t_2} \mathbf{v}(\tau) d\tau = \mathbf{0} \right\} \\
\mathbb{T}^*_{z} = \left\{ (t_1, t_2) \mid z(t_1) = z(t_2) \right\} \\
\mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}^*_{z}
\end{cases}
$$

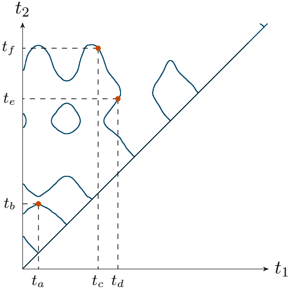


What about uncertainties?





$\mathbb{T}^*_{\mathbf{p}}$

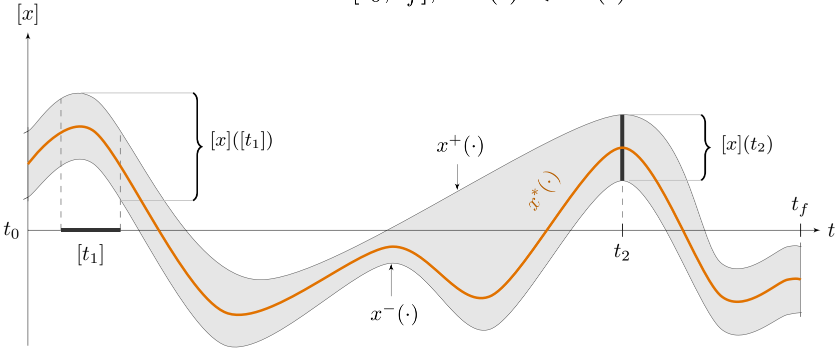


$\mathbb{T}^*_{z}$

Temporal resolution

Tubes enclosing uncertain trajectories

Tube $[x](\cdot)$: interval of trajectories $[x^-(\cdot), x^+(\cdot)]$ such that $\forall t \in [t_0, t_f],\ x^-(t) \leqslant x^+(t)$

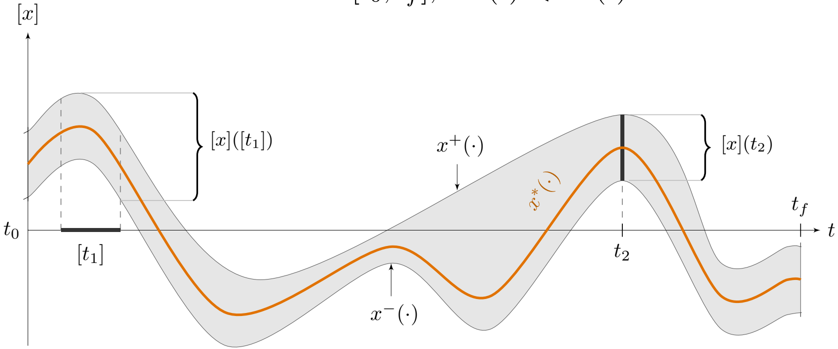


Tube $[x](\cdot)$ enclosing an uncertain trajectory $x^*(\cdot)$

Temporal resolution

Tubes enclosing uncertain trajectories

Tube $[x](\cdot)$**:** interval of trajectories $[x^-(\cdot), x^+(\cdot)]$
such that $\forall t \in [t_0, t_f],\ x^-(t) \leqslant x^+(t)$



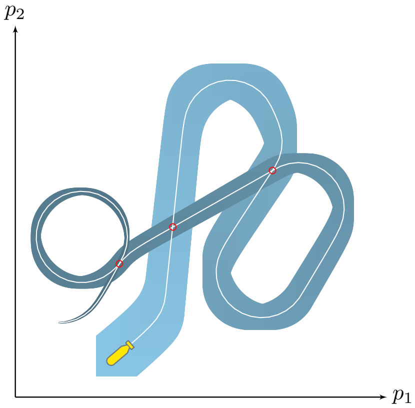
Tube $[x](\cdot)$ enclosing an uncertain trajectory $x^*(\cdot)$

Set-membership approach:
$x^*(\cdot) \in [x](\cdot)$, computations on bounds $\Rightarrow$ guaranteed outputs

Temporal resolution

Bounded-error context

Uncertain trajectories enclosed in tubes.

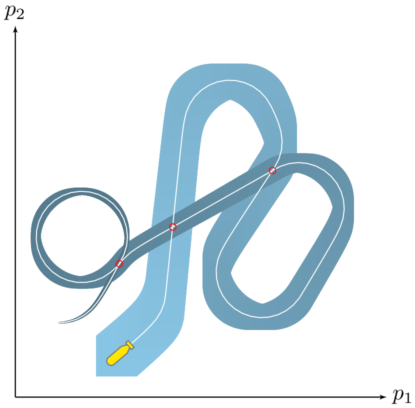


(a) Bounded trajectories

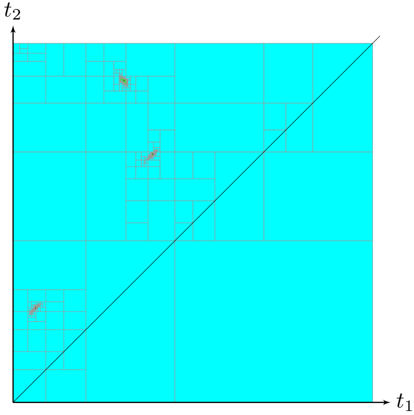
Temporal resolution

Bounded-error context

Uncertain trajectories enclosed in tubes.
Approximation of the enclosure of t-sets with SIVIA algorithms:



(c) Bounded trajectories

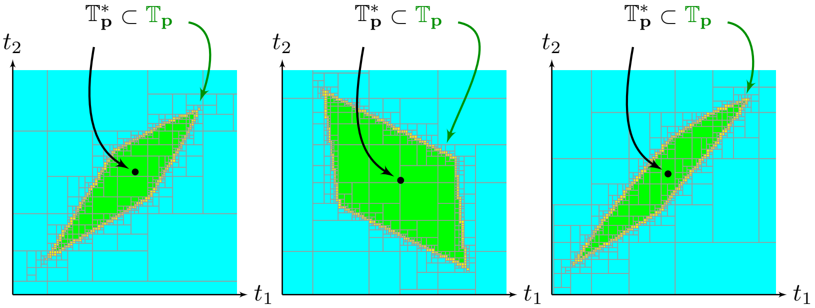


(d) Approximation of $\mathbb{T}_{\mathbf{p}}$

Temporal resolution

Bounded-error context

Approximation of the enclosure of t-sets with SIVIA algorithms:

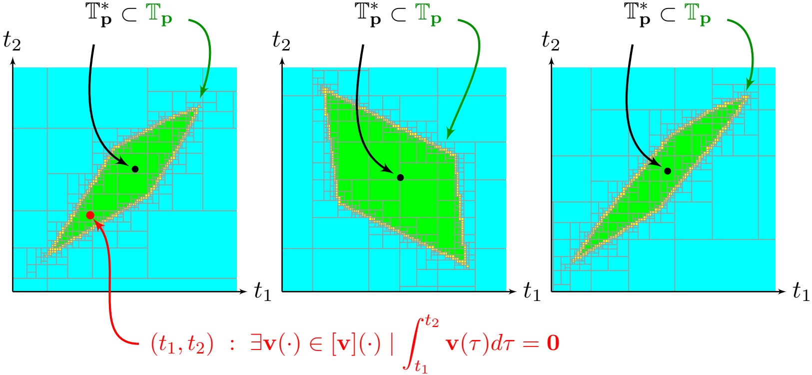


Zoom on the components of $\mathbb{T}_{\mathbf{p}}$

Temporal resolution

Bounded-error context

Approximation of the enclosure of t-sets with SIVIA algorithms:

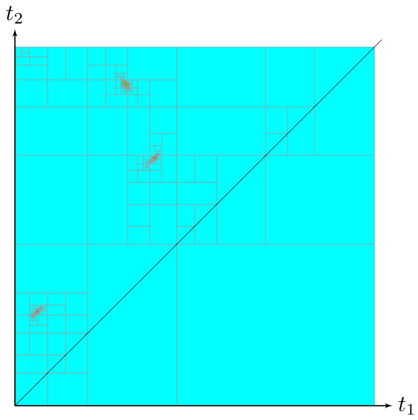


Zoom on the components of $\mathbb{T}_{\mathbf{p}}$

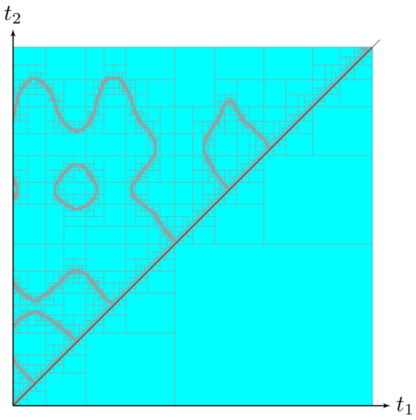
Temporal resolution

Bounded-error context

Approximation of the enclosure of t-sets with SIVIA algorithms:



(a) Approximation of $\mathbb{T}_{\mathbf{p}}$



(b) Approximation of $\mathbb{T}_z$

Temporal resolution

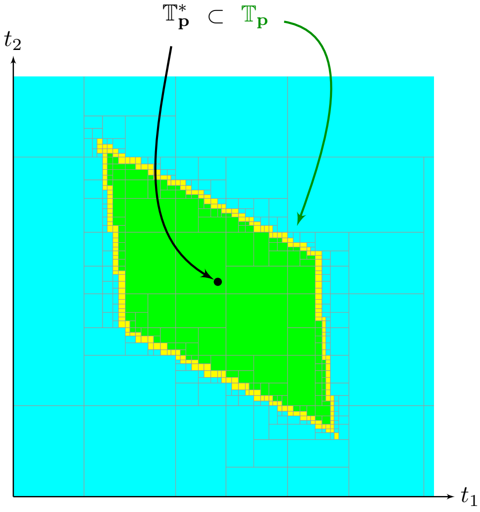
Intersection of the t-sets: fusion

Constraint:

- $\mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}^*_z$

Domains $\mathbb{T}_{\mathbf{p}}$, $\mathbb{T}_z$:

- $\mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}_{\mathbf{p}}$
- $\mathbb{T}^*_z \subset \mathbb{T}_z$



Approximation of $\mathbb{T}_{\mathbf{p}}$

Temporal resolution

Intersection of the t-sets: fusion

Constraint:

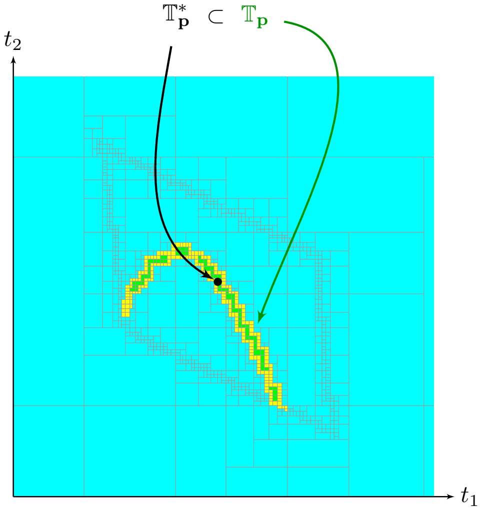
- $\mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}^*_z$

Domains $\mathbb{T}_{\mathbf{p}}$, $\mathbb{T}_z$:

- $\mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}_{\mathbf{p}}$
- $\mathbb{T}^*_z \subset \mathbb{T}_z$

Contraction:

- $\mathbb{T}_{\mathbf{p}} := \mathbb{T}_{\mathbf{p}} \cap \mathbb{T}_z$



Approximation of $\mathbb{T}_{\mathbf{p}}$

Temporal resolution

Intersection of the t-sets: fusion

Constraint:

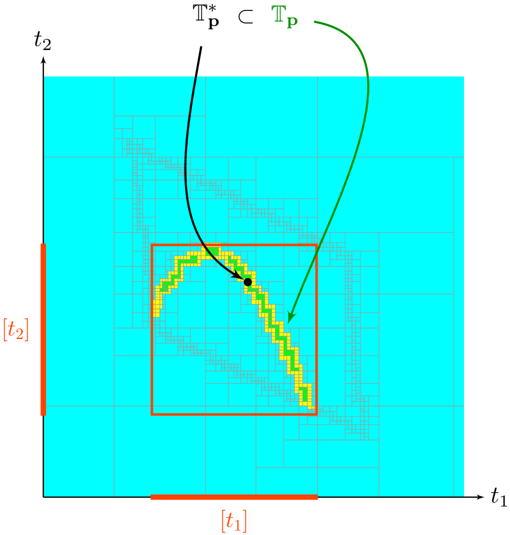
- $\mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}^*_z$

Domains $\mathbb{T}_{\mathbf{p}}$, $\mathbb{T}_z$:

- $\mathbb{T}^*_{\mathbf{p}} \subset \mathbb{T}_{\mathbf{p}}$
- $\mathbb{T}^*_z \subset \mathbb{T}_z$

Contraction:

- $\mathbb{T}_{\mathbf{p}} := \mathbb{T}_{\mathbf{p}} \cap \mathbb{T}_z$



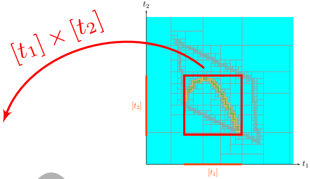
Approximation of $\mathbb{T}_{\mathbf{p}}$

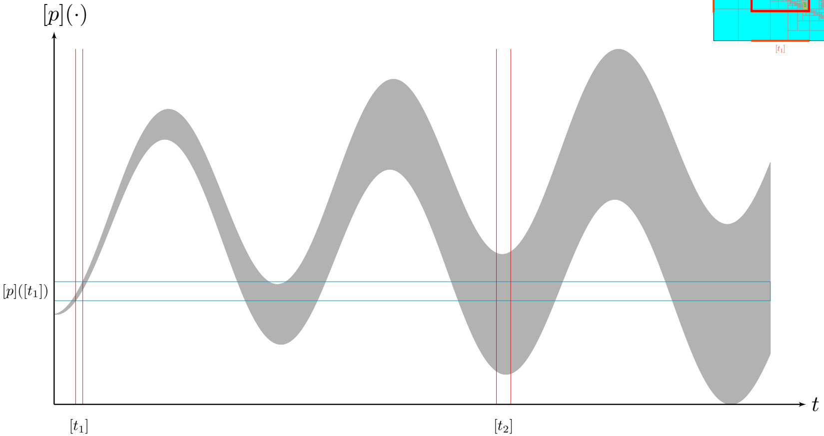
Temporal resolution

From times to positions

Robot localization = contraction of the tube $[\mathbf{p}](\cdot)$

- $\mathbf{t}^* \in [t_1] \times [t_2], \quad \mathbf{p}^*(\cdot) \in [\mathbf{p}](\cdot)$
- constraint: $\mathbf{p}(t_1) = \mathbf{p}(t_2)$



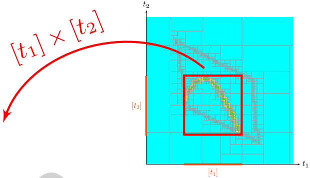


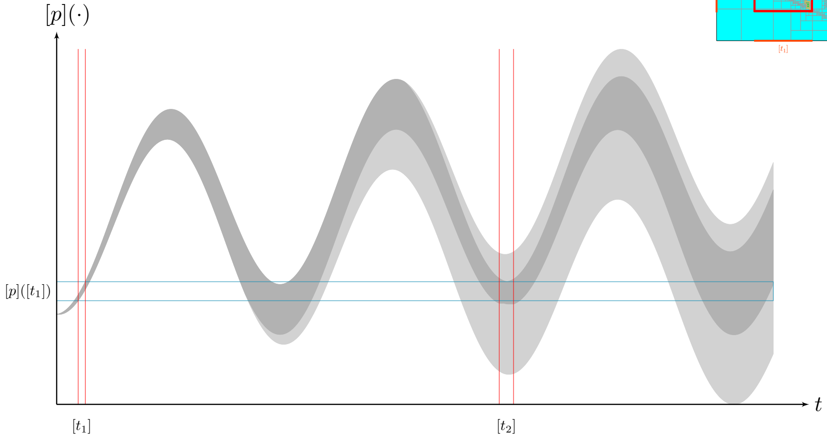
Temporal resolution

From times to positions

Robot localization = contraction of the tube $[\mathbf{p}](\cdot)$

- $\mathbf{t}^* \in [t_1] \times [t_2], \quad \mathbf{p}^*(\cdot) \in [\mathbf{p}](\cdot)$
- constraint: $\mathbf{p}(t_1) = \mathbf{p}(t_2)$



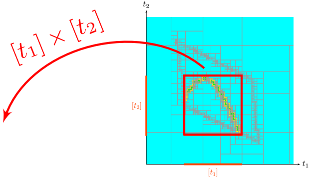


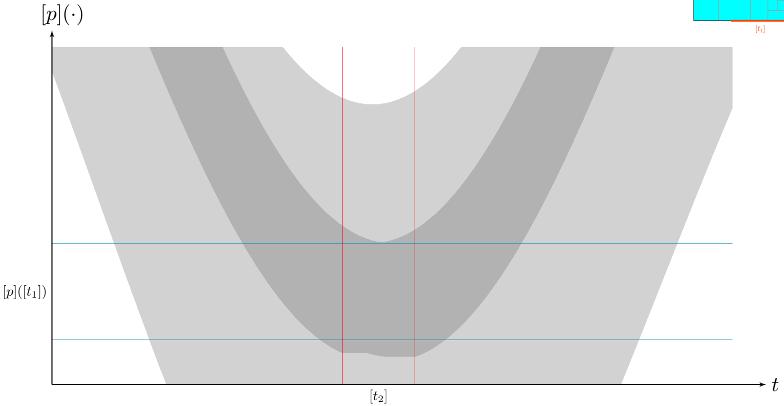
Temporal resolution

From times to positions

Robot localization = contraction of the tube $[\mathbf{p}](\cdot)$

- $\mathbf{t}^* \in [t_1] \times [t_2], \quad \mathbf{p}^*(\cdot) \in [\mathbf{p}](\cdot)$
- constraint: $\mathbf{p}(t_1) = \mathbf{p}(t_2)$



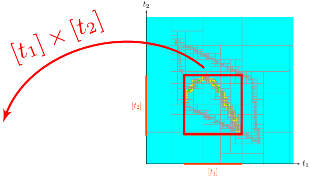


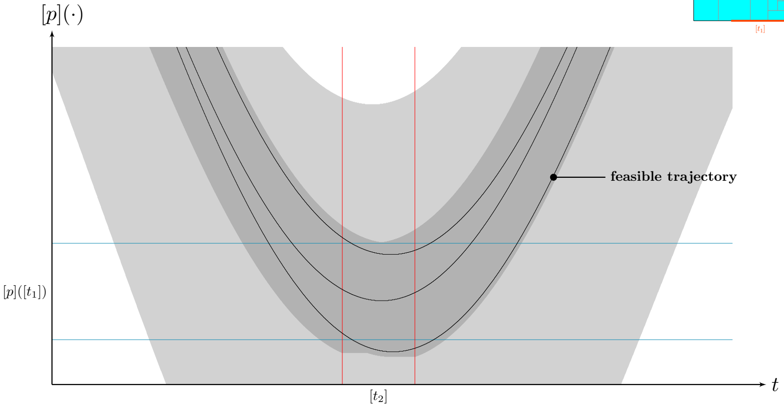
Temporal resolution

From times to positions

Robot localization = contraction of the tube $[\mathbf{p}](\cdot)$

- $\mathbf{t}^* \in [t_1] \times [t_2], \quad \mathbf{p}^*(\cdot) \in [\mathbf{p}](\cdot)$
- constraint: $\mathbf{p}(t_1) = \mathbf{p}(t_2)$





Section 4

Sea trials

Sea trials

Experimental mission with the Daurade AUV

- Daurade: Autonomous Underwater Vehicle
- weight: 1010kg – length: 5m – max depth: 300m

Special thanks to DGA-TN Brest (formerly GESMA) and SHOM

Sea trials

Experimental mission with the Daurade AUV

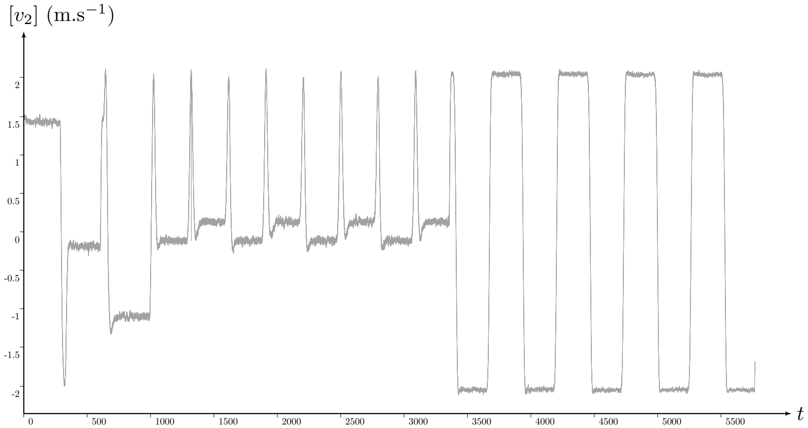
- Daurade: Autonomous Underwater Vehicle
- weight: 1010kg – length: 5m – max depth: 300m

Special thanks to DGA-TN Brest (formerly GESMA) and SHOM

Sea trials

Evolution measurements

- velocity measurements obtained with a DVL
- considering uncertainties, building a tube $[\mathbf{v}](\cdot)$

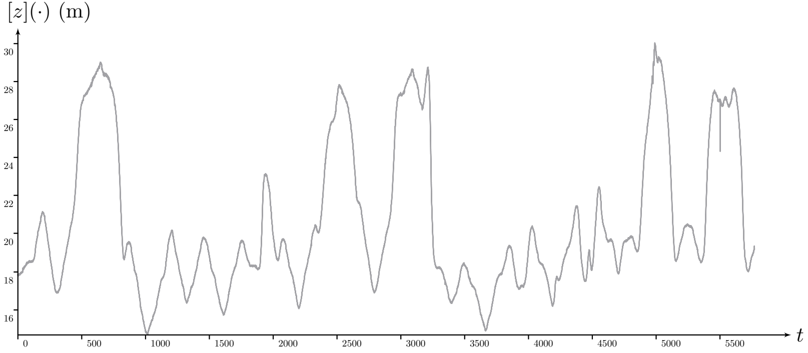


North speed velocity tube $[v_2](\cdot)$

Sea trials

Observations measurements: bathymetric values

- DVL, same sensor, can provide **altitude measurements** z_{alt}
- pressure sensor: depth values z_{depth}
- time-dependent measurements, use of **tide models**
- $z = z_{\mathrm{alt}} + z_{\mathrm{depth}} + z_{\mathrm{tide}}$



Observation tube $[z](\cdot)$

Sea trials

Dead-reckoning

Actual trajectory:

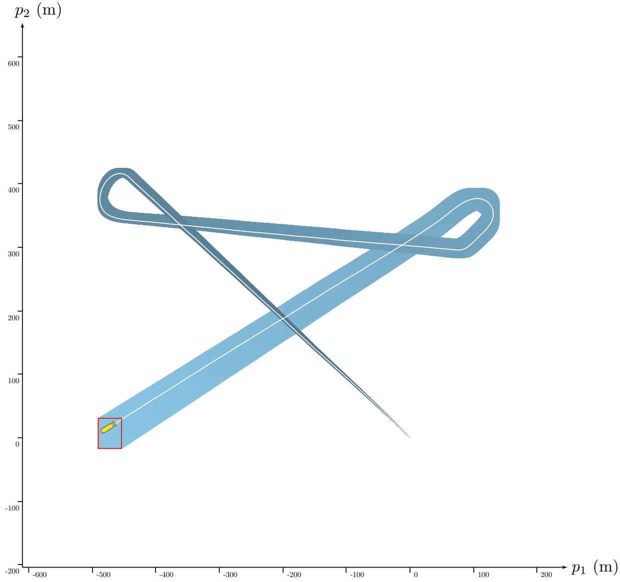
- white

Tube of positions:

- blue

Last position box:

- red



Sea trials

Dead-reckoning

Actual trajectory:

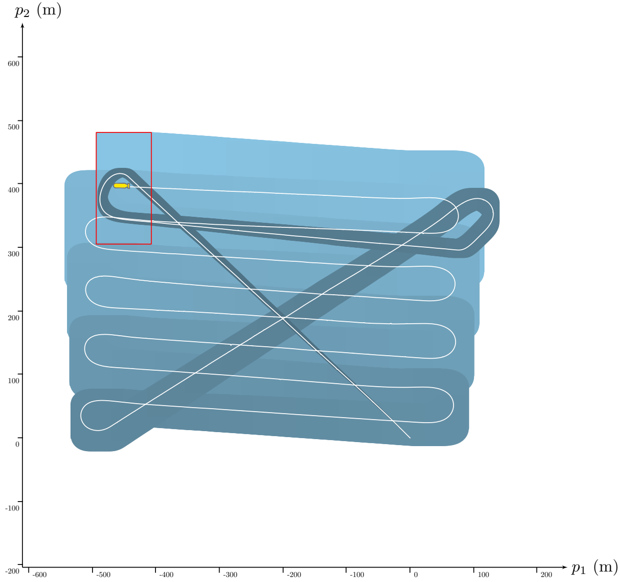
- white

Tube of positions:

- blue

Last position box:

- red



Sea trials

Dead-reckoning

Actual trajectory:

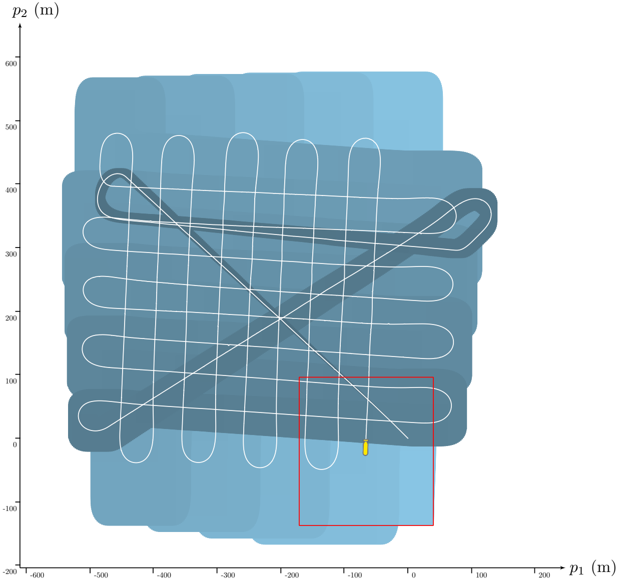
- white

Tube of positions:

- blue

Last position box:

- red



Sea trials

Dead-reckoning

Actual trajectory:

- white

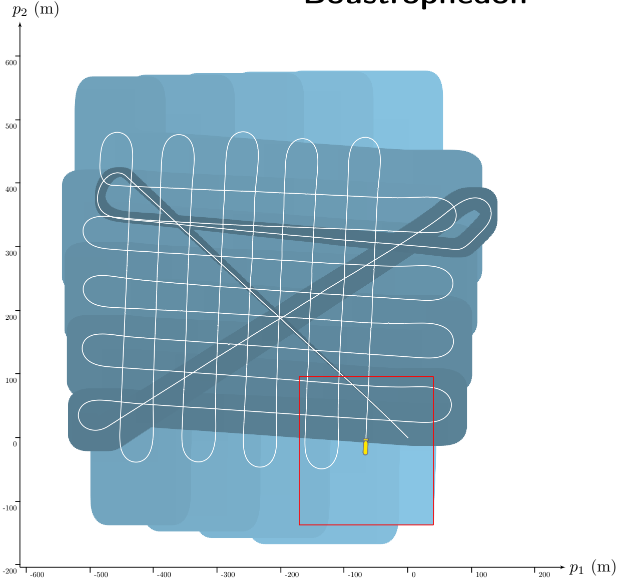
Tube of positions:

- blue

Last position box:

- red

Boustrophedon



Sea trials

SLAM results

Actual trajectory:

- white

Tube of positions:

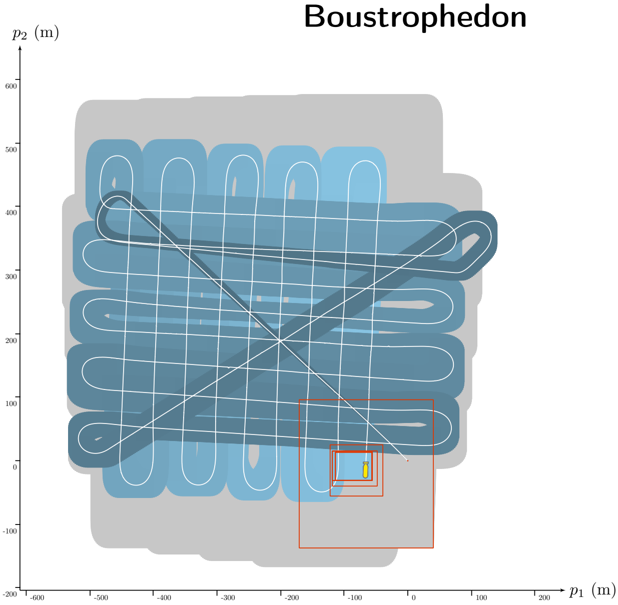
- blue

Last position box:

- red

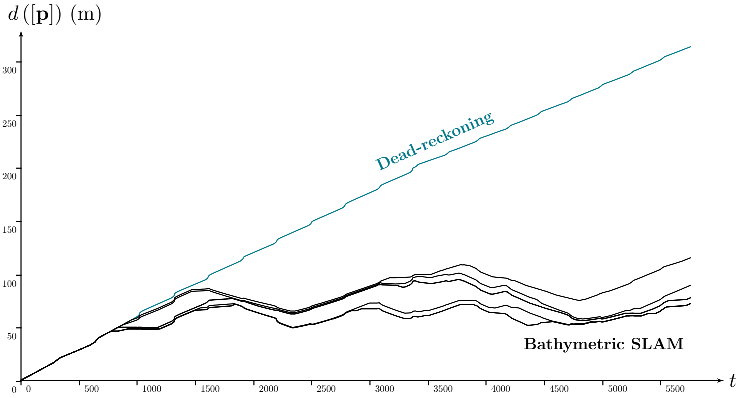
Contracted parts:

- gray



Sea trials

SLAM results



Thicknesses of the tube $[\mathbf{p}](\cdot)$

Localization:

- dead-reckoning: linear drift
- SLAM: no cumulated drift

Constraint method:

- iterative resolution
- reliable outputs, pessimism

Section 5

Conclusions

Conclusions

Originality of this work

- localization even in case of **unknown observation function** $\mathbf{g}$
  inter-temporal measurements

Conclusions

Originality of this work

- localization even in case of **unknown observation function** $\mathbf{g}$
  inter-temporal measurements
- consideration of any kind of **time-invariant measurements**
  for instance: temperatures, radioactivity, electric fields

Conclusions

Originality of this work

- localization even in case of **unknown observation function** $\mathbf{g}$
  inter-temporal measurements
- consideration of any kind of **time-invariant measurements**
  for instance: temperatures, radioactivity, electric fields
- **temporal resolution**
  approximation of time references

Conclusions

Originality of this work

- localization even in case of **unknown observation function** $\mathbf{g}$
  inter-temporal measurements
- consideration of any kind of **time-invariant measurements**
  for instance: temperatures, radioactivity, electric fields
- **temporal resolution**
  approximation of time references
- study of new **constraints over dynamical systems**
  ODEs, time uncertainties, delays, ...

Conclusions

Originality of this work

- localization even in case of **unknown observation function** $\mathbf{g}$
  inter-temporal measurements
- consideration of any kind of **time-invariant measurements**
  for instance: temperatures, radioactivity, electric fields
- **temporal resolution**
  approximation of time references
- study of new **constraints over dynamical systems**
  ODEs, time uncertainties, delays, ...

Tubex library: open-source library providing tools for constraint programming over dynamical systems

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

A temporal approach for the SLAM problem

— thank you for your attention —

Bibliography

- **Contractor Programming**
G. Chabert, L. Jaulin. *Artificial Intelligence*, 2009

- **A Constraint Satisfaction Approach for Enclosing Solutions to Parametric ODEs**
M. Janssen, P. Van Hentenryck, Y. Deville. *SIAM Journal on Numerical Analysis*, 2002

- **Analytic constraint solving and interval arithmetic**
T. J. Hickey. *ACM Press*, 2000

- **Constraint Satisfaction Differential Problems**
J. Cruz, P. Barahona. *Springer Berlin Heidelberg*, 2003

- **Set-membership state estimation with fleeting data**
F. Le Bars, J. Sliwka, L. Jaulin, O. Reynet *Automatica*, 2012

- **Solving Non-Linear Constraint Satisfaction Problems Involving Time-Dependant Functions**
A. Bethencourt, L. Jaulin. *Mathematics in Computer Science*, 2014

- **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. *Automatica*, 2018

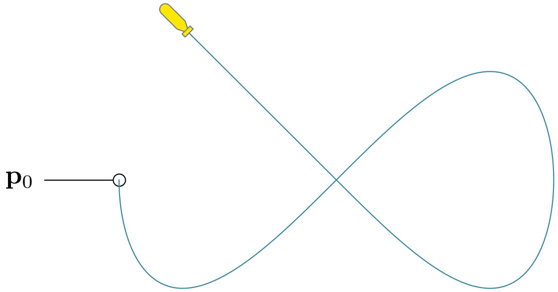
- **Proving the existence of loops in robot trajectories**
S. Rohou, P. Franek, C. Aubry, L. Jaulin. *International Journal of Robotics Research*, 2018

Section 6

Appendices

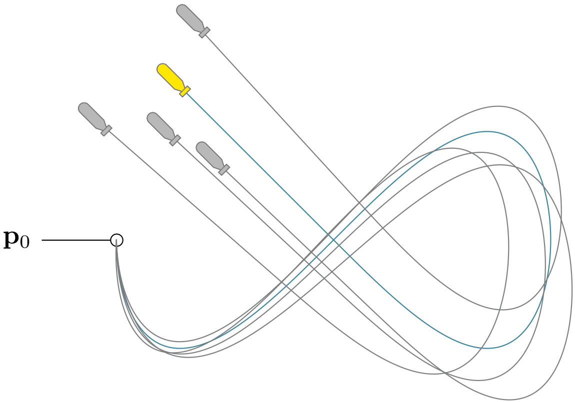
Appendices

Uncertain trajectories



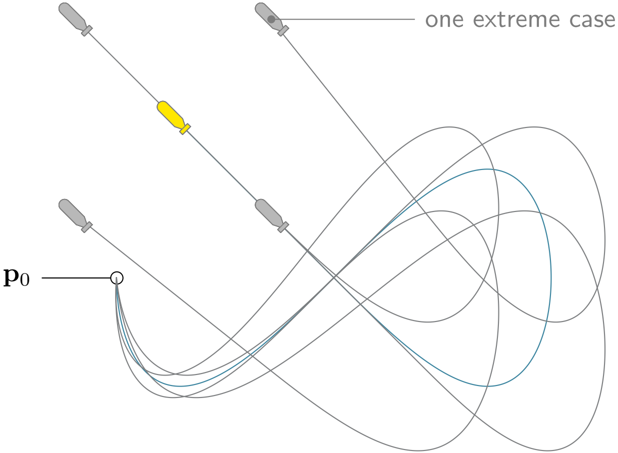
Appendices

Uncertain trajectories



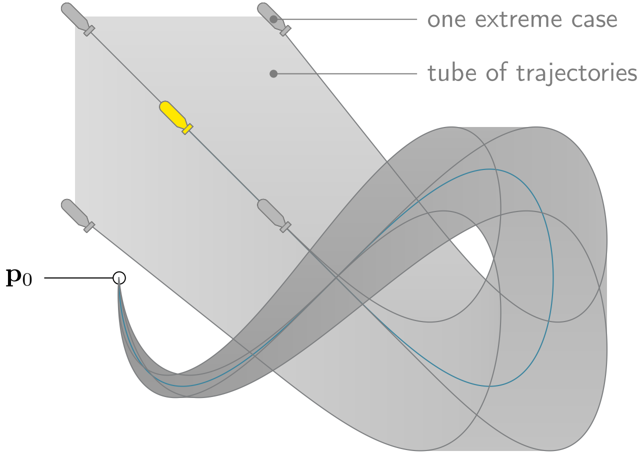
Appendices

Uncertain trajectories



Appendices

Uncertain trajectories



Appendices

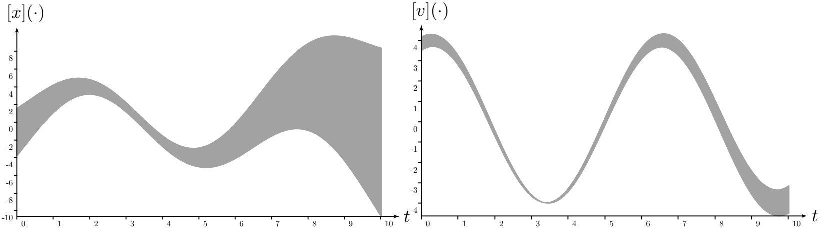
Differential constraint $\mathcal{L}_{\frac{d}{dt}}\big(\mathbf{x}(\cdot),\mathbf{v}(\cdot)\big) : \dot{\mathbf{x}}(\cdot)=\mathbf{v}(\cdot)$

Proposition: contractor $\mathcal{C}_{\frac{d}{dt}}$ defined as

$$\begin{pmatrix} [x](t) \\ [v](t) \end{pmatrix} \xmapsto{\mathcal{C}_{\frac{d}{dt}}} \begin{pmatrix} \displaystyle\bigcap_{t_1=t_0}^{t_f} \left([x](t_1)+\int_{t_1}^{t}[v](\tau)d\tau\right) \\ [v](t) \end{pmatrix}$$

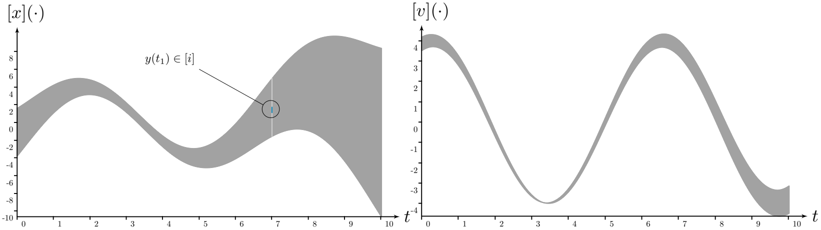
Appendices

Differential constraint $\mathcal{L}_{\frac{d}{dt}}\big(\mathbf{x}(\cdot),\mathbf{v}(\cdot)\big) : \dot{\mathbf{x}}(\cdot)=\mathbf{v}(\cdot)$



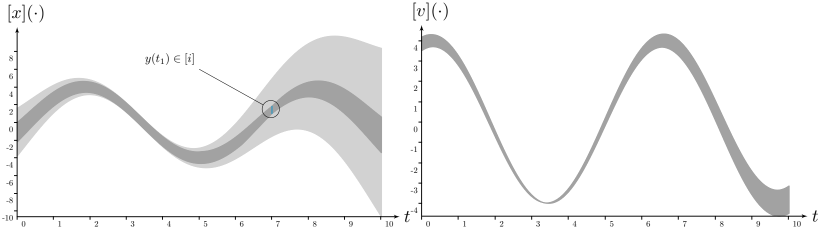
Appendices

Differential constraint $\mathcal{L}_{\frac{d}{dt}}\big(\mathbf{x}(\cdot),\mathbf{v}(\cdot)\big) : \dot{\mathbf{x}}(\cdot)=\mathbf{v}(\cdot)$



Appendices

Differential constraint $\mathcal{L}_{\frac{d}{dt}}\big(\mathbf{x}(\cdot),\mathbf{v}(\cdot)\big) : \dot{\mathbf{x}}(\cdot)=\mathbf{v}(\cdot)$



Appendices

Evaluation constraint $\mathcal{L}_{\text{eval}}\big(t,\mathbf{z},\mathbf{y}(\cdot),\mathbf{w}(\cdot)\big)$

$$\mathcal{L}_{\text{eval}}\big(t,\mathbf{z},\mathbf{y}(\cdot),\mathbf{w}(\cdot)\big) : \begin{cases} \mathbf{z} = \mathbf{y}(t) \\ \dot{\mathbf{y}}(\cdot) = \mathbf{w}(\cdot) \end{cases}$$

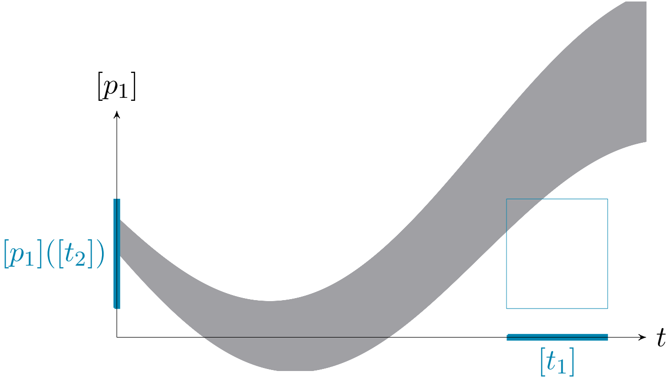
Proposition: contractor $\mathcal{C}_{\text{eval}}$ defined as

$$\begin{pmatrix} [t] \\ [z] \\ [y](\cdot) \\ [w](\cdot) \end{pmatrix} \xmapsto{\mathcal{C}_{\text{eval}}} \begin{pmatrix} [t] \cap [y]^{-1}([z]) \\ [z] \cap [y]([t]) \\ [y](\cdot) \cap \bigsqcup\limits_{t_1 \in [t]} \left(([y](t_1) \cap [z]) + \int_{t_1}^{\cdot} [w](\tau)d\tau \right) \\ [w](\cdot) \end{pmatrix}$$

Appendices

Evaluation constraint $\mathcal{L}_{\text{eval}}\big(t, \mathbf{z}, \mathbf{y}(\cdot), \mathbf{w}(\cdot)\big)$

$$\mathcal{C}_{\text{eval}}\big([t_1], [p_1]([t_2]), [p_1](\cdot), [v_1](\cdot)\big)$$



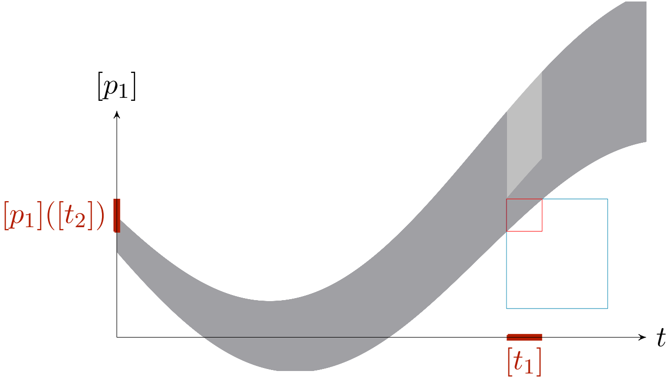
tube $[p_1](\cdot)$ before contraction

Reliable non-linear state estimation involving time uncertainties
Rohou, Jaulin, Mihaylova, Le Bars, Veres
Automatica, 2018

Appendices

Evaluation constraint $\mathcal{L}_{\text{eval}}\big(t, \mathbf{z}, \mathbf{y}(\cdot), \mathbf{w}(\cdot)\big)$

$$\mathcal{C}_{\text{eval}}\big([t_1], [p_1]([t_2]), [p_1](\cdot), [v_1](\cdot)\big)$$



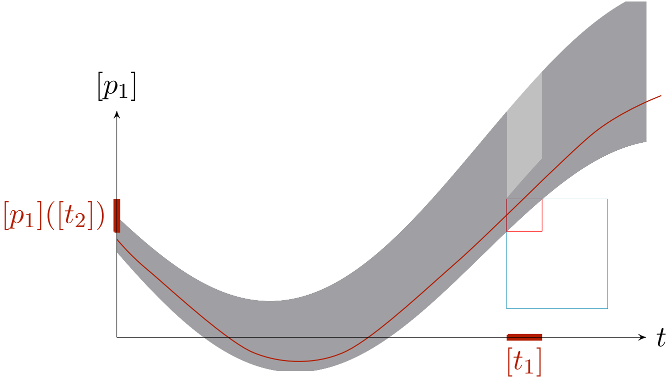
contraction of tube $[p_1](\cdot)$ and both $[p_1]([t_2])$ and $[t_1]$

Reliable non-linear state estimation involving time uncertainties
Rohou, Jaulin, Mihaylova, Le Bars, Veres
Automatica, 2018

Appendices

Evaluation constraint $\mathcal{L}_{\text{eval}}\big(t, \mathbf{z}, \mathbf{y}(\cdot), \mathbf{w}(\cdot)\big)$

$$\mathcal{C}_{\text{eval}}\big([t_1], [p_1]([t_2]), [p_1](\cdot), [v_1](\cdot)\big)$$



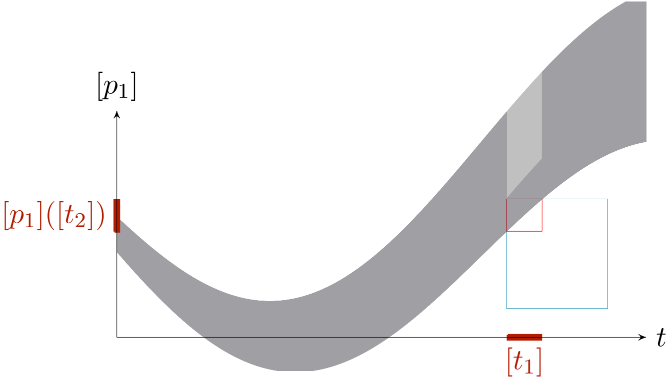
contraction of tube $[p_1](\cdot)$ and both $[p_1]([t_2])$ and $[t_1]$

Reliable non-linear state estimation involving time uncertainties
Rohou, Jaulin, Mihaylova, Le Bars, Veres
Automatica, 2018

Appendices

Evaluation constraint $\mathcal{L}_{\text{eval}}\big(t, \mathbf{z}, \mathbf{y}(\cdot), \mathbf{w}(\cdot)\big)$

$$\mathcal{C}_{\text{eval}}\big([t_1], [p_1]([t_2]), [p_1](\cdot), [v_1](\cdot)\big)$$



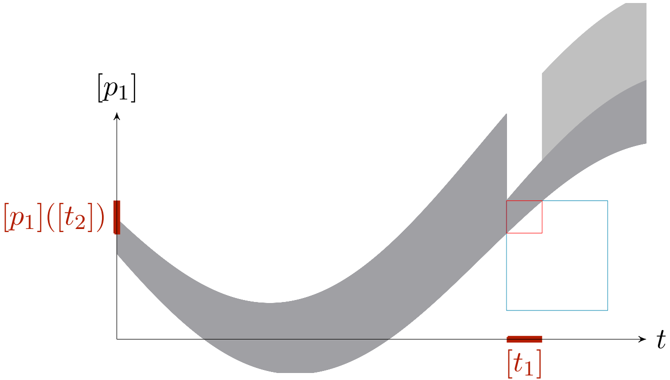
contraction of tube $[p_1](\cdot)$ and both $[p_1]([t_2])$ and $[t_1]$

Reliable non-linear state estimation involving time uncertainties
Rohou, Jaulin, Mihaylova, Le Bars, Veres
Automatica, 2018

Appendices

Evaluation constraint $\mathcal{L}_{\text{eval}}\big(t, \mathbf{z}, \mathbf{y}(\cdot), \mathbf{w}(\cdot)\big)$

$$\mathcal{C}_{\text{eval}}\big([t_1], [p_1]([t_2]), [p_1](\cdot), [v_1](\cdot)\big)$$



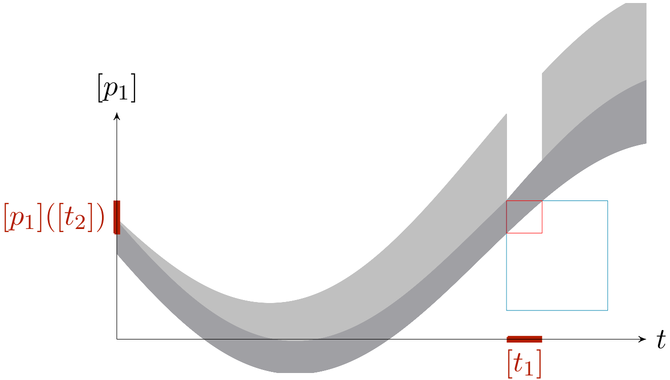
tube contraction in forward

Reliable non-linear state estimation involving time uncertainties
Rohou, Jaulin, Mihaylova, Le Bars, Veres
Automatica, 2018

Appendices

Evaluation constraint $\mathcal{L}_{\text{eval}}\big(t, \mathbf{z}, \mathbf{y}(\cdot), \mathbf{w}(\cdot)\big)$

$$\mathcal{C}_{\text{eval}}\big([t_1], [p_1]([t_2]), [p_1](\cdot), [v_1](\cdot)\big)$$

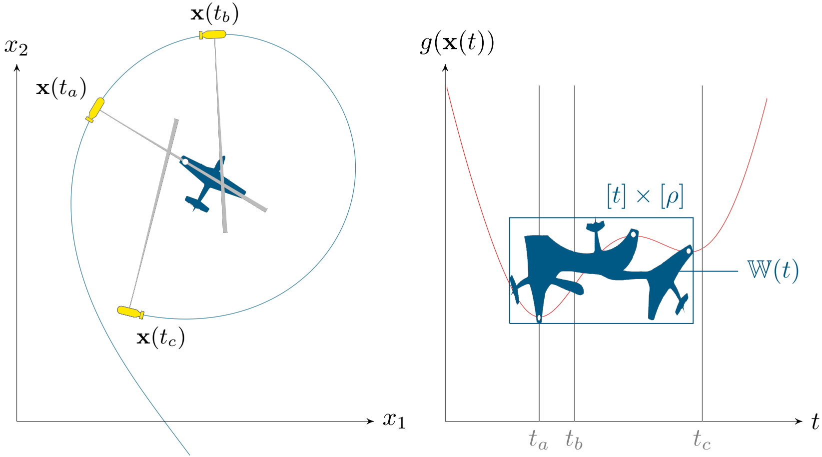


tube contraction in forward/backward

Reliable non-linear state estimation involving time uncertainties
Rohou, Jaulin, Mihaylova, Le Bars, Veres
Automatica, 2018

Appendices

Wreck-based localization method



Appendices

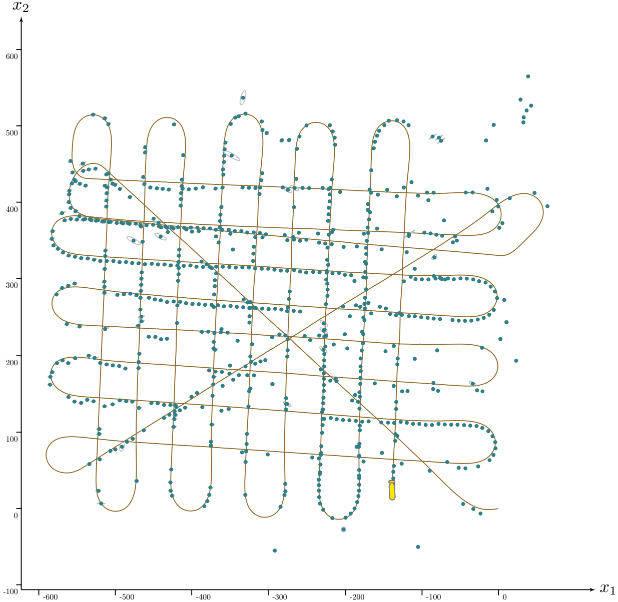
USBL

Appendices

USBL

Appendices

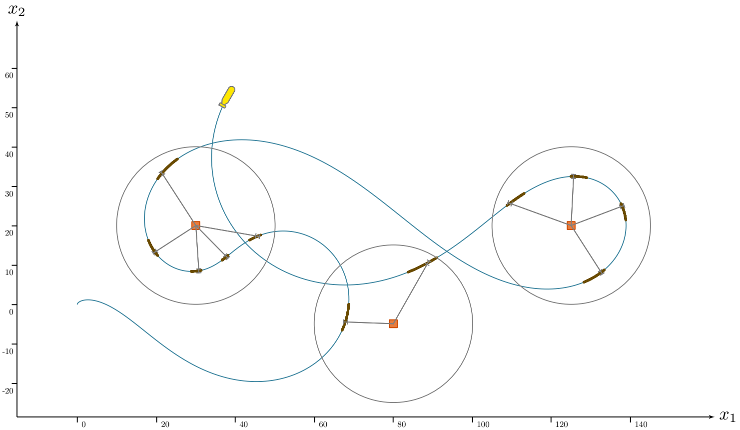
USBL



Appendices

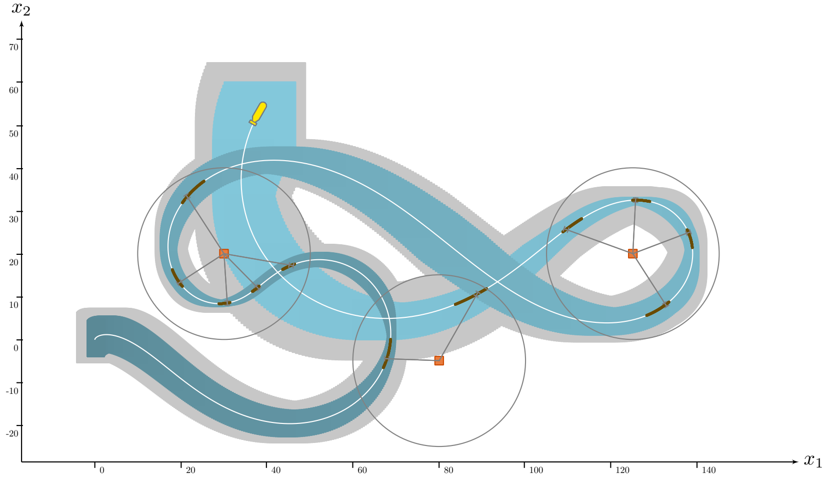
State estimation: mobile robotics

$$\begin{cases} \dot{\mathbf{x}}(\cdot) = \mathbf{f}(\mathbf{x}(\cdot), \mathbf{u}(\cdot)) \\ z_i = g(\mathbf{x}(t_i)) \end{cases}$$



Appendices

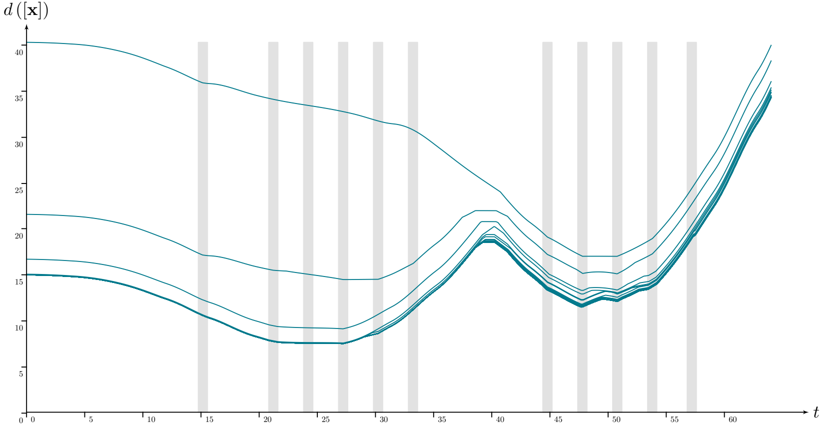
State estimation: mobile robotics



A mobile robot evolving among beacons – bounded error context

Appendices

State estimation: mobile robotics



Drifting curves – thicknesses of positioning tubes

Appendices

The $\mathcal{L}_{t_1,t_2}$ constraint: decomposition

$\mathcal{L}_{t_1,t_2}$, not canonic, amounts to the following composition:

Variables: t_1, t_2, $\mathbf{y}(\cdot)$

Constraints:

- $\mathbf{y}(t_1) = \mathbf{y}(t_2)$

Domains: $[t_1]$, $[t_2]$, $[\mathbf{y}](\cdot)$

Appendices

The $\mathcal{L}_{t_1,t_2}$ constraint: decomposition

$\mathcal{L}_{t_1,t_2}$, not canonic, amounts to the following composition:

Variables: t_1, t_2, $\mathbf{y}(\cdot)$

Constraints:

- $\mathbf{y}(t_1) = \mathbf{y}(t_2) \iff \begin{cases} \mathbf{a} = \mathbf{y}(t_1) \\ \mathbf{b} = \mathbf{y}(t_2) \\ \mathbf{a} = \mathbf{b} \end{cases}$

Domains: $[t_1]$, $[t_2]$, $[\mathbf{y}](\cdot)$

Appendices

The $\mathcal{L}_{t_1,t_2}$ constraint: decomposition

$\mathcal{L}_{t_1,t_2}$, not canonic, amounts to the following composition:

$$\left\{ \begin{array}{l} \textbf{Variables: } t_1,\ t_2,\ \mathbf{y}(\cdot) \\ \textbf{Constraints:} \\ \quad \blacktriangleright\ \mathbf{y}(t_1) = \mathbf{y}(t_2) \iff \left\{ \begin{array}{l} \mathbf{a} = \mathbf{y}(t_1) \\ \mathbf{b} = \mathbf{y}(t_2) \\ \mathbf{a} = \mathbf{b} \end{array} \right. \iff \left\{ \begin{array}{l} \mathcal{L}_{\text{eval}}\left(t_1, \mathbf{a}, \mathbf{y}(\cdot), \mathbf{w}(\cdot)\right) \\ \mathcal{L}_{\text{eval}}\left(t_2, \mathbf{b}, \mathbf{y}(\cdot), \mathbf{w}(\cdot)\right) \\ \mathbf{a} = \mathbf{b} \end{array} \right. \\ \textbf{Domains: } [t_1],\ [t_2],\ [\mathbf{y}](\cdot) \end{array} \right.$$

Appendices

The $\mathcal{L}_{t_1,t_2}$ constraint: decomposition

$\mathcal{L}_{t_1,t_2}$, not canonic, amounts to the following composition:

$$\left\{\begin{array}{l}
\textbf{Variables: } t_1,\ t_2,\ \mathbf{y}(\cdot),\ \mathbf{w}(\cdot) \\[1em]
\textbf{Constraints:} \\[1em]
\quad \blacktriangleright\ \mathbf{y}(t_1)=\mathbf{y}(t_2) \iff \left\{\begin{array}{l}\mathbf{a}=\mathbf{y}(t_1)\\ \mathbf{b}=\mathbf{y}(t_2)\\ \mathbf{a}=\mathbf{b}\end{array}\right. \iff \left\{\begin{array}{l}\mathcal{L}_{\mathrm{eval}}\left(t_1,\mathbf{a},\mathbf{y}(\cdot),\mathbf{w}(\cdot)\right)\\ \mathcal{L}_{\mathrm{eval}}\left(t_2,\mathbf{b},\mathbf{y}(\cdot),\mathbf{w}(\cdot)\right)\\ \mathbf{a}=\mathbf{b}\end{array}\right. \\[1em]
\quad \blacktriangleright\ \dot{\mathbf{y}}(\cdot)=\mathbf{w}(\cdot) \\[1em]
\textbf{Domains: } [t_1],\ [t_2],\ [\mathbf{y}](\cdot),\ [\mathbf{w}](\cdot)
\end{array}\right.$$

Appendices

The $\mathcal{L}_{t_1,t_2}$ constraint: decomposition

$\mathcal{L}_{t_1,t_2}$, not canonic, amounts to the following composition:

Variables: t_1, t_2, $\mathbf{y}(\cdot)$, $\mathbf{w}(\cdot)$

Constraints:

- $\mathbf{y}(t_1) = \mathbf{y}(t_2) \iff \begin{cases} \mathbf{a} = \mathbf{y}(t_1) \\ \mathbf{b} = \mathbf{y}(t_2) \\ \mathbf{a} = \mathbf{b} \end{cases} \iff \begin{cases} \mathcal{L}_{\text{eval}}\left(t_1, \mathbf{a}, \mathbf{y}(\cdot), \mathbf{w}(\cdot)\right) \\ \mathcal{L}_{\text{eval}}\left(t_2, \mathbf{b}, \mathbf{y}(\cdot), \mathbf{w}(\cdot)\right) \\ \mathbf{a} = \mathbf{b} \end{cases}$
- $\dot{\mathbf{y}}(\cdot) = \mathbf{w}(\cdot)$

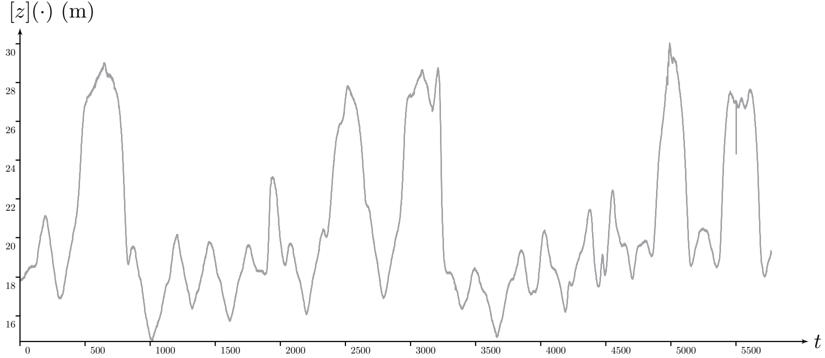
Domains: $[t_1]$, $[t_2]$, $[\mathbf{y}](\cdot)$, $[\mathbf{w}](\cdot)$

$\mathcal{L}_{\text{eval}}$ constraint:

- $\mathcal{L}_{\text{eval}}\left(t, \mathbf{z}, \mathbf{y}(\cdot), \mathbf{w}(\cdot)\right) : \begin{cases} \mathbf{z} = \mathbf{y}(t) \\ \dot{\mathbf{y}}(\cdot) = \mathbf{w}(\cdot) \end{cases}$

Appendices

Daurade mission: 20/10/2015 11h



Observation tube $[z](\cdot)$: bathymetric measurements

Appendices

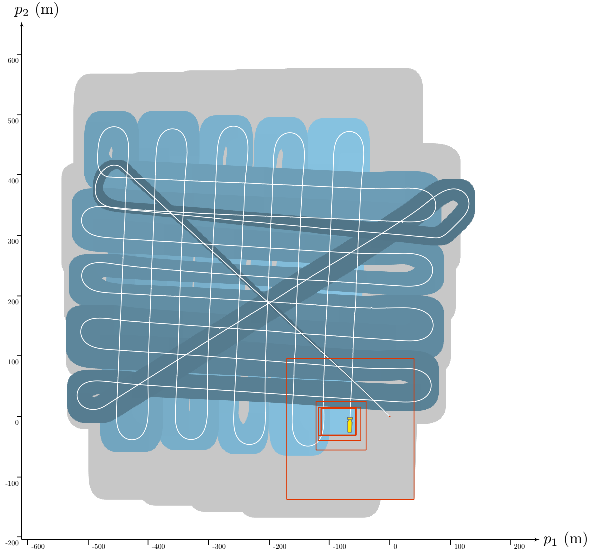
Daurade mission: 20/10/2015 11h

Actual trajectory:
- white

Tube of positions:
- blue

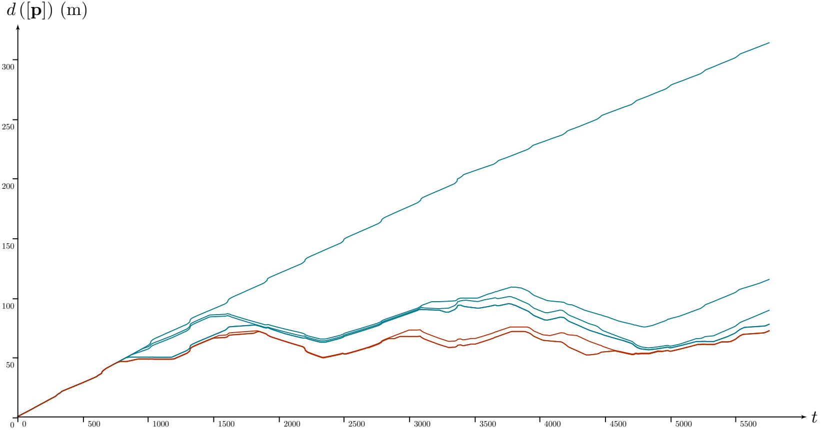
Last position box:
- red

Contracted parts:
- gray



Appendices

Daurade mission: 20/10/2015 11h



Thicknesses of the tube $[\mathbf{p}](\cdot)$

Appendices

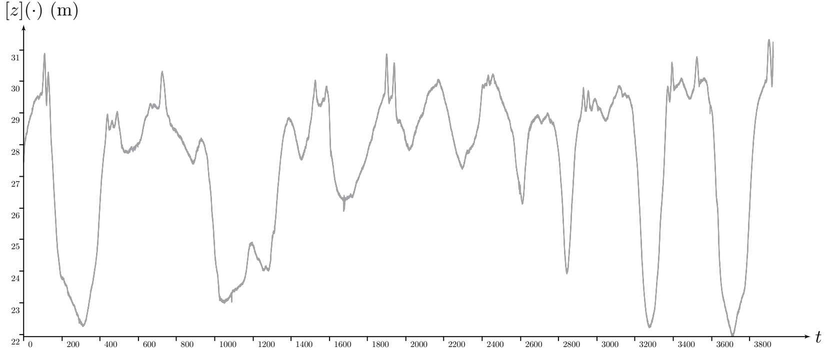
Daurade mission: 20/10/2015 11h

SLAM iterations on *Daurade*'s experiment:

i	loop detections	loop proofs	computation time	cumulated comp. time	$[\mathbf{p}](t_f)$ contraction	SLAM algorithm
1	122	104	259s	259s	63.22%	fast
2	128	112	192s	451s	71.46%	fast
3	128	112	172s	623s	75.17%	fast
4	129	115	180s	803s	75.22%	fast
5	129	115	182s	985s (0h16)	**75.22%**	fast
			fixed point			
6	129	115	2708s (0h45)	3693s (1h02)	76.91%	accurate
7	129	115	2506s (0h41)	6199s (1h43)	76.96%	accurate
8	129	115	2391s (0h40)	8590s (2h23)	**76.96%**	accurate
			fixed point			

Appendices

Daurade mission: 19/10/2015 10h



Observation tube $[z](\cdot)$: bathymetric measurements

Appendices

Daurade mission: 19/10/2015 10h

Actual trajectory:

- white

Tube of positions:

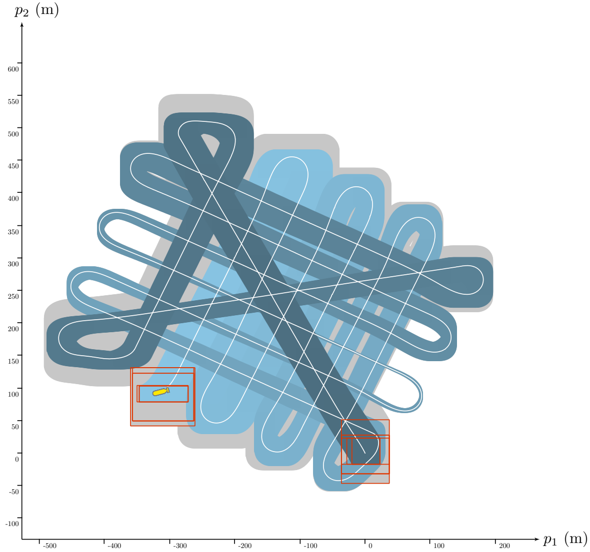
- blue

Last position box:

- red

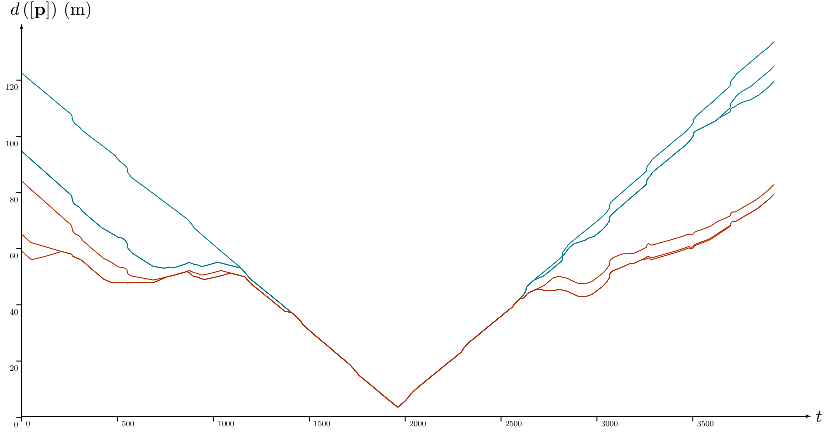
Contracted parts:

- gray



Appendices

Daurade mission: 19/10/2015 10h



Thicknesses of the tube $[\mathbf{p}](\cdot)$

Appendices

Daurade mission: 19/10/2015 10h

SLAM iterations on *Daurade*'s experiment:

i	loop detections	loop proofs	computation time	cumulated comp. time	$[\mathbf{p}](t_0)$ contraction	SLAM algorithm
1	76	65	93s	93s	22.76%	fast
2	78	67	90s	183s	22.76%	fast
3	78	67	108s	291s (0h05)	**22.76%**	fast
fixed point						
4	78	67	1726s (0h29)	2017s (0h34)	31.47%	accurate
5	77	67	1392s (0h23)	3409s (0h57)	46.96%	accurate
6	77	67	1424s (0h24)	4833s (1h21)	51.85%	accurate
7	77	68	1470s (0h24)	6303s (1h45)	**51.85%**	accurate
fixed point						

Appendices

Dynamical constraints

SLAM problem was an opportunity to study the following elementary constraints:

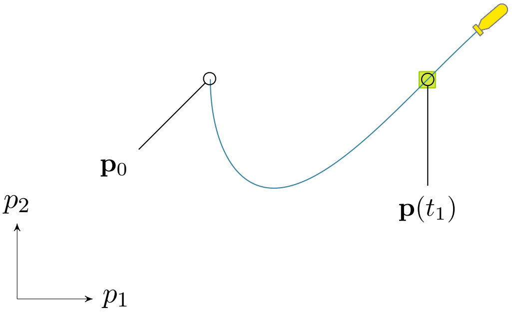
1. Evolution constraint
$$\mathcal{L}_{\frac{d}{dt}}\big(\mathbf{x}(\cdot),\mathbf{v}(\cdot)\big) : \dot{\mathbf{x}}(\cdot)=\mathbf{v}(\cdot)$$
2. Evaluation constraint
$$\mathcal{L}_{\text{eval}}\big(t,\mathbf{z},\mathbf{y}(\cdot),\mathbf{w}(\cdot)\big) : \begin{cases} \mathbf{z}=\mathbf{y}(t) \\ \dot{\mathbf{y}}(\cdot)=\mathbf{w}(\cdot) \end{cases}$$
3. Inter-temporal evaluation constraint
$$\mathcal{L}_{t_1,t_2}\big(t_1,t_2,\mathbf{y}(\cdot),\mathbf{w}(\cdot)\big) : \begin{cases} \mathbf{y}(t_1)=\mathbf{y}(t_2) \\ \dot{\mathbf{y}}(\cdot)=\mathbf{w}(\cdot) \end{cases}$$
4. Inter-temporal implication constraint
$$\mathcal{L}_{\mathbf{p}\Rightarrow\mathbf{z}}\big(\mathbf{y}(\cdot),\mathbf{w}(\cdot),\mathbf{z}(\cdot)\big) : \begin{cases} \mathbf{y}(t_1)=\mathbf{y}(t_2) \implies \mathbf{z}(t_1)=\mathbf{z}(t_2) \\ \dot{\mathbf{y}}(\cdot)=\mathbf{w}(\cdot) \end{cases}$$

Appendices

Dynamical constraints

Example:

- $\mathbf{x} = (p_1, p_2, p_3, \theta)^\intercal \in \mathbb{R}^4$
- $\mathbf{p}(t) = \mathbf{h}(\mathbf{x}(t)) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$

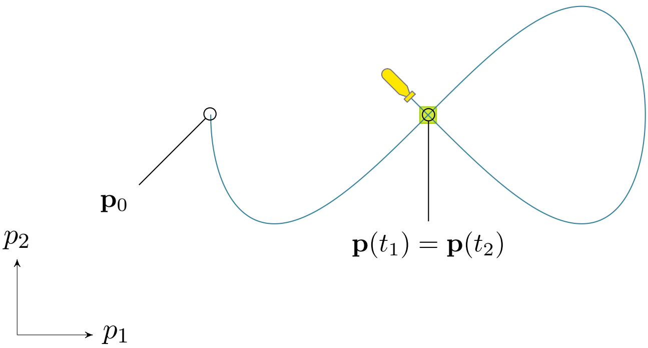


Appendices

Dynamical constraints

Example:

- $\mathbf{x} = (p_1, p_2, p_3, \theta)^\intercal \in \mathbb{R}^4$
- $\mathbf{p}(t) = \mathbf{h}(\mathbf{x}(t)) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$

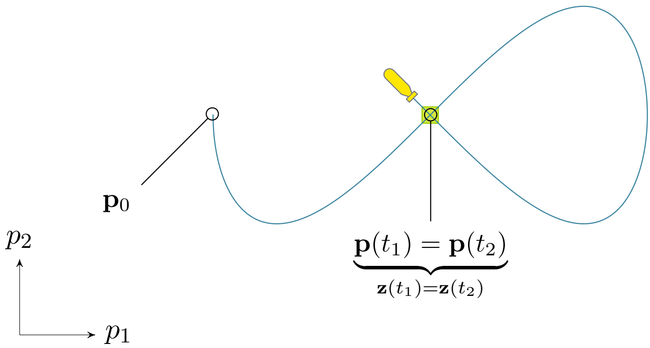


Appendices

Dynamical constraints

Example:

- $\mathbf{x} = (p_1, p_2, p_3, \theta)^\intercal \in \mathbb{R}^4$
- $\mathbf{p}(t) = \mathbf{h}(\mathbf{x}(t)) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$



Appendices

Constraints: decomposition

Complex constraints can be broken down.
Example, observation function for range-only state estimation:

$$\mathcal{L}_{\mathbf{g}}\left(\rho,\mathbf{a},\mathbf{b}\right):\rho=\sqrt{\left(a_1-b_1\right)^2+\left(a_2-b_2\right)^2}\Longleftrightarrow\left\{\begin{array}{rcl} c & = & a_1-b_1 \\ d & = & a_2-b_2 \\ i & = & c^2 \\ j & = & d^2 \\ l & = & i+j \\ \rho & = & \sqrt{l} \end{array}\right.$$

- $c,d,\ldots,l$: intermediate variables used for ease of decomposition
- network of **elementary constraints**: $\mathcal{L}_{-}$, $\mathcal{L}_{+}$, $\mathcal{L}_{(\cdot)^2}$, $\mathcal{L}_{\sqrt{\cdot}}$

Appendices

Constraints: application

Each elementary constraint $\mathcal{L}$ is applied by an operator:

- a **contractor** $\mathcal{C}_{\mathcal{L}} : \mathbb{IR}^n \rightarrow \mathbb{IR}^n$
- example, $\mathcal{C}_+$:

$$\begin{pmatrix} [a] \\ [x] \\ [y] \end{pmatrix} \mapsto \begin{pmatrix} [a] \cap ([x]+[y]) \\ [x] \cap ([a]-[y]) \\ [y] \cap ([a]-[x]) \end{pmatrix}$$

Contractor programming: chabert_contractor_2009

- contractor seen as a subset of $\mathbb{R}^n$

  $\Longrightarrow$ operations on sets applicable on contractors: $\cup$, $\cap$, ...

  $\Longrightarrow$ simple **combinations** of primitive contractors

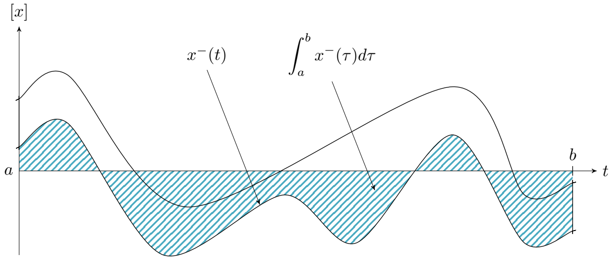
Appendices

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]



blue area: lower bound of the tube's integral

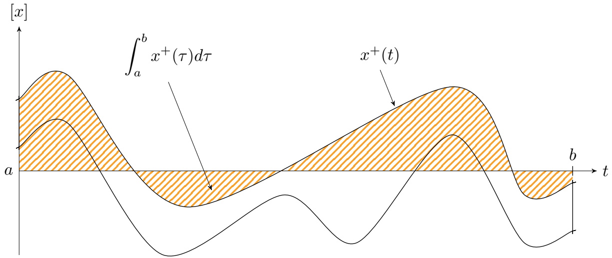
Appendices

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]



orange area: upper bound of the tube's integral

Back to the trajectories space

At this point:

- temporal set $\mathbb{T}_{\mathbf{p}}$ contracted,
- it remains to contract the positions tube $[\mathbf{p}](\cdot)$

Constraint of interest:

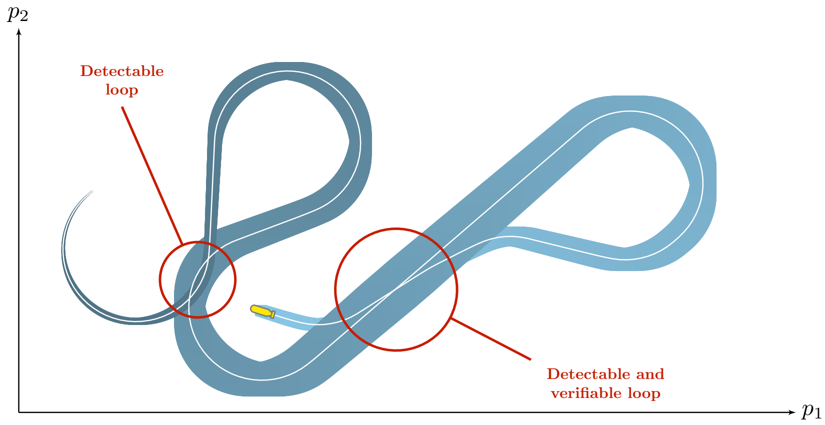
- $\mathbb{T}^{*}_{\mathbf{p}} = \{(t_1, t_2) \mid \mathbf{p}(t_1) = \mathbf{p}(t_2)\}$
- *backward way*: from the set $\mathbb{T}^{*}_{\mathbf{p}}$ to the trajectory $\mathbf{p}(\cdot)$

However:

- pessimistic enclosure $[\mathbf{p}](\cdot)$: $\mathbb{T}_{\mathbf{p}}$ may not contain a solution
  $\implies$ **risk of false contraction**
- before contracting $[\mathbf{p}](\cdot)$, need to prove that $\exists \mathbf{t} \in \mathbb{T}_{\mathbf{p}} \mid \mathbf{p}(t_1) = \mathbf{p}(t_2)$
- physically: we need to **prove loops** along the trajectory $\mathbf{p}(\cdot)$

Appendices

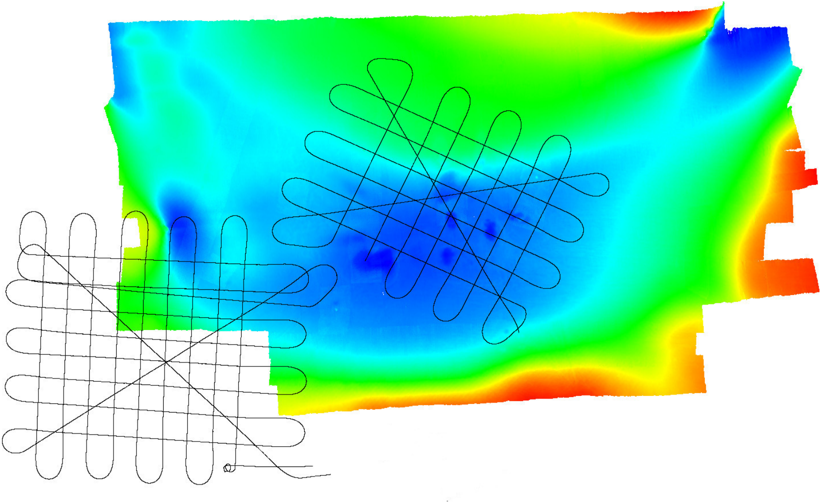
Proving the existence of loops



Proving the existence of loops in robot trajectories
S. Rohou, P. Franek, C. Aubry, L. Jaulin. *International Journal of Robotics Research*, 2018

Appendices

DEM of the experiments

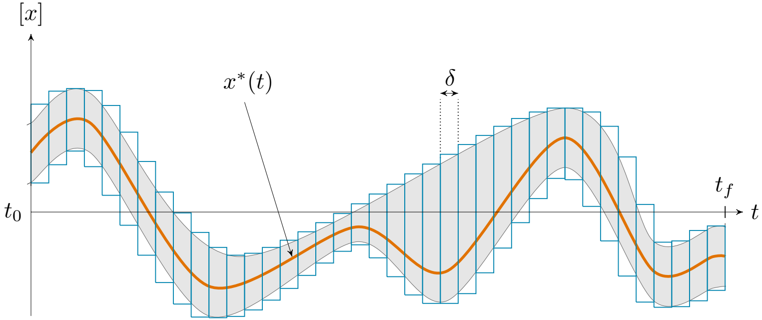


Actual Digital Elevation Model (DEM)

Appendices

Tubex library

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



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