Tube Programming
Applied to Underwater Localization

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

DSTL 2017-01

Section 1

Introduction

Introduction

Motivations: exploration of a wide underwater area

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, complex SLAM

Usual solution, **dead-reckoning**:

- navigation based on *proprioceptive* measurements
- fast drift on position estimation

Introduction

Mobile Robotics: formalization

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, $\dots$)
- $\mathbf{u} \in \mathbb{R}^m$ is the input vector (command)
- $y \in \mathbb{R}$ is some exteroceptive measurement (*e.g.* bathymetry)
- $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the *evolution* function
- $g : \mathbb{R}^n \to \mathbb{R}$ is the *observation* function

Introduction

Main thread of the PhD: inter-temporal measurements

Recalled state equations:

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

We focus of on some inter-temporal equation formally defined by:

$$h\left(\mathbf{x}(t_1), \mathbf{x}(t_2)\right) = 0 \implies y(t_1) = y(t_2)$$

Where:

- $h : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a function to be defined according to the considered problem

Introduction

Inter-temp. measurements: loop-based localization

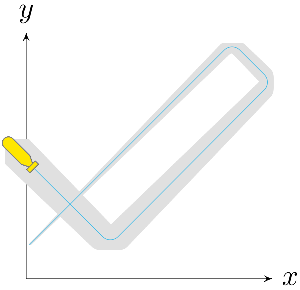
Application #1: a robot performing loops in its trajectory.
We compare measurements made over each cross.

Inter-temporal equation:

$$h\left(\mathbf{x}(t_1),\mathbf{x}(t_2)\right)=0 \implies y(t_1)=y(t_2)$$

Definition of h:

$$h\left(\mathbf{x}(t_1),\mathbf{x}(t_2)\right)=\mathbf{x}(t_1)-\mathbf{x}(t_2)$$



Introduction

Inter-temp. measurements: in symmetrical environments

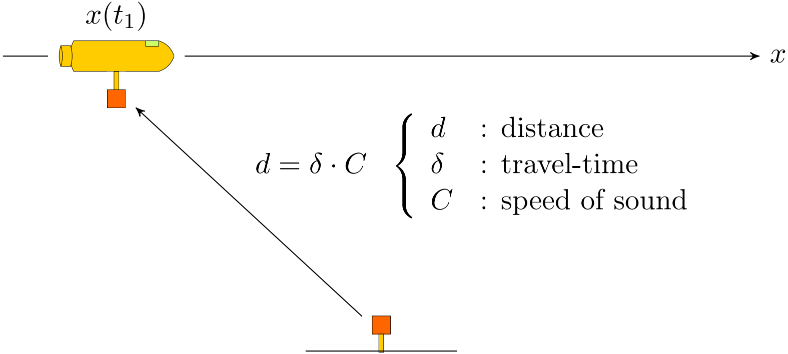
Application #2: localization in unknown underwater environments.
Considering a known speed of sound and linear acoustic rays:

Figure: AUV's localization with an acoustic beacon on the seabed

Introduction

Inter-temp. measurements: in symmetrical environments

Application #2: localization in unknown underwater environments.
Considering a known speed of sound and linear acoustic rays:



$$d = \delta \cdot C \quad \begin{cases} d & : \text{distance} \\ \delta & : \text{travel-time} \\ C & : \text{speed of sound} \end{cases}$$

Figure: AUV's localization with an acoustic beacon on the seabed

Introduction

Inter-temp. measurements: in symmetrical environments

Application #2: localization in unknown underwater environments.
Considering refractions (Snell-Descartes) and no knowledge on C:

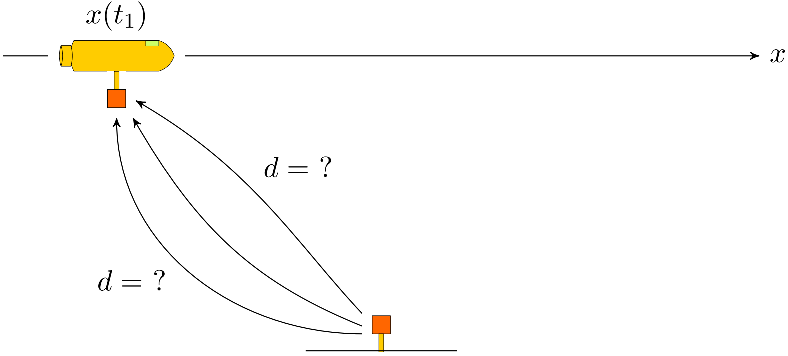


Figure: AUV's localization with an acoustic beacon on the seabed

Introduction

Inter-temp. measurements: in symmetrical environments

Compensation of uncertainties with inter-temporal measurements.

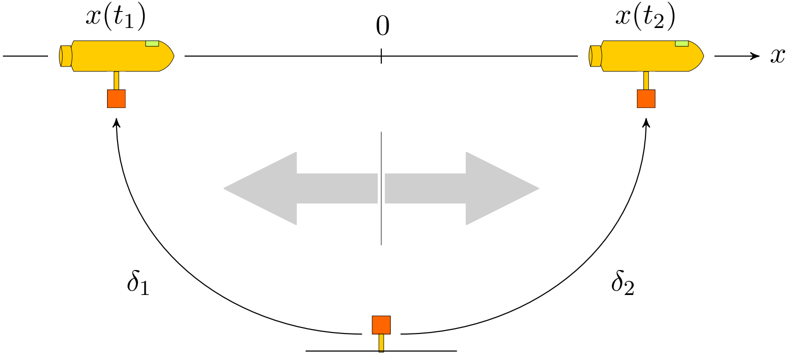


Figure: AUV's localization with an acoustic beacon on the seabed

Introduction

Inter-temp. measurements: in symmetrical environments

Compensation of uncertainties with inter-temporal measurements.

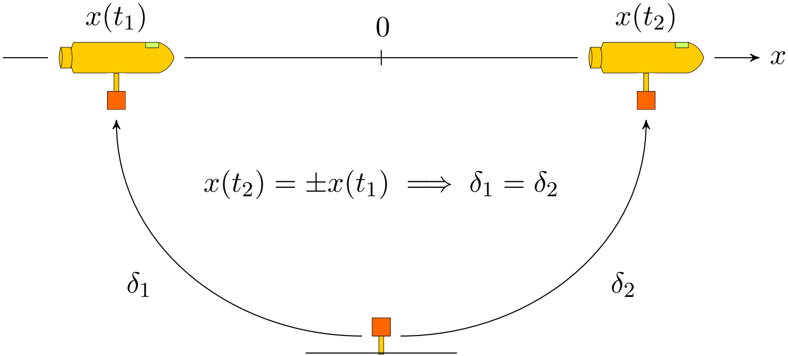


Figure: AUV's localization with an acoustic beacon on the seabed

Introduction

Inter-temp. measurements and set-membership methods

Considering the inter-temporal equation:

$$h\left(\mathbf{x}(t_1), \mathbf{x}(t_2)\right) = 0 \implies y(t_1) = y(t_2)$$

Introduction

Inter-temp. measurements and set-membership methods

Considering the inter-temporal equation:

$$h\left(\mathbf{x}(t_1),\mathbf{x}(t_2)\right)=0 \implies y(t_1)=y(t_2)$$

Two identical measurements do not lead to equivalent states:

$$y(t_1)=y(t_2) \;\not\Longrightarrow\; h\left(\mathbf{x}(t_1),\mathbf{x}(t_2)\right)$$

Introduction

Inter-temp. measurements and set-membership methods

Considering the inter-temporal equation:

$$h\left(\mathbf{x}(t_1),\mathbf{x}(t_2)\right)=0 \implies y(t_1)=y(t_2)$$

Two identical measurements do not lead to equivalent states:

$$y(t_1)=y(t_2) \;\not\Longrightarrow\; h\left(\mathbf{x}(t_1),\mathbf{x}(t_2)\right)$$

However, two different measurements lead to different states:

$$y(t_1)\neq y(t_2) \implies h\left(\mathbf{x}(t_1),\mathbf{x}(t_2)\right)\neq 0$$

Introduction

Inter-temp. measurements and set-membership methods

Considering the inter-temporal equation:

$$h\left(\mathbf{x}(t_1), \mathbf{x}(t_2)\right) = 0 \implies y(t_1) = y(t_2)$$

Two identical measurements do not lead to equivalent states:

$$y(t_1) = y(t_2) \;\not\Longrightarrow\; h\left(\mathbf{x}(t_1), \mathbf{x}(t_2)\right)$$

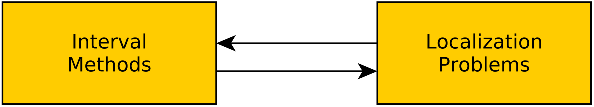
However, two different measurements lead to different states:

$$y(t_1) \neq y(t_2) \implies h\left(\mathbf{x}(t_1), \mathbf{x}(t_2)\right) \neq 0$$

Set-membership methods is a relevant tool to consider this last relation, relying on bounds and envelope of solutions.

Introduction

Objective and Motivations



Section 2

Interval Analysis

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
- $[\mathbf{x}] \in \mathbb{IR}^n$

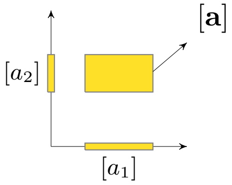


Figure: a box $[\mathbf{a}] \in \mathbb{IR}^2$

Interval Analysis

Interval Analysis

Using intervals to enclose measurements in a guaranteed way:

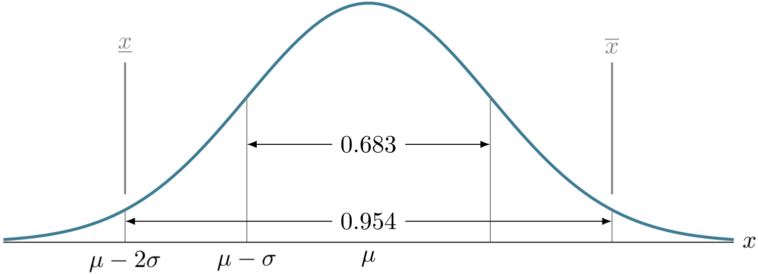


Figure: An interval $[x] = [\underline{x}, \overline{x}]$ computed from a Gaussian distribution to guarantee a 95% confidence rate over a measurement μ.

Interval Analysis

Interval Analysis

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

Interval Analysis

Set-membership state estimation

Recalled robot $\mathcal{R}$'s state equations:

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

Inputs and measurements given by sensors with known uncertainties ; initial state $\mathbf{x}_0$ bounded:

$$\mathbf{u} \in [\mathbf{u}] \ \ , \ \ y \in [y] \ \ , \ \ \mathbf{x}_0 \in [\mathbf{x}_0]$$

Consequently with interval arithmetic, other variables contained in intervals:

$$\mathbf{x} \in [\mathbf{x}] \ \ , \ \ \dot{\mathbf{x}} \in [\dot{\mathbf{x}}]$$

Values evolving with time are called **trajectories** and enclosed by **tubes**.

Section 3

Tubes as Envelopes of Trajectories

Tubes as Envelopes of Trajectories

Tubes: definition

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

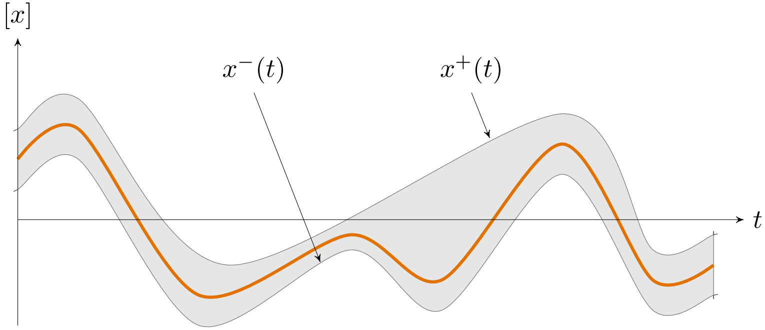


Figure: tube $[x](\cdot)$ enclosing an uncertain trajectory $x^*(\cdot)$

Tubes: arithmetic and contractors

Example:

Tube arithmetic makes it possible to compute the following tubes:

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

Tubes as Envelopes of Trajectories

Tubes: arithmetic and contractors

Example:

Tube arithmetic makes it possible to compute the following tubes:

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

Contractors for tubes:

To each primitive constraint on trajectories, tubes are contracted without removing any feasible solution.

Tubes as Envelopes of Trajectories

Tubes: minimal and non-minimal contractors

Example:

The minimal contractor associated to the constraint $a(\cdot) = x(\cdot) + y(\cdot)$:

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

Tubes as Envelopes of Trajectories

Tubes: minimal and non-minimal contractors

Example:
The minimal contractor associated to the constraint $a(\cdot) = x(\cdot) + y(\cdot)$:

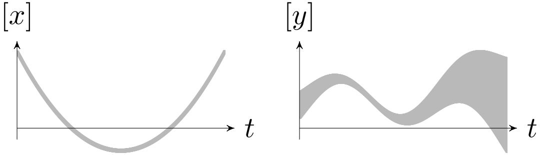
$$\begin{pmatrix} [a](\cdot) \\ [x](\cdot) \\ [y](\cdot) \end{pmatrix} \mapsto \begin{pmatrix} [a](\cdot) \cap ([x](\cdot) + [y](\cdot)) \\ [x](\cdot) \cap ([a](\cdot) - [y](\cdot)) \\ [y](\cdot) \cap ([a](\cdot) - [x](\cdot)) \end{pmatrix}$$

Example:
The non-minimal contractor associated to the constraint $c(\cdot) = \int_0 x(\tau)d\tau$:

$$\begin{pmatrix} [x](\cdot) \\ [c](\cdot) \end{pmatrix} \mapsto \begin{pmatrix} [x](\cdot) \\ [c](\cdot) \cap \int_0 [x](\tau)\, d\tau \end{pmatrix}$$

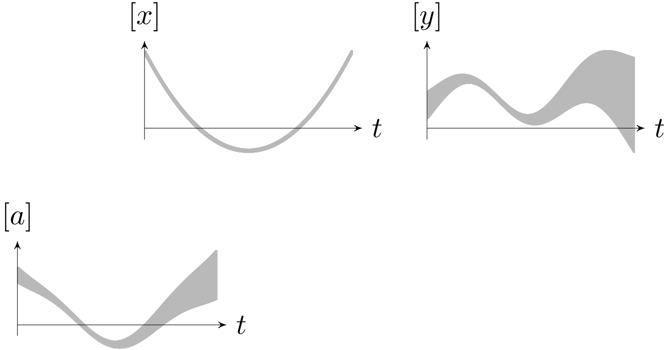
Tubes as Envelopes of Trajectories

Tubes programming: example



Tubes as Envelopes of Trajectories

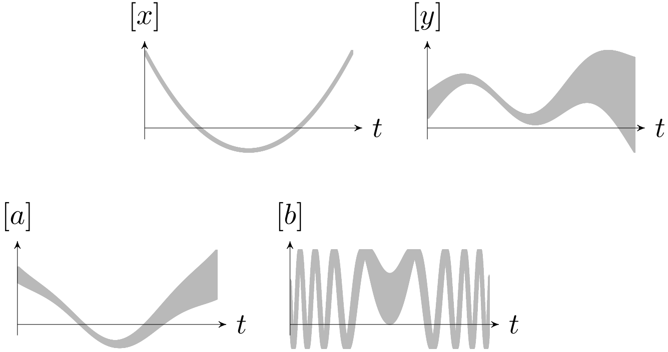
Tubes programming: example



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

Tubes as Envelopes of Trajectories

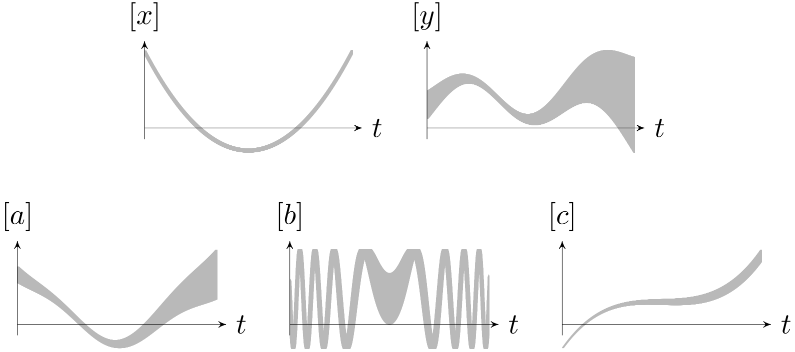
Tubes programming: example



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

Tubes as Envelopes of Trajectories

Tubes programming: example



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

Section 4

Contributions

Contributions

1^{st} contribution of this PhD

Guaranteed computation of robots trajectories.
We introduce a new *contractor* for tubes, to deal with differential equations such as:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t)$$

This is of high interest for robotics applications in order to:

- work with sets of trajectories
- consider differential and non-linear equations
- enclose data-sets in a reliable way

Contributions

1^{st} contribution of this PhD

Application on a real sea mission (Daurade AUV)

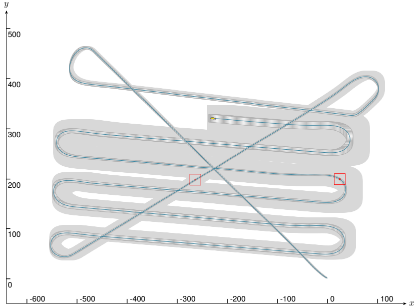


Figure: State estimation of an Autonomous Underwater Vehicle with tubes (top view). Firstly, dead-recknoning with DVL measurements (pictured in light gray). Secondly, localization thanks to two USBL values (final estimation depicted with boxes, measurements displayed in red).

Contributions

2^{nd} contribution of this PhD

Dealing with time uncertainties.

We introduce a new *contractor* for tubes, to deal with observations involving strong temporal uncertainties:

$$\begin{cases} y_1 & = & x(t_1) \\ t_1 & \in & [t_1] \\ y_1 & \in & [y_1] \\ x(\cdot) & \in & [x](\cdot) \end{cases}$$

This provides:

- a reliable tool to contract a tube
- a new approach to consider time uncertainties

Contributions

2^{nd} contribution of this PhD

Contractor based on the observation $[y_1]$ made at time $[t_1]$.

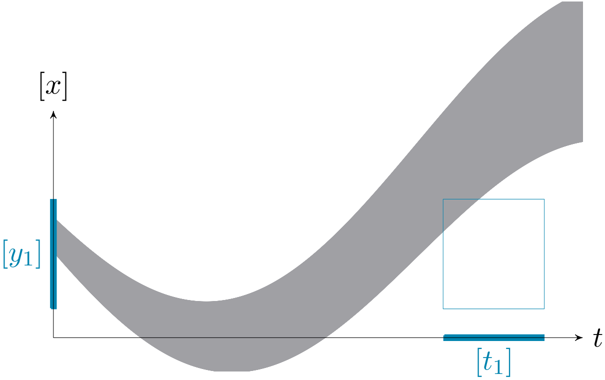


Figure: tube $[x](\cdot)$ before contraction

Contributions

2^{nd} contribution of this PhD

Contractor based on the observation $[y_1]$ made at time $[t_1]$.

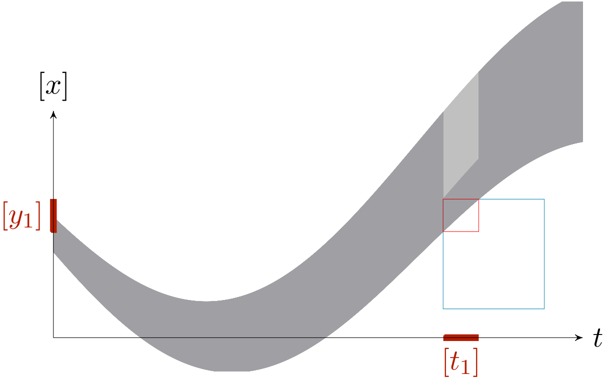


Figure: contraction of tube $[x](\cdot)$ and both $[y_1]$ and $[t_1]$

Contributions

2^{nd} contribution of this PhD

Contractor based on the observation $[y_1]$ made at time $[t_1]$.

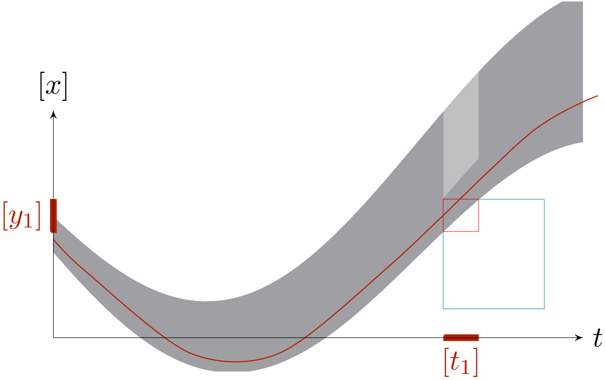


Figure: contraction of tube $[x](\cdot)$ and both $[y_1]$ and $[t_1]$

Contributions

2^{nd} contribution of this PhD

Contractor based on the observation $[y_1]$ made at time $[t_1]$.

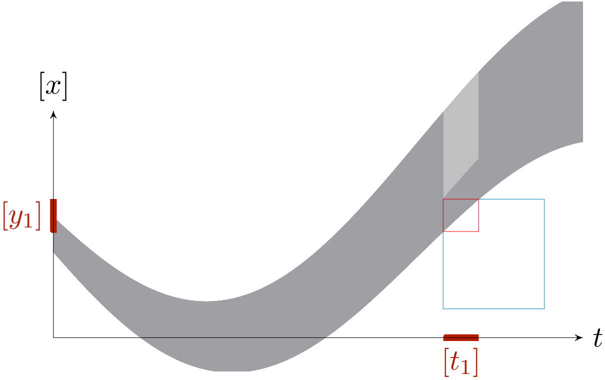


Figure: contraction of tube $[x](\cdot)$ and both $[y_1]$ and $[t_1]$

Contributions

2^{nd} contribution of this PhD

Contractor based on the observation $[y_1]$ made at time $[t_1]$.

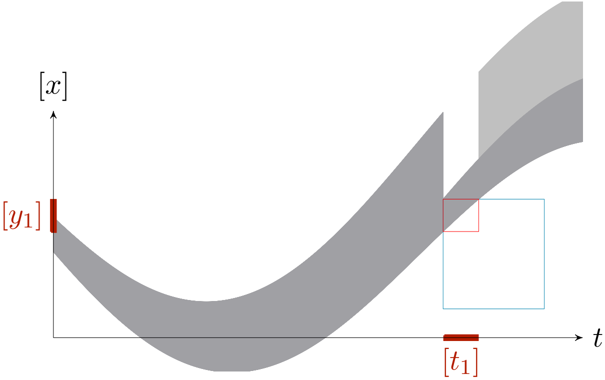


Figure: tube contraction in forward

Contributions

2^{nd} contribution of this PhD

Contractor based on the observation $[y_1]$ made at time $[t_1]$.

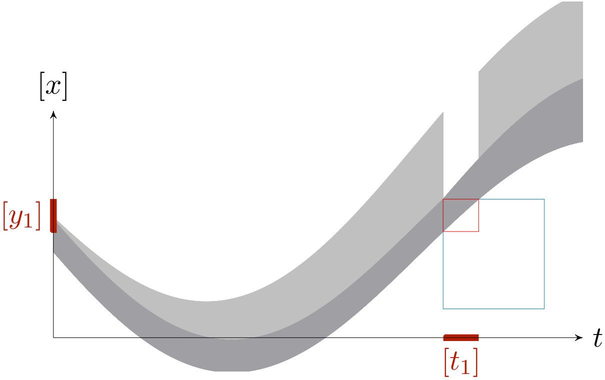


Figure: tube contraction in forward/backward

Contributions

3rd contribution of this PhD

Loop-based localization method.
We introduce a new state estimation method for AUVs:

$$\left\{\begin{array}{rcl}\dot{\mathbf{x}}(t) & = & \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t))\\ y(t) & = & g(\mathbf{x}(t))\end{array}\right.$$

Considering the inter-temporal equation formally defined by:

$$y(t_1) \neq y(t_2) \implies \mathbf{x}(t_1) \neq \mathbf{x}(t_2)$$

This provides:

- a competitive method for underwater localization in wide unstructured environments

Contributions

3rd contribution of this PhD

Figure: DAURADE AUV managed by *DGA Techniques Navales Brest* and the *Service Hydrographique et Océanographique de la Marine*, during an experiment in the Rade de Brest, October 2015.

Contributions

3rd contribution of this PhD

Application on a real sea mission (Daurade AUV)

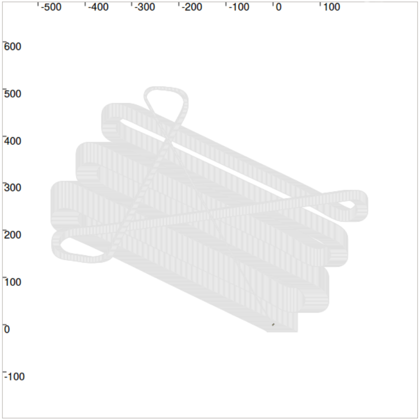


Figure: Loop-based localization applied on a real dataset – robot localization with bathymetric measurements

Contributions

3^{rd} contribution of this PhD

Application on a real sea mission (Daurade AUV)

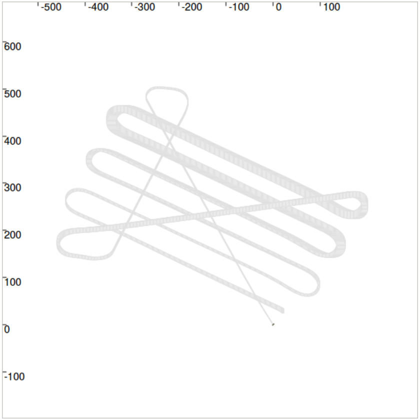


Figure: Loop-based localization applied on a real dataset – robot localization with bathymetric measurements

Contributions

Other contributions of this PhD

Some on-going works:

- **wreck based localization**
  time uncertainties, side scan sonar, boats wrecks
- **robot localization in an unknown but symmetric environment**
  symmetries, t-planes, unknown sound celerity profiles
- **new optimal loop existence test**
  topological degree test, collaboration with Peter Franek
- **range-only localization with unknown bias**
  method for both robot localization and bias estimation
- **robot collaborative exploration: walking strategy**
  stack of AUVs, anchors, explorers, steps

Conclusion

3 equations, 3 main contributions:

$$
\begin{cases}
\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) & \text{(navigation)} \\
y(t) = g(\mathbf{x}(t)) & \text{(measurements)} \\
y(t_1) \neq y(t_2) \implies h\left(\mathbf{x}(t_1), \mathbf{x}(t_2)\right) \neq 0 & \text{(inter-temporality)}
\end{cases}
$$

Applied on real underwater robot localization problems.