Constraint programming for mobile robotics

Simon Rohou

ENSTA Bretagne, Lab-STICC, UMR CNRS 6285, Brest, France

École Centrale, Nantes
2nd December 2019

Reliability

How to reliably represent irrational numbers?

$$\pi = 3.14159265359...$$

Reliability

How to reliably represent irrational numbers?

$$\pi = 3.14159265359...$$

$$^{223}/_{71} < \pi < {^{22}/_{7}}$$

Reliability

How to reliably represent irrational numbers?

$$\pi = 3.14159265359...$$

$$^{223}/_{71} < \pi < {}^{22}/_{7}$$

$$\pi \in \left[{}^{223}/_{71}, {}^{22}/_{7}\right]$$

Reliability

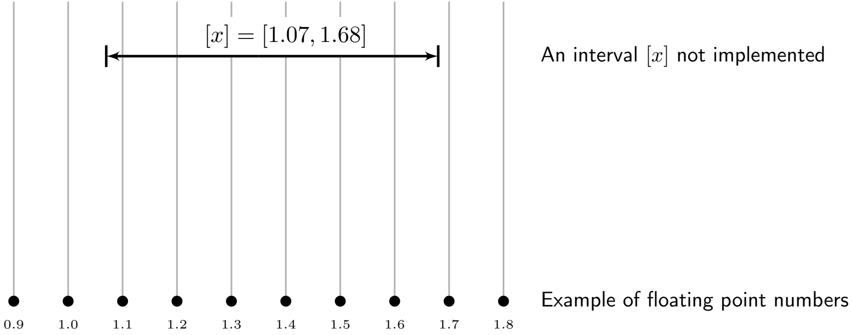
How to reliably represent floating point numbers?

$$0.1$$

Reliability

How to reliably represent floating point numbers?

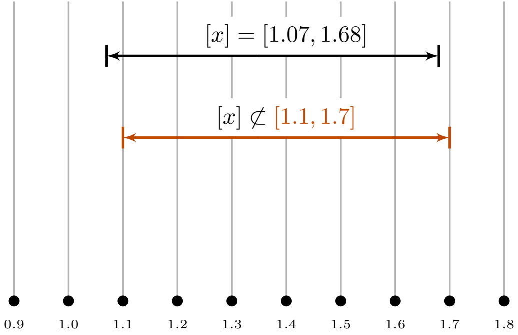
$$[1.07, 1.68]$$



Reliability

How to reliably represent floating point numbers?

$$[1.07, 1.68]$$



An interval $[x]$ not implemented

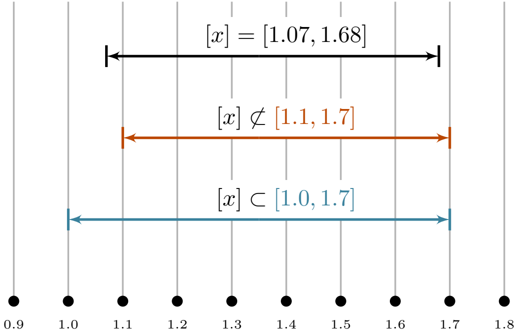
Automatic representation of $[x]$ (nearest float. point: no reliability)

Example of floating point numbers

Reliability

How to reliably represent floating point numbers?

$$[1.07, 1.68]$$



An interval $[x]$ not implemented

Automatic representation of $[x]$
(nearest float. point: no reliability)

Outward rounding of $[x]$
(reliable implementation)

Example of floating point numbers

Reliability

What is the benefit for robotics?

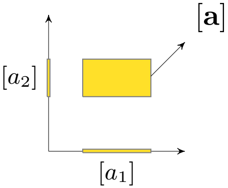
Interval Analysis

An interval $[x]$:

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

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

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



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

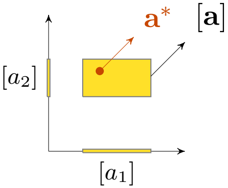
Interval Analysis

An interval $[x]$:

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

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

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



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

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

Interval Analysis

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

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

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

Mobile robotics

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

Special thanks to DGA-TN Brest (formerly GESMA)

Uncertainties as sets

Example of **range-only** robot localization (three beacons):

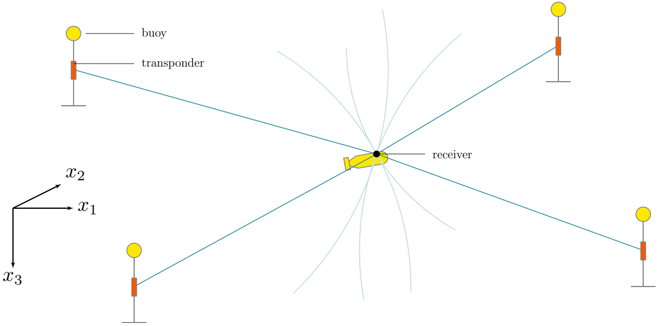
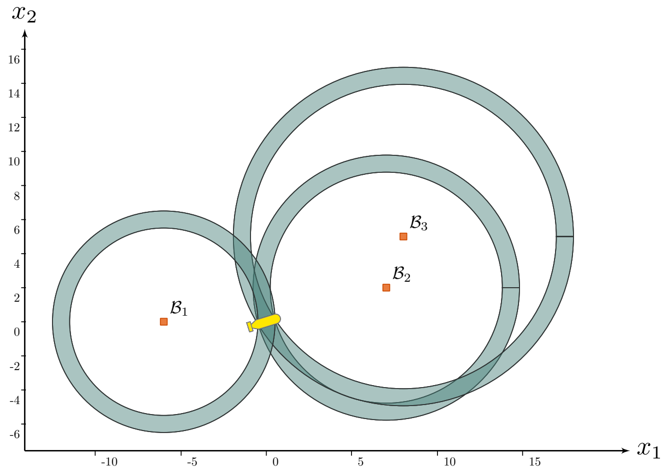


Illustration of Long BaseLine (LBL) positioning

Uncertainties as sets

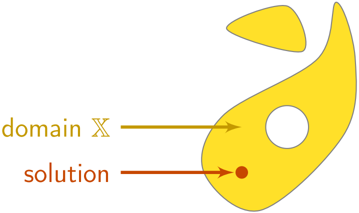
Example of **range-only** robot localization (three beacons):



LBL positioning with bounded uncertainties

Constraint programming: overall concept

- system described by a *constraint network*
- **variables** belonging to **domains** $\mathbb{X}$



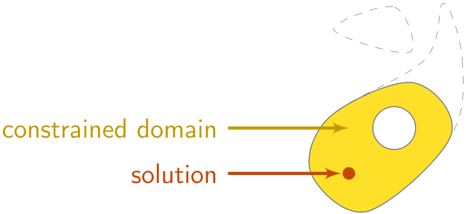
Constraint network:

$$\begin{cases} \textbf{Variables: } \mathbf{x} \\ \textbf{Constraints:} \\ \\ \\ \textbf{Domains: } \mathbb{X} \end{cases}$$

Contractor Programming
Chabert, Jaulin *Artifical Intelligence*, 2009

Constraint programming: overall concept

- system described by a *constraint network*
- **variables** belonging to **domains** $\mathbb{X}$
- continuous **constraints** $\mathcal{L}$: non-linear equations, inequalities, ...



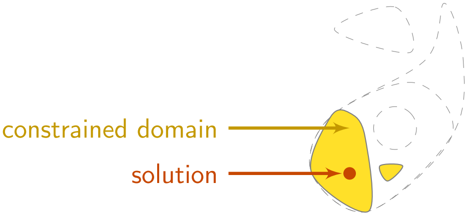
Constraint network:

$$\left\{ \begin{array}{l} \textbf{Variables: } \mathbf{x} \\ \textbf{Constraints:} \\ \quad 1.\ \mathcal{L}_1(\mathbf{x}) \\ \\ \textbf{Domains: } \mathbb{X} \end{array} \right.$$

Contractor Programming
Chabert, Jaulin *Artifical Intelligence*, 2009

Constraint programming: overall concept

- system described by a *constraint network*
- **variables** belonging to **domains** $\mathbb{X}$
- continuous **constraints** $\mathcal{L}$: non-linear equations, inequalities, ...



Constraint network:

Variables: $\mathbf{x}$

Constraints:

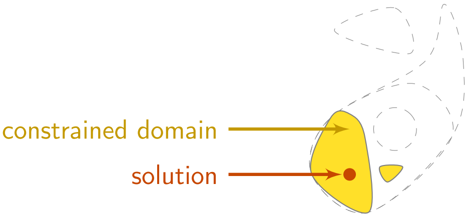
1. $\mathcal{L}_1(\mathbf{x})$
2. $\mathcal{L}_2(\mathbf{x})$

Domains: $\mathbb{X}$

Contractor Programming
Chabert, Jaulin *Artifical Intelligence*, 2009

Constraint programming: overall concept

- system described by a *constraint network*
- **variables** belonging to **domains** $\mathbb{X}$
- continuous **constraints** $\mathcal{L}$: non-linear equations, inequalities, ...



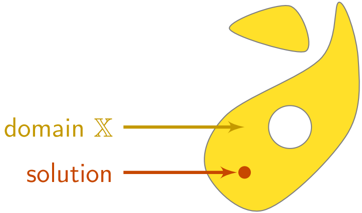
Constraint network:

$$\left\{\begin{array}{l}\textbf{Variables: } \mathbf{x} \\ \textbf{Constraints:} \\ \quad 1.\ \mathcal{L}_1(\mathbf{x}) \\ \quad 2.\ \mathcal{L}_2(\mathbf{x}) \\ \quad 3.\ \ldots \\ \textbf{Domains: } \mathbb{X}\end{array}\right.$$

Contractor Programming
Chabert, Jaulin *Artifical Intelligence*, 2009

Constraint programming: overall concept

- system described by a *constraint network*
- **variables** belonging to **domains** $\mathbb{X}$
- continuous **constraints** $\mathcal{L}$: non-linear equations, inequalities, ...



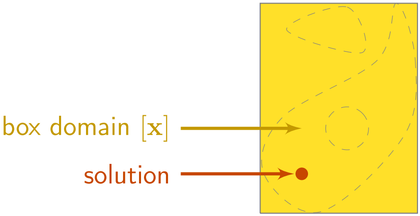
Constraint network:

$$\left\{\begin{array}{l}\textbf{Variables: } \mathbf{x}\\ \textbf{Constraints:}\\ \quad 1.\ \mathcal{L}_1(\mathbf{x})\\ \quad 2.\ \mathcal{L}_2(\mathbf{x})\\ \quad 3.\ \dots\\ \textbf{Domains: } \mathbb{X}\end{array}\right.$$

Contractor Programming
Chabert, Jaulin *Artifical Intelligence*, 2009

Constraint programming: overall concept

- system described by a *constraint network*
- **variables** belonging to **domains** $\mathbb{X}$
- continuous **constraints** $\mathcal{L}$: non-linear equations, inequalities, ...
- representable domains: *e.g.* boxes $[\mathbf{x}]$



Constraint network:

Variables: $\mathbf{x}$

Constraints:

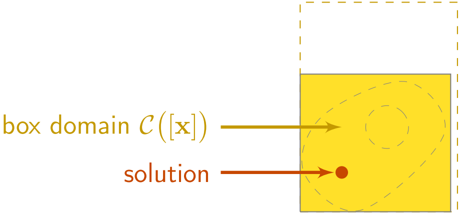
1. $\mathcal{L}_1(\mathbf{x})$
2. $\mathcal{L}_2(\mathbf{x})$
3. ...

Domains: $[\mathbf{x}]$

Contractor Programming
Chabert, Jaulin *Artifical Intelligence*, 2009

Constraint programming: overall concept

- system described by a *constraint network*
- **variables** belonging to **domains** $\mathbb{X}$
- continuous **constraints** $\mathcal{L}$: non-linear equations, inequalities, ...
- representable domains: *e.g.* boxes $[\mathbf{x}]$
- resolution by **contractors**, $\mathcal{C}_{\mathcal{L}}([\mathbf{x}])$



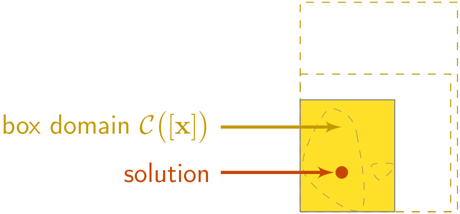
Constraint network:

$$\begin{cases} \textbf{Variables: } \mathbf{x} \\ \textbf{Constraints:} \\ \quad 1.\ \mathcal{L}_1(\mathbf{x}) \\ \quad 2.\ \mathcal{L}_2(\mathbf{x}) \\ \quad 3.\ \dots \\ \textbf{Domains: } [\mathbf{x}] \end{cases}$$

Contractor Programming
Chabert, Jaulin *Artifical Intelligence*, 2009

Constraint programming: overall concept

- system described by a *constraint network*
- **variables** belonging to **domains** $\mathbb{X}$
- continuous **constraints** $\mathcal{L}$: non-linear equations, inequalities, ...
- representable domains: *e.g.* boxes $[\mathbf{x}]$
- resolution by **contractors**, $\mathcal{C}_{\mathcal{L}}([\mathbf{x}])$



Constraint network:

Variables: $\mathbf{x}$

Constraints:

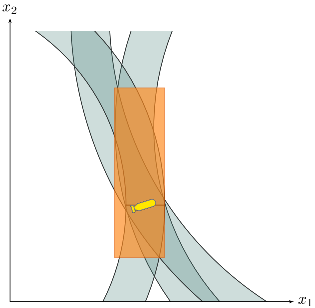
1. $\mathcal{L}_1(\mathbf{x})$
2. $\mathcal{L}_2(\mathbf{x})$
3. ...

Domains: $[\mathbf{x}]$

Contractor Programming
Chabert, Jaulin *Artifical Intelligence*, 2009

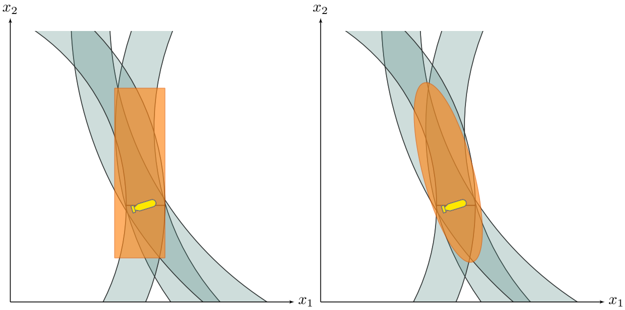
Wrappers

- box



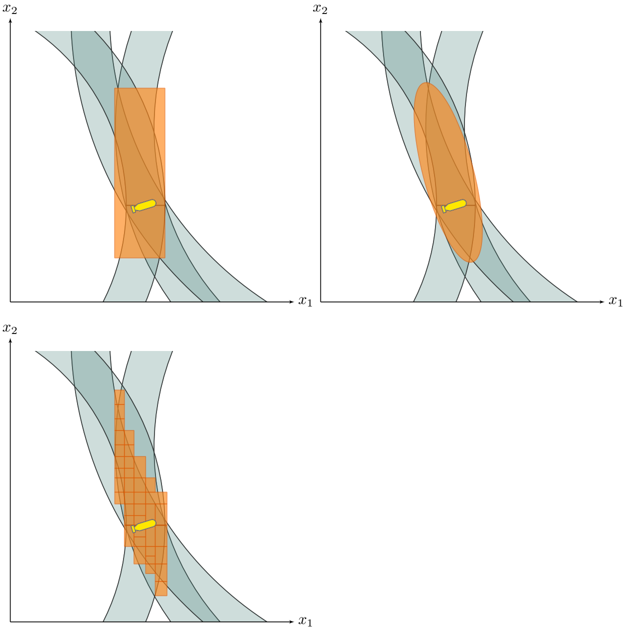
Wrappers

- box
- ellipse



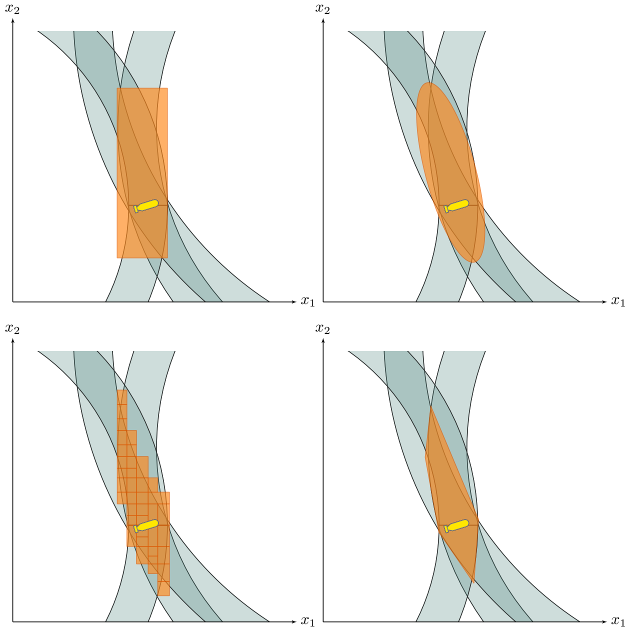
Wrappers

- box
- ellipse
- paving



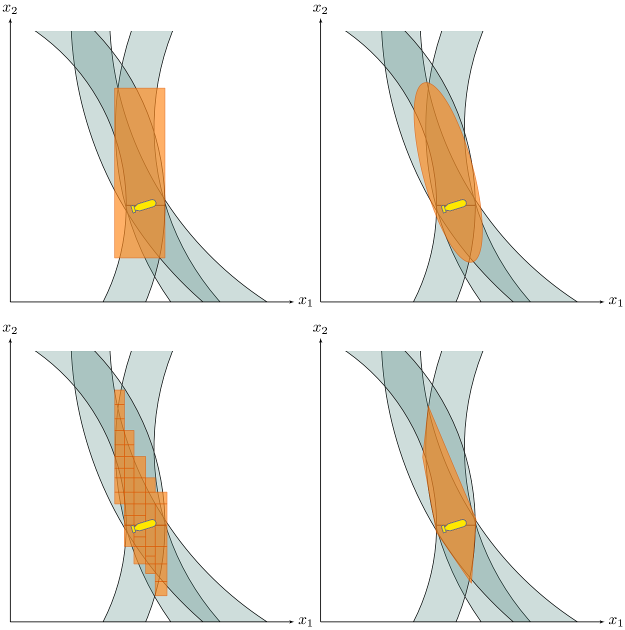
Wrappers

- box
- ellipse
- paving
- polygon



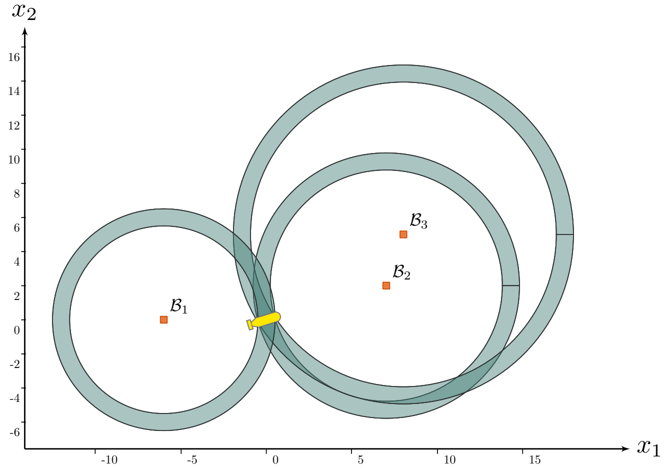
Wrappers

- box
- ellipse
- paving
- polygon
- ...



Set-membership state estimation

Three observations $\rho^{(k)}$ from three beacons $\mathcal{B}^{(k)}$:



Constraints

Observation constraint, links a measurement $\rho^{(k)}$ to the state $\mathbf{x}$:

$$\rho^{(k)} = \sqrt{\left(x_1 - \mathcal{B}_1^{(k)}\right)^2 + \left(x_2 - \mathcal{B}_2^{(k)}\right)^2}.$$

Constraints

Observation constraint, links a measurement $\rho^{(k)}$ to the state $\mathbf{x}$:

$$\mathcal{L}_g^{(k)} : \rho^{(k)} = \sqrt{\left(x_1 - \mathcal{B}_1^{(k)}\right)^2 + \left(x_2 - \mathcal{B}_2^{(k)}\right)^2}.$$

Problem synthesized as a **constraint network**:

Variables: $\mathbf{x}$, $\rho^{(1)}$, $\rho^{(2)}$, $\rho^{(3)}$

Constraints:

1. $\mathcal{L}_g^{(1)}\left(\mathbf{x}, \rho^{(1)}\right)$
2. $\mathcal{L}_g^{(2)}\left(\mathbf{x}, \rho^{(2)}\right)$
3. $\mathcal{L}_g^{(3)}\left(\mathbf{x}, \rho^{(3)}\right)$

Domains: $[\mathbf{x}]$, $[\rho^{(1)}]$, $[\rho^{(2)}]$, $[\rho^{(3)}]$

Constraints

Observation constraint, links a measurement $\rho^{(k)}$ to the state $\mathbf{x}$:

$$\mathcal{L}_g^{(k)} : \rho^{(k)} = \sqrt{\left(x_1 - \mathcal{B}_1^{(k)}\right)^2 + \left(x_2 - \mathcal{B}_2^{(k)}\right)^2}.$$

Problem synthesized as a **constraint network**:

$$\begin{cases}
\textbf{Variables: } \mathbf{x},\ \rho^{(1)},\ \rho^{(2)},\ \rho^{(3)} \\
\textbf{Constraints:} \\
\quad 1.\ \mathcal{L}_g^{(1)}\left(\mathbf{x}, \rho^{(1)}\right) \quad \Longrightarrow \quad \mathcal{C}_g^{(1)}\left([\mathbf{x}], [\rho^{(1)}]\right) \\
\quad 2.\ \mathcal{L}_g^{(2)}\left(\mathbf{x}, \rho^{(2)}\right) \quad \Longrightarrow \quad \mathcal{C}_g^{(2)}\left([\mathbf{x}], [\rho^{(2)}]\right) \\
\quad 3.\ \mathcal{L}_g^{(3)}\left(\mathbf{x}, \rho^{(3)}\right) \quad \Longrightarrow \quad \mathcal{C}_g^{(3)}\left([\mathbf{x}], [\rho^{(3)}]\right) \\
\textbf{Domains: } [\mathbf{x}],\ [\rho^{(1)}],\ [\rho^{(2)}],\ [\rho^{(3)}]
\end{cases}$$

Contractors to apply constraints

Example: **decomposition** of the observation constraint:

$$\mathcal{L}_g^{(k)} : \rho^{(k)} = \sqrt{\left(x_1 - \mathcal{B}_1^{(k)}\right)^2 + \left(x_2 - \mathcal{B}_2^{(k)}\right)^2}$$

Contractors to apply constraints

Example: **decomposition** of the observation constraint:

$$\mathcal{L}_g^{(k)} : \rho^{(k)} = \sqrt{\left(x_1 - \mathcal{B}_1^{(k)}\right)^2 + \left(x_2 - \mathcal{B}_2^{(k)}\right)^2}$$

$$\Leftrightarrow \left\{ \begin{array}{rcl} a & = & x_1 - \mathcal{B}_1^{(k)} \\ b & = & x_2 - \mathcal{B}_2^{(k)} \\ c & = & a^2 \\ d & = & b^2 \\ e & = & c + d \\ \rho^{(k)} & = & \sqrt{e} \end{array} \right.$$

Contractors to apply constraints

Example: **decomposition** of the observation constraint:

$$\mathcal{L}_g^{(k)} : \rho^{(k)} = \sqrt{\left(x_1 - \mathcal{B}_1^{(k)}\right)^2 + \left(x_2 - \mathcal{B}_2^{(k)}\right)^2}$$

$$\Leftrightarrow \left\{ \begin{array}{rcl} a & = & x_1 - \mathcal{B}_1^{(k)} \\ b & = & x_2 - \mathcal{B}_2^{(k)} \\ c & = & a^2 \\ d & = & b^2 \\ e & = & c + d \\ \rho^{(k)} & = & \sqrt{e} \end{array} \right.$$

$$\left(\begin{array}{c} [e] \\ [c] \\ [d] \end{array} \right) \xmapsto{\mathcal{C}_+} \left(\begin{array}{c} [e] \cap ([c] + [d]) \\ [c] \cap ([e] - [d]) \\ [d] \cap ([e] - [c]) \end{array} \right)$$

Contractors to apply constraints

Example: **decomposition** of the observation constraint:

$$\mathcal{L}_g^{(k)} : \rho^{(k)} = \sqrt{\left(x_1 - \mathcal{B}_1^{(k)}\right)^2 + \left(x_2 - \mathcal{B}_2^{(k)}\right)^2}$$

$$\Leftrightarrow \left\{ \begin{array}{rclcl} a & = & x_1 - \mathcal{B}_1^{(k)} & \quad & \mathcal{C}_-([a],[x_1],[\mathcal{B}_1^{(k)}]) \\ b & = & x_2 - \mathcal{B}_2^{(k)} & & \mathcal{C}_-([b],[x_2],[\mathcal{B}_2^{(k)}]) \\ c & = & a^2 & & \mathcal{C}_{.^2}([c],[a]) \\ d & = & b^2 & & \mathcal{C}_{.^2}([d],[b]) \\ e & = & c+d & & \mathcal{C}_+([e],[c],[d]) \\ \rho^{(k)} & = & \sqrt{e} & & \mathcal{C}_{\sqrt{\cdot}}([\rho^{(k)}],[e]) \end{array} \right.$$

$$\begin{pmatrix} [e] \\ [c] \\ [d] \end{pmatrix} \xmapsto{\mathcal{C}_+} \begin{pmatrix} [e] \cap ([c]+[d]) \\ [c] \cap ([e]-[d]) \\ [d] \cap ([e]-[c]) \end{pmatrix}$$

Contractors to apply constraints

Example: **decomposition** of the observation constraint:

$$\mathcal{L}_g^{(k)} : \rho^{(k)} = \sqrt{\left(x_1 - \mathcal{B}_1^{(k)}\right)^2 + \left(x_2 - \mathcal{B}_2^{(k)}\right)^2}$$

$$\Leftrightarrow \left\{ \begin{array}{rcl} a & = & x_1 - \mathcal{B}_1^{(k)} \\ b & = & x_2 - \mathcal{B}_2^{(k)} \\ c & = & a^2 \\ d & = & b^2 \\ e & = & c + d \\ \rho^{(k)} & = & \sqrt{e} \end{array} \qquad \begin{array}{l} \mathcal{C}_-([a],[x_1],[\mathcal{B}_1^{(k)}]) \\ \mathcal{C}_-([b],[x_2],[\mathcal{B}_2^{(k)}]) \\ \mathcal{C}_{.^2}([c],[a]) \\ \mathcal{C}_{.^2}([d],[b]) \\ \mathcal{C}_+([e],[c],[d]) \\ \mathcal{C}_{\sqrt{\cdot}}([\rho^{(k)}],[e]) \end{array} \right\} \Longrightarrow \mathcal{C}_g^{(k)}$$

$$\begin{pmatrix} [e] \\ [c] \\ [d] \end{pmatrix} \xmapsto{\mathcal{C}_+} \begin{pmatrix} [e] \cap ([c] + [d]) \\ [c] \cap ([e] - [d]) \\ [d] \cap ([e] - [c]) \end{pmatrix}$$

Contractor programming

Now: problem to be solved with a **set of contractors**:

$$\left\{\begin{array}{l}\textbf{Variables: } \mathbf{x},\ \rho^{(1)},\ \rho^{(2)},\ \rho^{(3)} \\[1ex] \textbf{Constraints:} \\ \quad 1.\ \mathcal{C}_g^{(1)}\left([\mathbf{x}],[\rho^{(1)}]\right) \\ \quad 2.\ \mathcal{C}_g^{(2)}\left([\mathbf{x}],[\rho^{(2)}]\right) \\ \quad 3.\ \mathcal{C}_g^{(3)}\left([\mathbf{x}],[\rho^{(3)}]\right) \\[1ex] \textbf{Domains: } [\mathbf{x}],\ [\rho^{(1)}],\ [\rho^{(2)}],\ [\rho^{(3)}]\end{array}\right.$$

Contractor programming

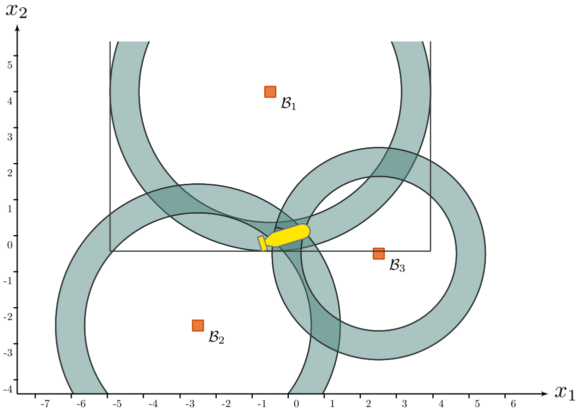
Now: problem to be solved with a **set of contractors**:

$$\begin{cases} \textbf{Variables: } \mathbf{x},\ \rho^{(1)},\ \rho^{(2)},\ \rho^{(3)} \\ \textbf{Constraints:} \\ \quad 1.\ \mathcal{C}_g^{(1)}\left([\mathbf{x}],[\rho^{(1)}]\right) \\ \quad 2.\ \mathcal{C}_g^{(2)}\left([\mathbf{x}],[\rho^{(2)}]\right) \\ \quad 3.\ \mathcal{C}_g^{(3)}\left([\mathbf{x}],[\rho^{(3)}]\right) \\ \textbf{Domains: } [\mathbf{x}],\ [\rho^{(1)}],\ [\rho^{(2)}],\ [\rho^{(3)}] \end{cases}$$

Initializations:

- $[\mathbf{x}] = [-\infty, \infty]^2$
- $[\rho^{(1)}]$, $[\rho^{(2)}]$, $[\rho^{(3)}]$ set from measurements

Fixed point propagations

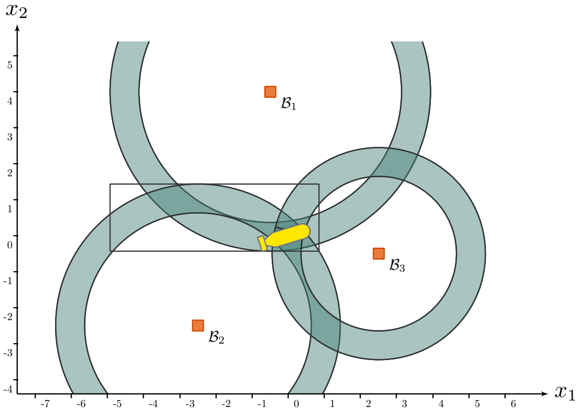


Applying $\mathcal{C}_g^{(1)}$.

Study of robust set estimation methods for a high integrity multi-sensor localization.

Vincent Drevelle *Thesis*, 2011

Fixed point propagations

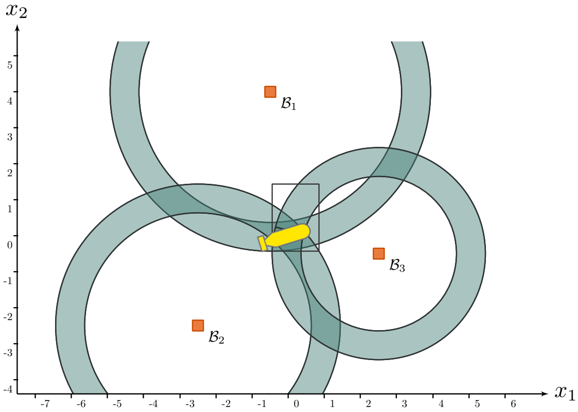


Applying $\mathcal{C}_g^{(2)}$.

Study of robust set estimation methods for a high integrity multi-sensor localization.

Vincent Drevelle *Thesis*, 2011

Fixed point propagations

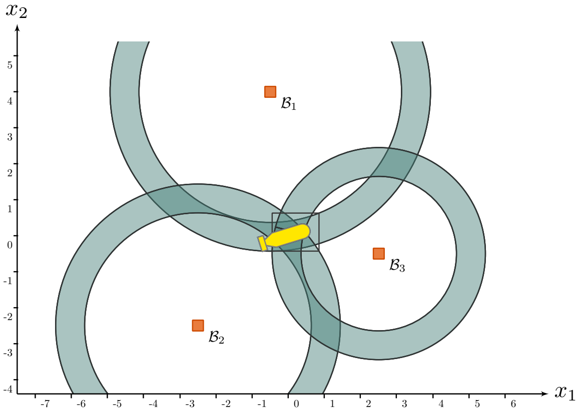


Applying $\mathcal{C}_g^{(3)}$.

Study of robust set estimation methods for a high integrity multi-sensor localization.

Vincent Drevelle *Thesis*, 2011

Fixed point propagations

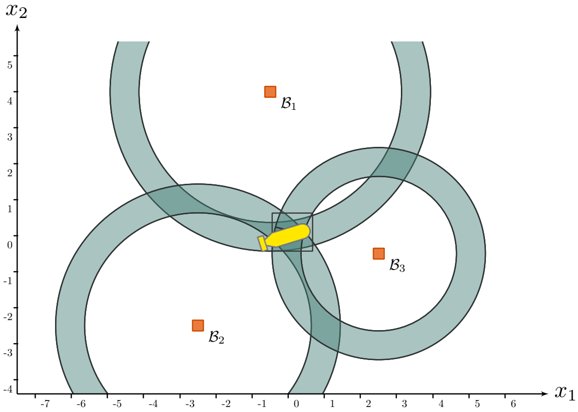


Applying $\mathcal{C}_g^{(1)}$.

■ Study of robust set estimation methods for a high integrity multi-sensor localization.

Vincent Drevelle *Thesis*, 2011

Fixed point propagations

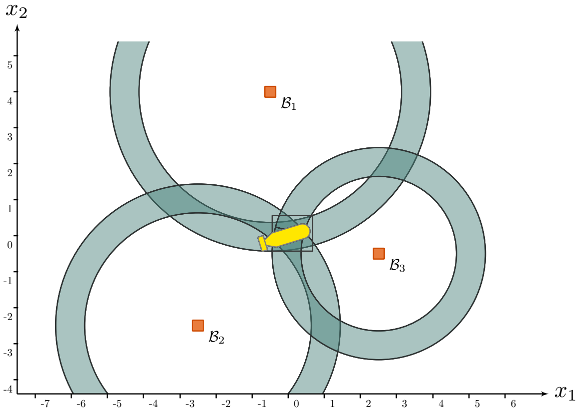


Applying $\mathcal{C}_g^{(2)}$.

■ Study of robust set estimation methods for a high integrity multi-sensor localization.

Vincent Drevelle *Thesis*, 2011

Fixed point propagations

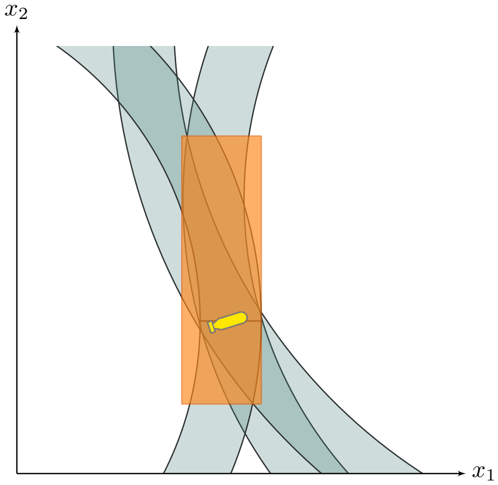


Fixed point reached.

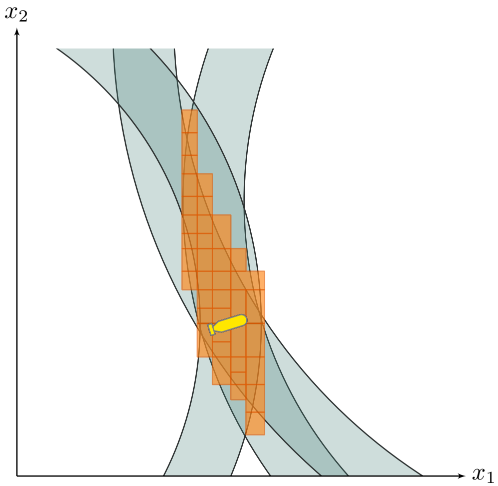
Study of robust set estimation methods for a high integrity multi-sensor localization.

Vincent Drevelle *Thesis*, 2011

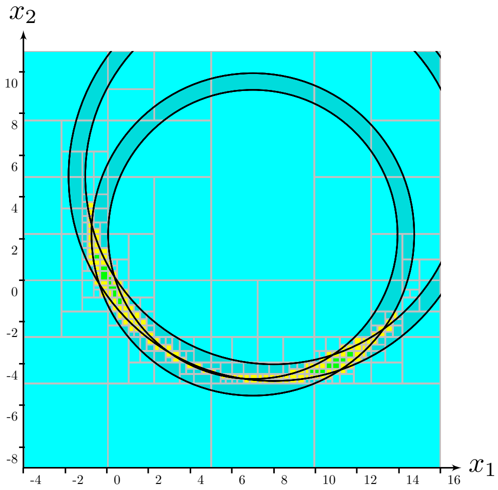
Sub-pavings



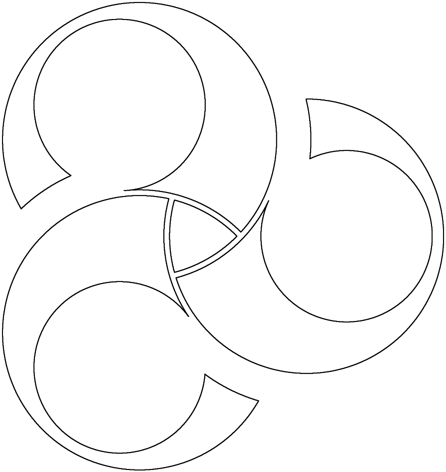
Sub-pavings



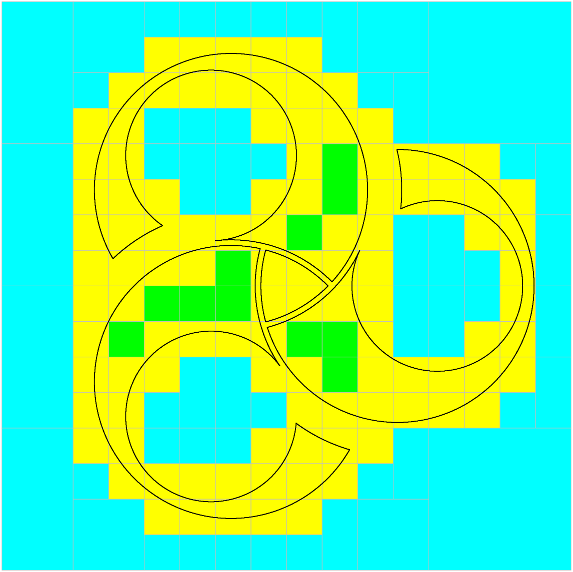
Sub-pavings



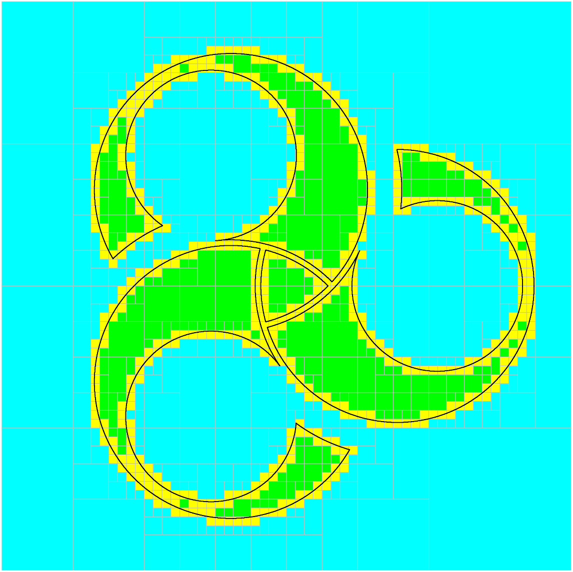
Sub-pavings: precision



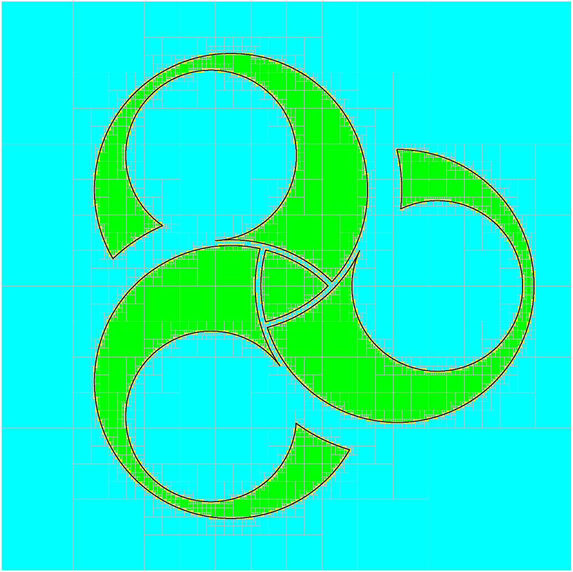
Sub-pavings: precision



Sub-pavings: precision



Sub-pavings: precision



Constraint programming for mobile robotics

Constraint programming coupled with **mobile robotics**:

- robot's state vector $\mathbf{x}$ to be estimated
- several proprioceptive/exteroceptive measurements

$\Longrightarrow$ more constraints than unknowns

Constraint programming for mobile robotics

Constraint programming coupled with **mobile robotics**:

- robot's state vector $\mathbf{x}$ to be estimated
- several proprioceptive/exteroceptive measurements

  $\Longrightarrow$ more constraints than unknowns

Assets:

- no need for linearization
- safety of systems:
  reliable outputs
- useful for numerical proofs

Constraint programming for mobile robotics

Constraint programming coupled with **mobile robotics**:

- robot's state vector $\mathbf{x}$ to be estimated
- several proprioceptive/exteroceptive measurements

$\Longrightarrow$ more constraints than unknowns

Assets:

- no need for linearization
- safety of systems: reliable outputs
- useful for numerical proofs

Drawbacks:

- unwanted pessimism
- sets as outputs

Sets from sensor data

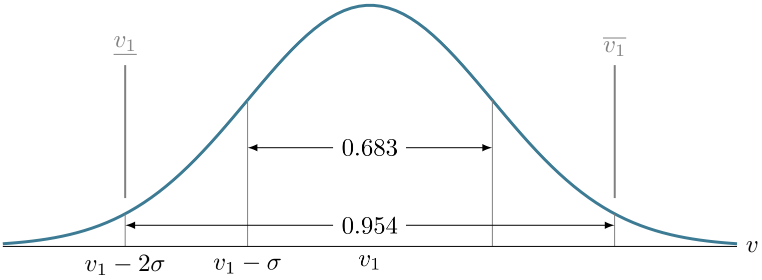
Sets from sensor data

Video

Sets from sensor data

Uncertainties:

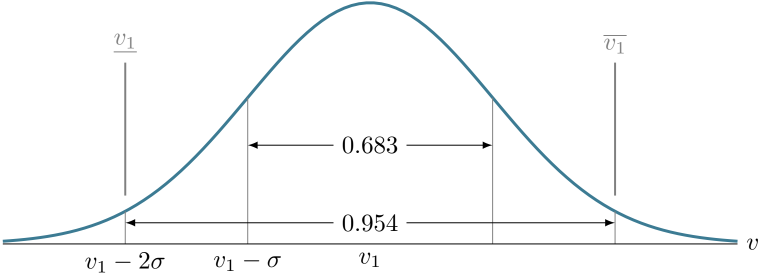
- datasheets $\implies$ standard deviation σ for each sensor
- 95% confidence rate: $v_1^* \in [v_1] = [v_1 - 2\sigma, v_1 + 2\sigma]$



Sets from sensor data

Uncertainties:

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

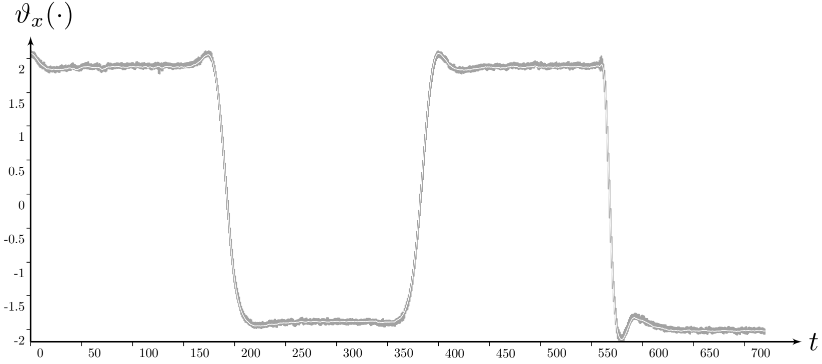


- uncertainties then reliably propagated in the system

  ex: $[x] + [y] = \left[\underline{x} + \underline{y}, \overline{x} + \overline{y}\right]$

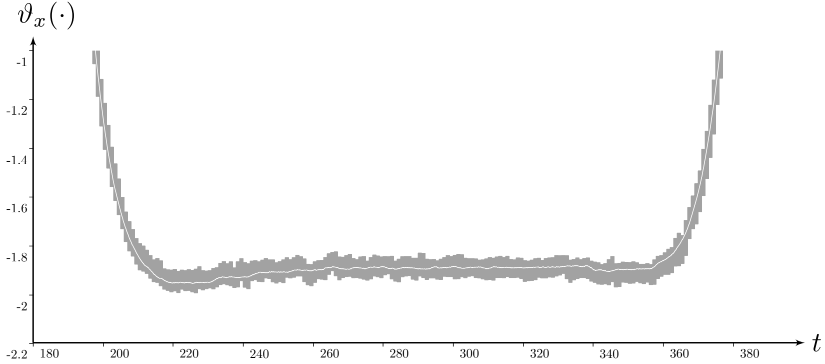
Example: velocity sensing

East velocity given by DVL + IMU:



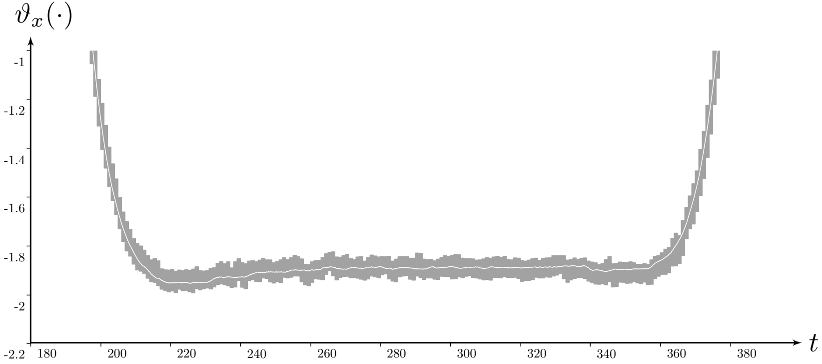
Example: velocity sensing

East velocity given by DVL + IMU (zoom):



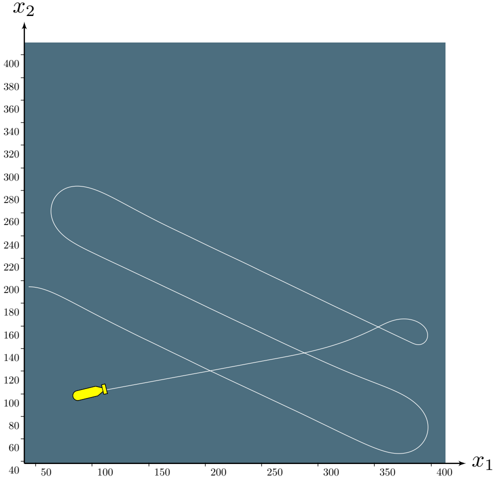
Example: velocity sensing

East velocity given by DVL + IMU (zoom):



- new variable: **trajectory** $x(\cdot)$
- new domain (set): **tube** $[x](\cdot)$, interval of trajectories

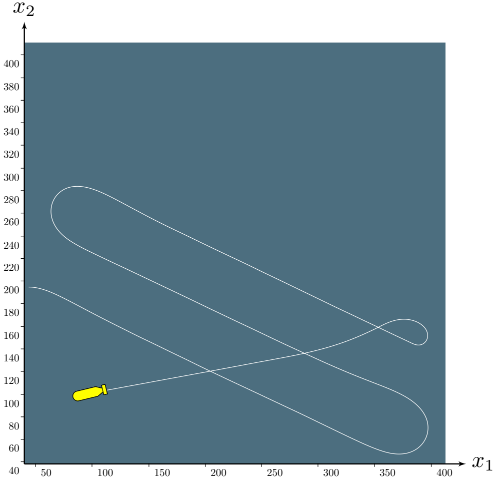
Dynamic state estimation



State estimation:

$$\left\{ \begin{array}{l} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) \\ \\ \end{array} \right.$$

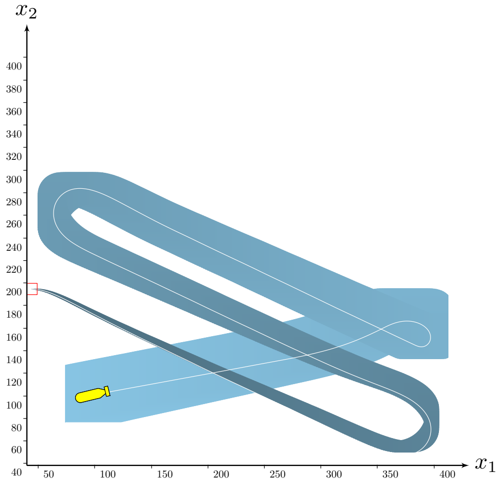
Dynamic state estimation



State estimation:

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

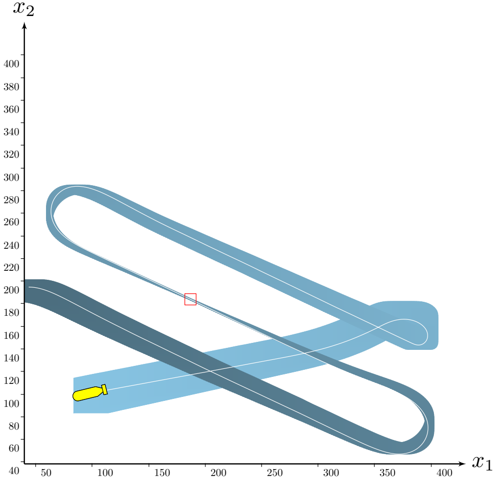
Dynamic state estimation



State estimation:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) \\ \dot{\mathbf{x}}(t) = \mathbf{v}(t) \\ \mathbf{x}(t_0) \in [\mathbf{x}_0] \end{cases}$$

Dynamic state estimation



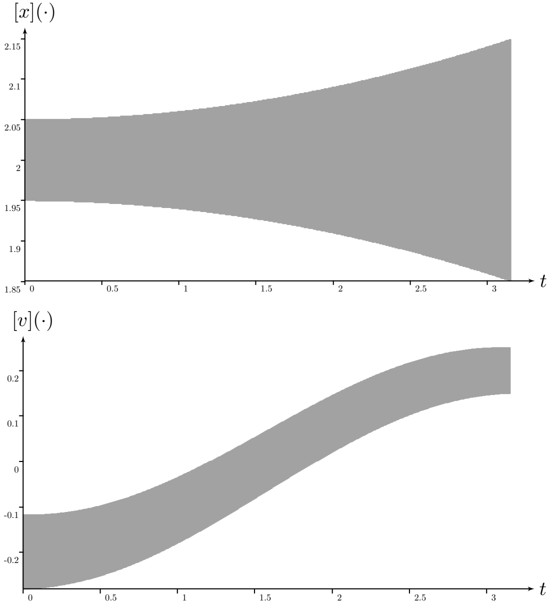
State estimation:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) \\ \dot{\mathbf{x}}(t) = \mathbf{v}(t) \\ \mathbf{x}(t_1) \in [\mathbf{x}_1] \end{cases}$$

Derivative constraint

Differential constraint:

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



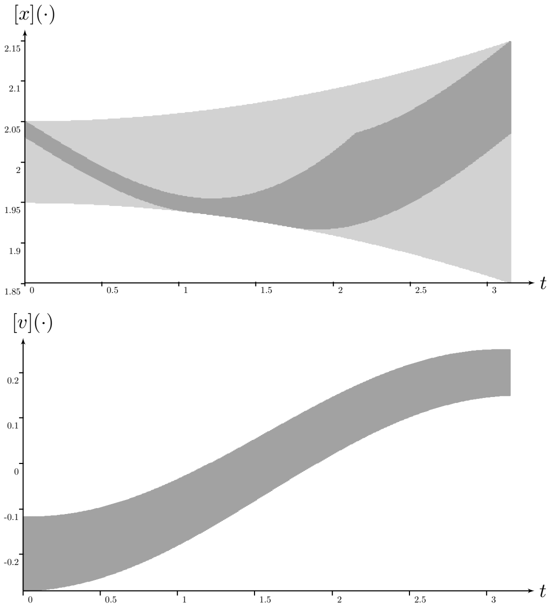
Derivative constraint

Differential constraint:

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

Contractor programming:

$$\mathcal{C}_{\frac{d}{dt}}\big([\mathbf{x}](\cdot), [\mathbf{v}](\cdot)\big)$$



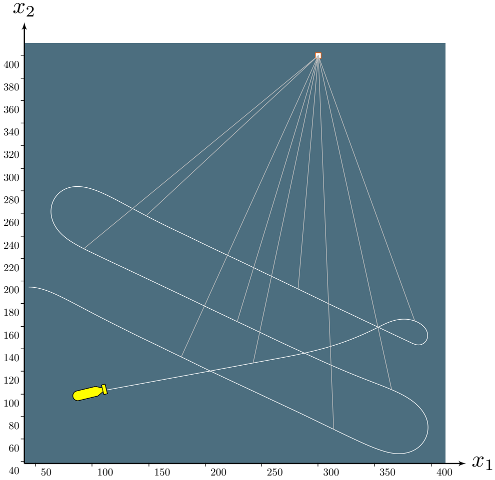
Guaranteed computation of robot trajectories

Rohou, Jaulin, Mihaylova, Le Bars, Veres

Robotics and Autonomous Systems, 2017

Dynamic state estimation

Considering **range-only measurements** from a known beacon.

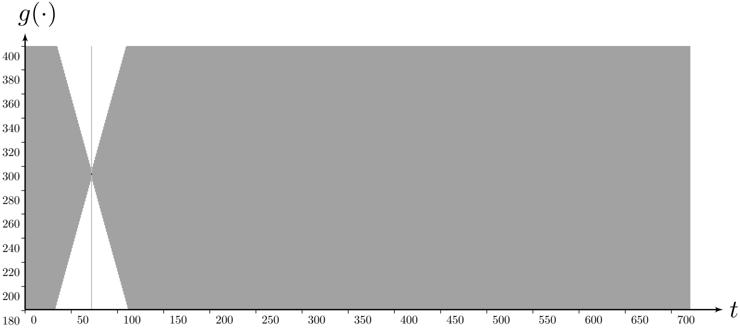


Non-linear state estimation:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) \\ y_i = g\big(\mathbf{x}(t_i)\big) \end{cases}$$

Exteroceptive measurements

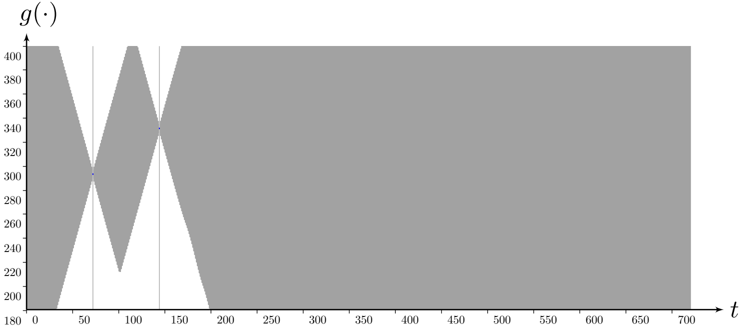
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



Observation tube, considering 1 range-only measurement from the beacon.

Exteroceptive measurements

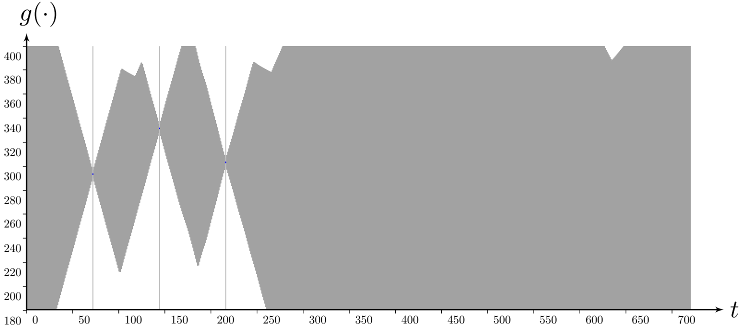
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



Observation tube, considering 2 range-only measurements from the beacon.

Exteroceptive measurements

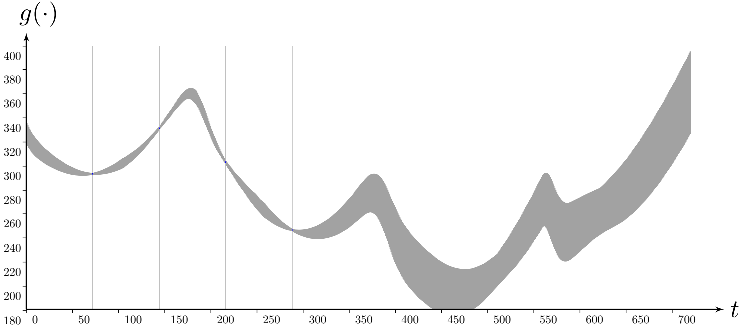
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



Observation tube, considering 3 range-only measurements from the beacon.

Exteroceptive measurements

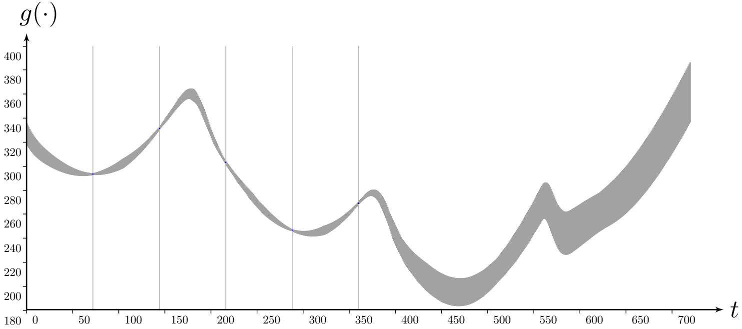
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



Observation tube, considering 4 range-only measurements from the beacon.

Exteroceptive measurements

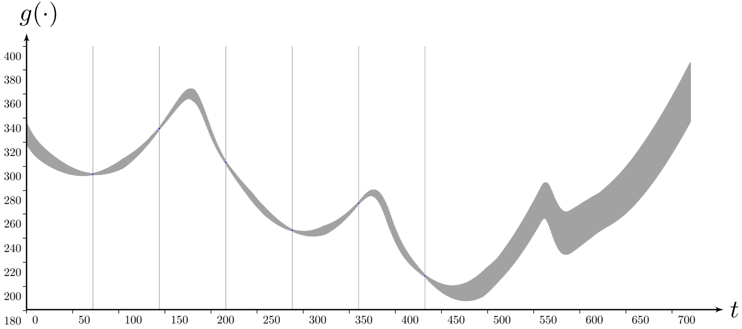
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



Observation tube, considering 5 range-only measurements from the beacon.

Exteroceptive measurements

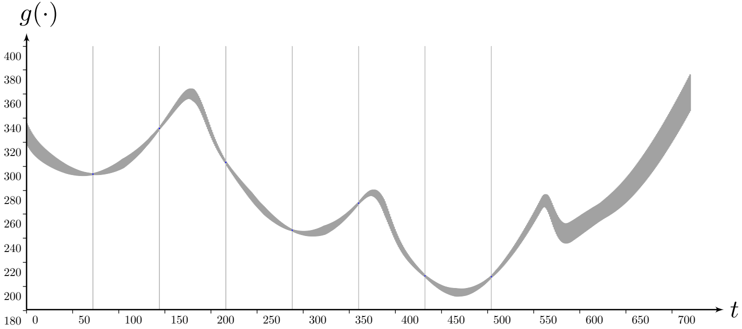
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



Observation tube, considering 6 range-only measurements from the beacon.

Exteroceptive measurements

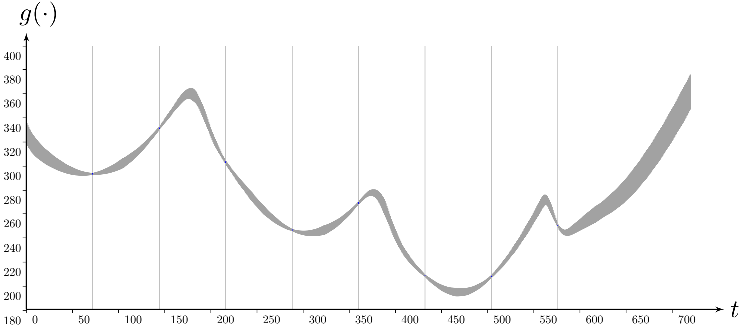
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



Observation tube, considering 7 range-only measurements from the beacon.

Exteroceptive measurements

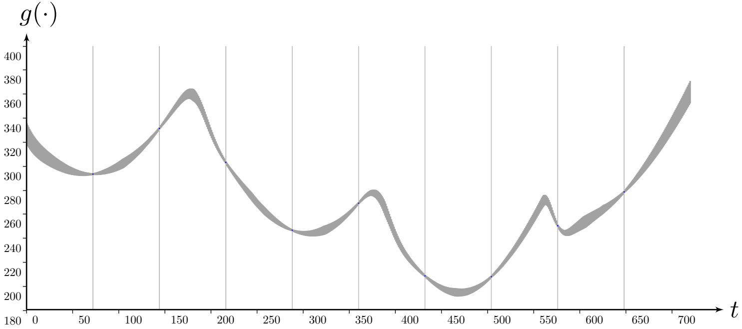
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



Observation tube, considering 8 range-only measurements from the beacon.

Exteroceptive measurements

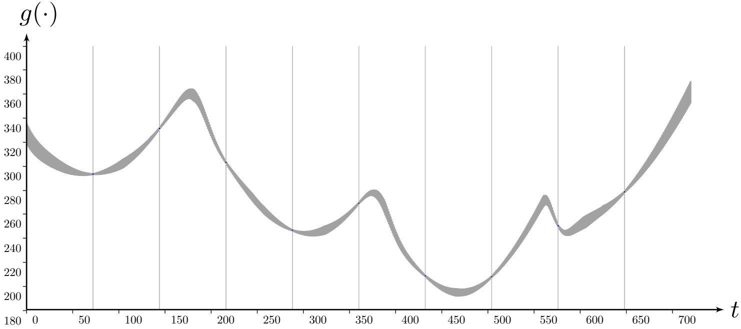
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



Observation tube, considering 9 range-only measurements from the beacon.

Exteroceptive measurements

Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



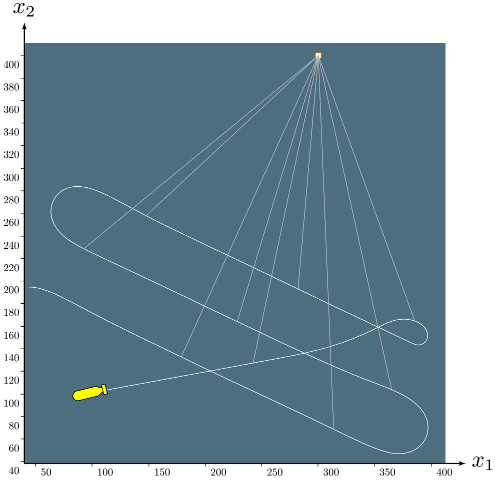
Observation tube, considering 9 range-only measurements from the beacon.

Then the state tube $[\mathbf{x}](\cdot)$ will be constrained by $[g](\cdot)$.

$$\mathcal{L}_g : \quad g(\cdot) = \sqrt{\big(x_1(\cdot) - \mathcal{B}_1\big)^2 + \big(x_2(\cdot) - \mathcal{B}_2\big)^2}.$$

Dynamic state estimation

Considering **range-only measurements** from a known beacon.

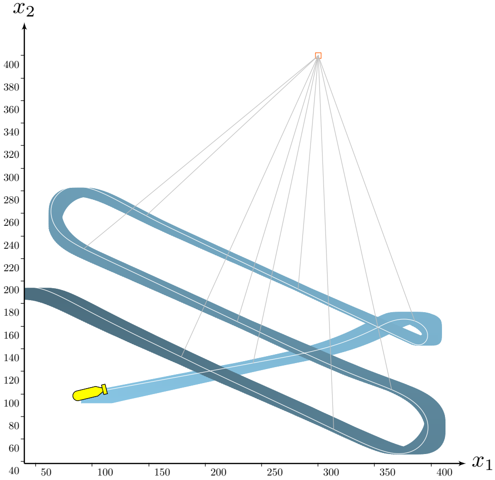


State estimation:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) \\ y_i = g\big(\mathbf{x}(t_i)\big) \end{cases}$$

Dynamic state estimation

Considering **range-only measurements** from a known beacon.



State estimation:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) \\ y_i = g\big(\mathbf{x}(t_i)\big) \end{cases}$$

Trajectory evaluation constraint

$$\text{Trajectory evaluation} \left\{ \begin{array}{l} \mathbf{z} = \mathbf{y}(t) \\ \\ \\ \end{array} \right.$$

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

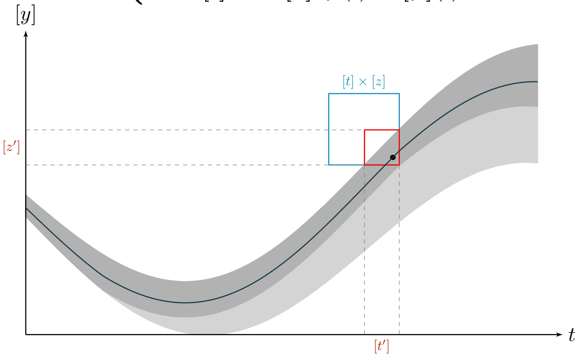
Trajectory evaluation constraint

$$\text{Trajectory evaluation} \left\{ \begin{array}{l} \mathbf{z} = \mathbf{y}(t) \\ \\ t \in [t], \mathbf{z} \in [\mathbf{z}], \mathbf{y}(\cdot) \in [\mathbf{y}](\cdot) \end{array} \right.$$

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

Trajectory evaluation constraint

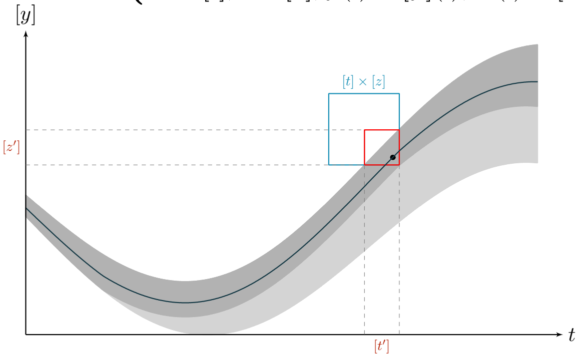
Trajectory evaluation $\begin{cases} \mathbf{z} = \mathbf{y}(t) \\ t \in [t], \mathbf{z} \in [\mathbf{z}], \mathbf{y}(\cdot) \in [\mathbf{y}](\cdot) \end{cases}$



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

Trajectory evaluation constraint

Trajectory evaluation $\begin{cases} \mathbf{z} = \mathbf{y}(t) \\ \dot{\mathbf{y}}(\cdot) = \mathbf{w}(\cdot) \\ t \in [t], \mathbf{z} \in [\mathbf{z}], \mathbf{y}(\cdot) \in [\mathbf{y}](\cdot), \mathbf{w}(\cdot) \in [\mathbf{w}](\cdot) \end{cases}$



Contractor programming: $\mathcal{C}_{\text{eval}}\big([t], [\mathbf{z}], [\mathbf{y}](\cdot), [\mathbf{w}](\cdot)\big)$

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

Assets of constraint programming

- **simplicity** of the approach
  transparent application of contractors on elementary constraints

Assets of constraint programming

- **simplicity** of the approach
  transparent application of contractors on elementary constraints
- **reliability** of the results: no solution can be lost
  useful for proof purposes and the safety of systems

Assets of constraint programming

- **simplicity** of the approach
  transparent application of contractors on elementary constraints
- **reliability** of the results: no solution can be lost
  useful for proof purposes and the safety of systems
- focus on **the *what* instead of the *how***
  no expertise required on how to solve a problem

Assets of constraint programming

- **simplicity** of the approach
  transparent application of contractors on elementary constraints
- **reliability** of the results: no solution can be lost
  useful for proof purposes and the safety of systems
- focus on **the *what* instead of the *how***
  no expertise required on how to solve a problem
- **complex systems** easily handled
  non-linearities, differential equations, values from datasets

Assets of constraint programming

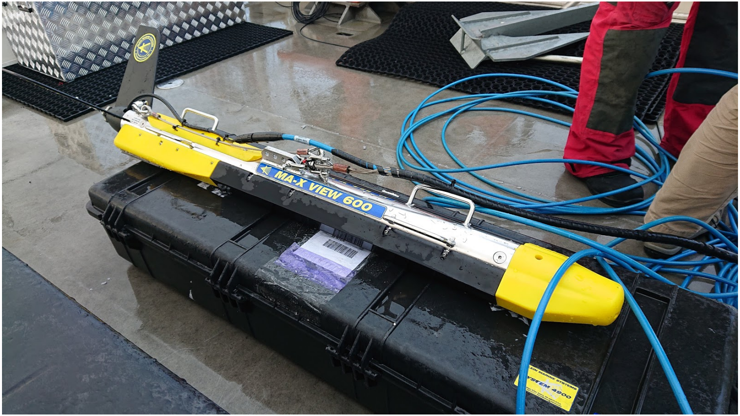
- **simplicity** of the approach
  transparent application of contractors on elementary constraints
- **reliability** of the results: no solution can be lost
  useful for proof purposes and the safety of systems
- focus on **the *what* instead of the *how***
  no expertise required on how to solve a problem
- **complex systems** easily handled
  non-linearities, differential equations, values from datasets

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

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

Towards more applications...

Example: underwater robotics with side-scan sonar:



Towards more applications...

Example: underwater robotics with side-scan sonar:

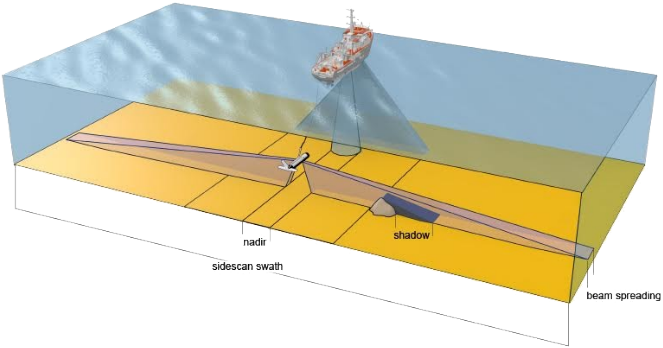
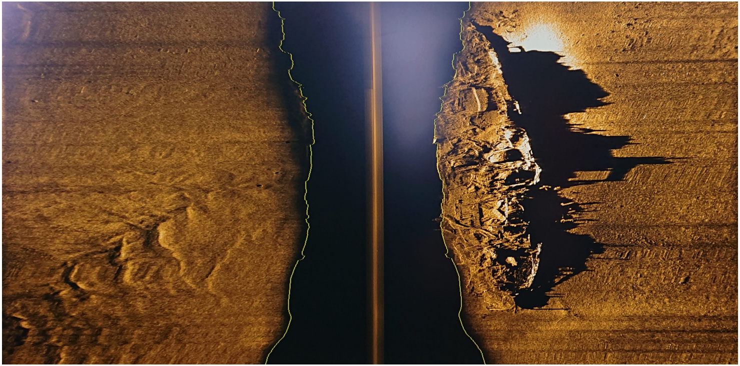


Image from www.ga.gov.au

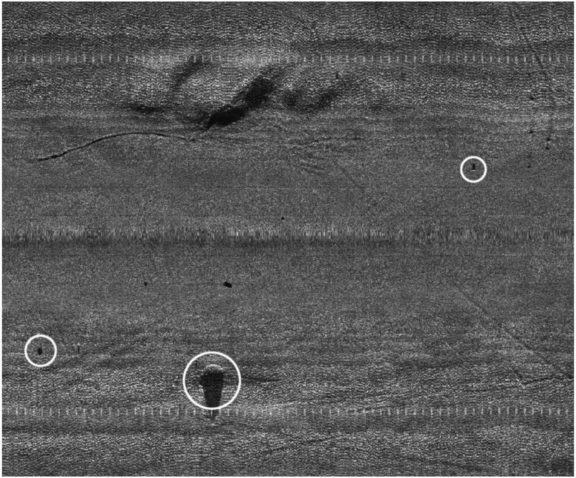
Towards more applications...

Example: underwater robotics with side-scan sonar:



Towards more applications...

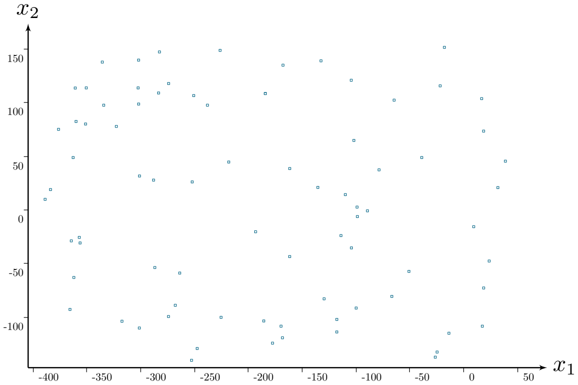
Now, onboard of an Autonomous Underwater Vehicle (AUV):



Detection of unidentifiable rocks on the seabed.

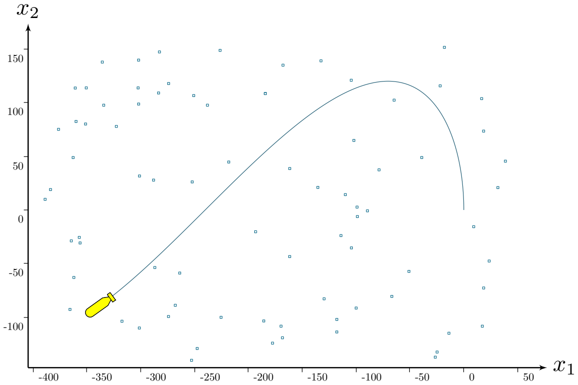
Localization with data association

$$\left\{ \begin{array}{ll} \mathbf{m}(t_i) \in \mathbb{M} & \text{(mapped landmark constraint)} \\ & \\ & \end{array} \right.$$



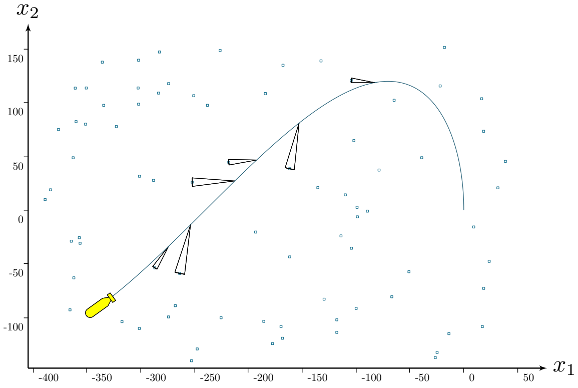
Localization with data association

$$\begin{cases} \mathbf{m}(t_i) \in \mathbb{M} & \text{(mapped landmark constraint)} \\ \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) & \text{(evolution equation)} \\ & \end{cases}$$



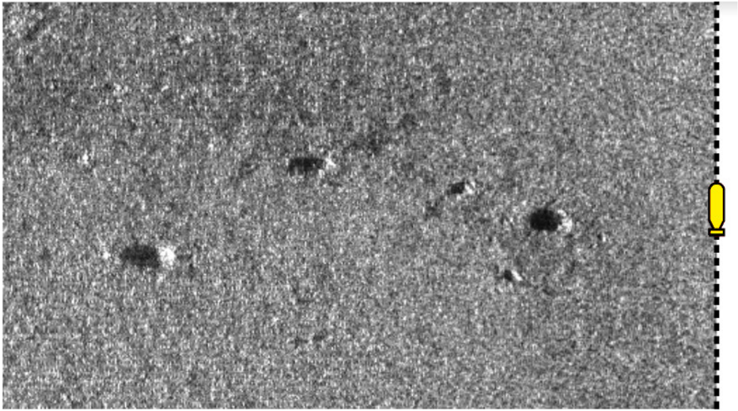
Localization with data association

$$\left\{\begin{array}{ll}\mathbf{m}(t_i)\in\mathbb{M} & \text{(mapped landmark constraint)}\\ \dot{\mathbf{x}}(t)=\mathbf{f}\big(\mathbf{x}(t),\mathbf{u}(t)\big) & \text{(evolution equation)}\\ \mathbf{g}\big(\mathbf{x}(t_i),\mathbf{y}(t_i),\mathbf{m}(t_i)\big)=\mathbf{0} & \text{(observation equation)}\end{array}\right.$$



Localization with data association

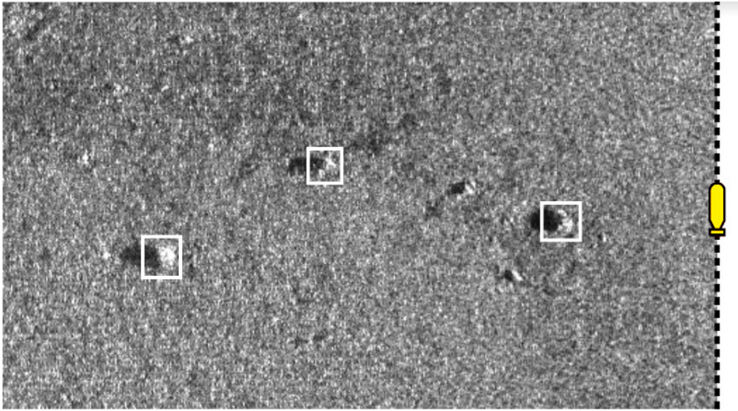
Constraint $\mathbf{m}(t_i) \in \mathbb{M}$:
An observation $\mathbf{m}(t_i)$ is related to one of the known seamarks $\mathbb{M}$.



Perception of the seabed with a side-scan sonar.

Localization with data association

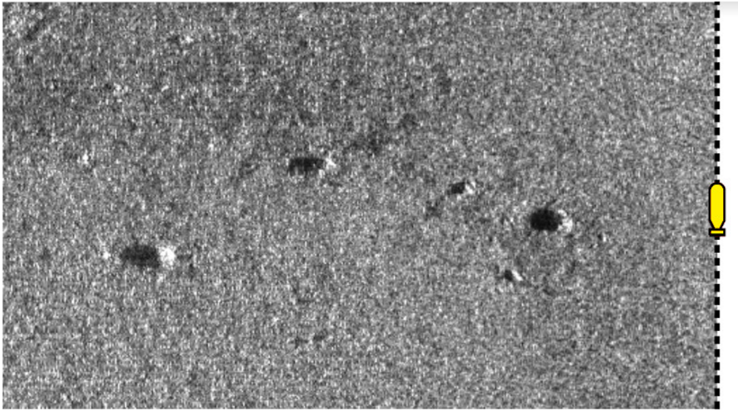
Constraint $\mathbf{m}(t_i) \in \mathbb{M}$:
An observation $\mathbf{m}(t_i)$ is related to one of the known seamarks $\mathbb{M}$.



Seamarks are already known with some uncertainty.

Localization with data association

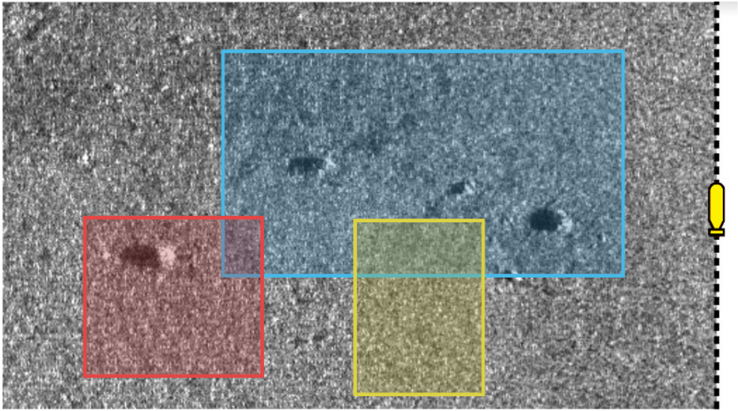
Constraint $\mathbf{m}(t_i) \in \mathbb{M}$:
An observation $\mathbf{m}(t_i)$ is related to one of the known seamarks $\mathbb{M}$.



Some of the rocks may be observed by the robot with its sonar.

Localization with data association

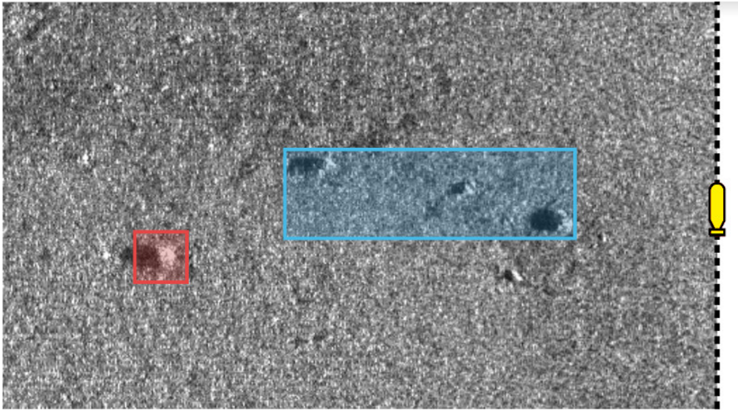
Constraint $\mathbf{m}(t_i) \in \mathbb{M}$:
An observation $\mathbf{m}(t_i)$ is related to one of the known seamarks $\mathbb{M}$.



The position of the rock is first estimated from robot's position estimate.

Localization with data association

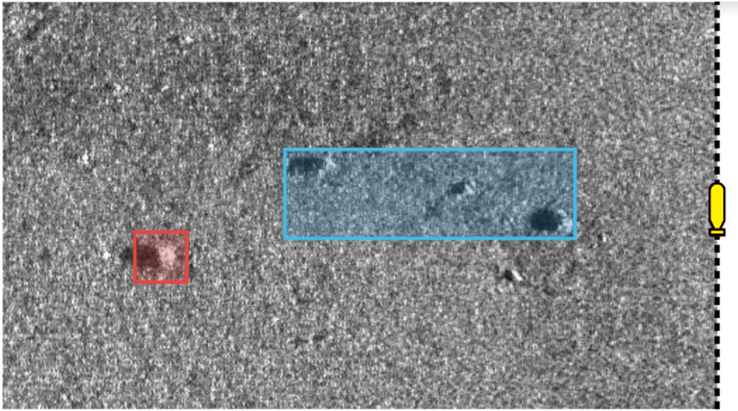
Constraint $\mathbf{m}(t_i) \in \mathbb{M}$:
An observation $\mathbf{m}(t_i)$ is related to one of the known seamarks $\mathbb{M}$.



Then the position of the rock is contracted from the known map.

Localization with data association

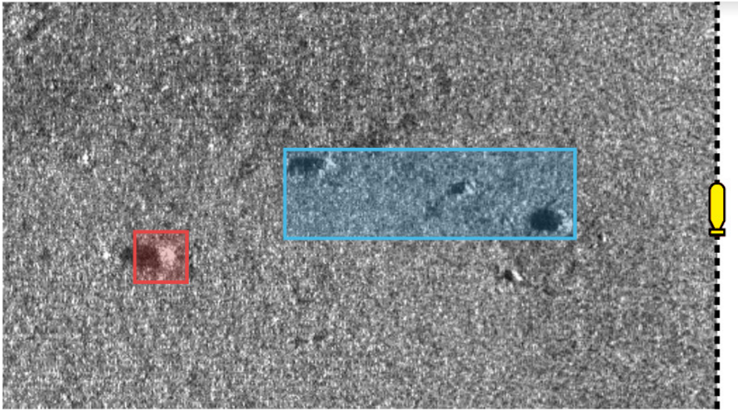
Constraint $\mathbf{m}(t_i) \in \mathbb{M}$:
An observation $\mathbf{m}(t_i)$ is related to one of the known seamarks $\mathbb{M}$.



If the boxed-position is a singleton, then the rock is *identified*.

Localization with data association

Constraint $\mathbf{m}(t_i) \in \mathbb{M}$:
An observation $\mathbf{m}(t_i)$ is related to one of the known seamarks $\mathbb{M}$.



In any cases, the boxed-positions of the rocks allow localization updates.

Localization with data association

Video