Tubex tutorial – Lesson 1

Simon Rohou

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

Tubex tutorial
11th March 2020

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

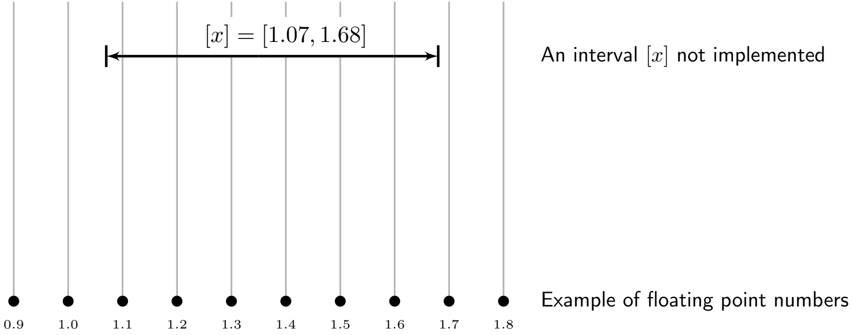
How to reliably represent floating point numbers?

$$0.1$$

Reliability

How to reliably represent floating point numbers?

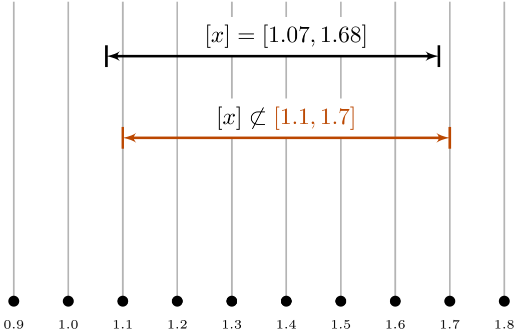
$$[1.07, 1.68]$$



Reliability

How to reliably represent floating point numbers?

$$[1.07, 1.68]$$



An interval $[x]$ not implemented

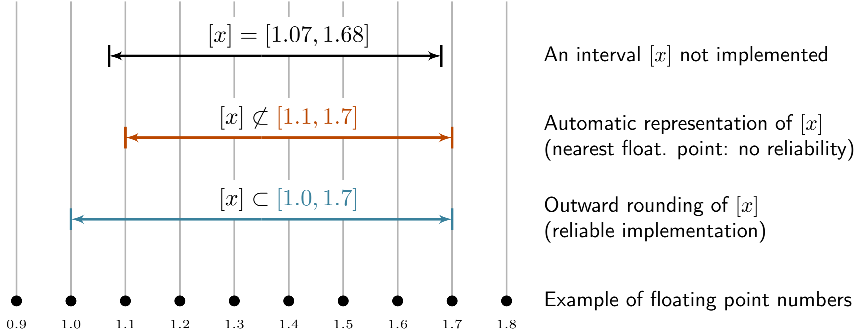
Automatic representation of $[x]$ (nearest float. point: no reliability)

Example of floating point numbers

Reliability

How to reliably represent floating point numbers?

$$[1.07, 1.68]$$



Reliability

What is the benefit for robotics?

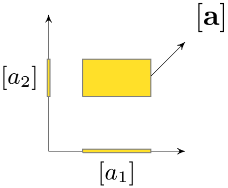
Interval Analysis

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

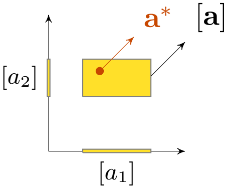
An interval $[x]$:

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

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



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

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):

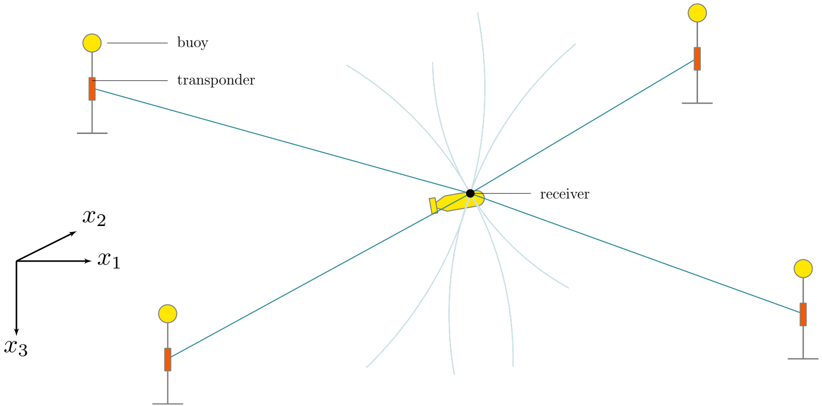
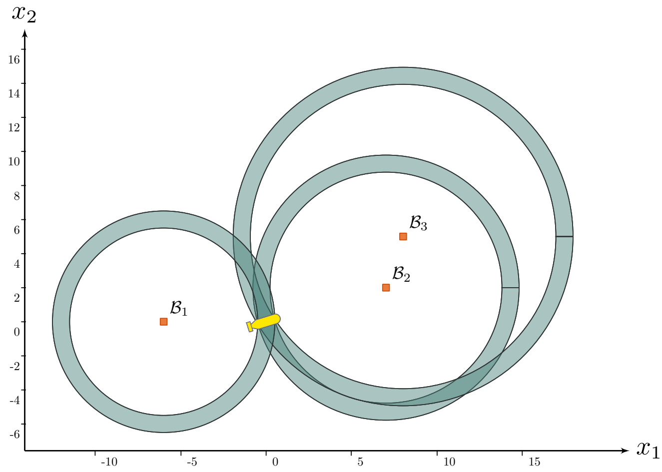


Illustration of Long BaseLine (LBL) positioning

Uncertainties as sets

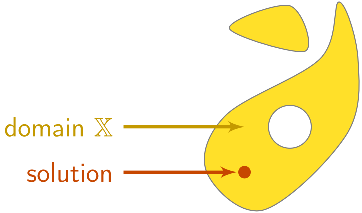
Example of **range-only** robot localization (three beacons):



LBL positioning with bounded uncertainties

Constraint programming: overall concept

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



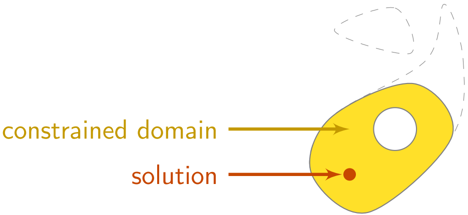
Constraint network:

- **Variables:** $\mathbf{x}$
- **Constraints:**
- **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:

Variables: $\mathbf{x}$

Constraints:

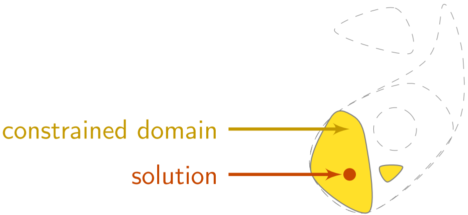
1. $\mathcal{L}_1(\mathbf{x})$

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, ...



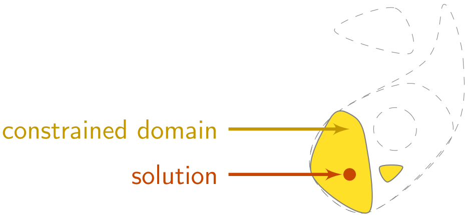
Constraint network:

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

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:

Variables: $\mathbf{x}$

Constraints:

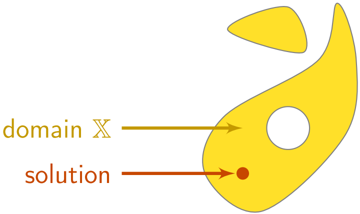
1. $\mathcal{L}_1(\mathbf{x})$
2. $\mathcal{L}_2(\mathbf{x})$
3. ...

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:

Variables: $\mathbf{x}$

Constraints:

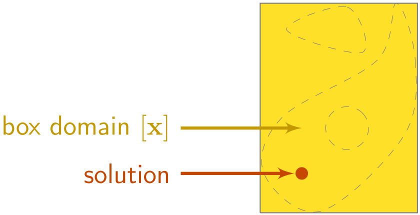
1. $\mathcal{L}_1(\mathbf{x})$
2. $\mathcal{L}_2(\mathbf{x})$
3. ...

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, ...
- representable domains: *e.g.* boxes $[\mathbf{x}]$



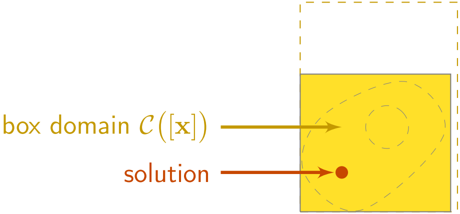
Constraint network:

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

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:

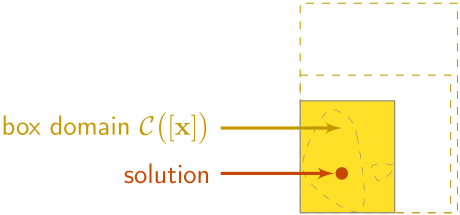
1. $\mathcal{L}_1(\mathbf{x})$
2. $\mathcal{L}_2(\mathbf{x})$
3. ...

Domains: $[\mathbf{x}]$

Contractor Programming
Chabert, Jaulin *Artifical Intelligence*, 2009

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:

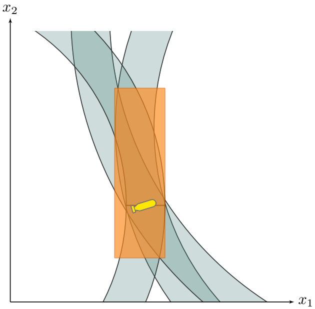
1. $\mathcal{L}_1(\mathbf{x})$
2. $\mathcal{L}_2(\mathbf{x})$
3. ...

Domains: $[\mathbf{x}]$

Contractor Programming
Chabert, Jaulin *Artifical Intelligence*, 2009

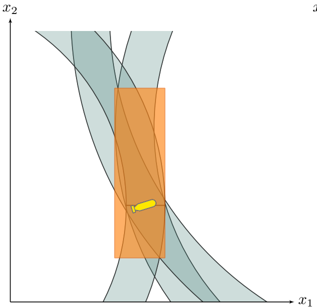
Wrappers

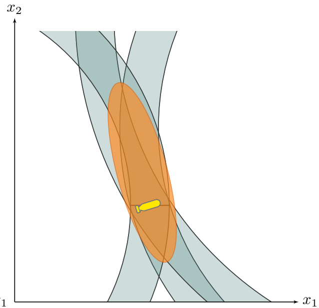
- box



Wrappers

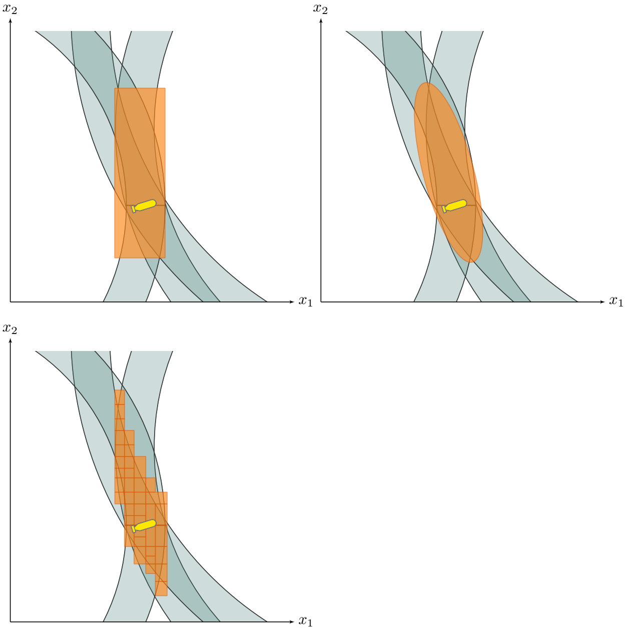
- box
- ellipse





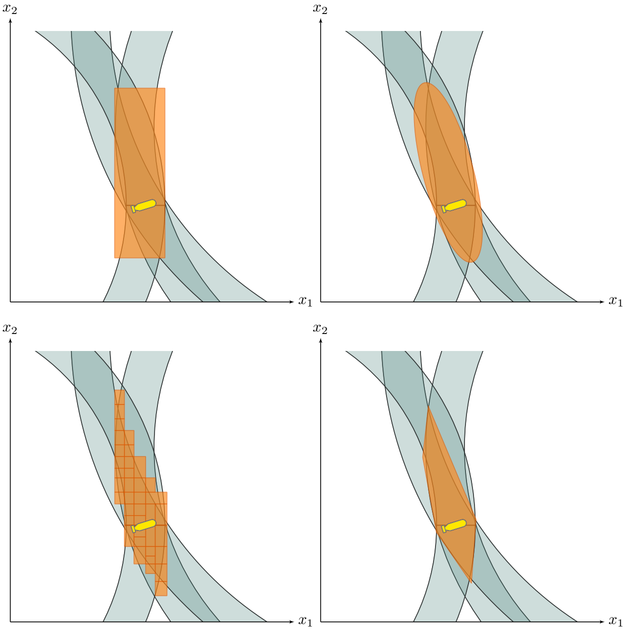
Wrappers

- box
- ellipse
- paving



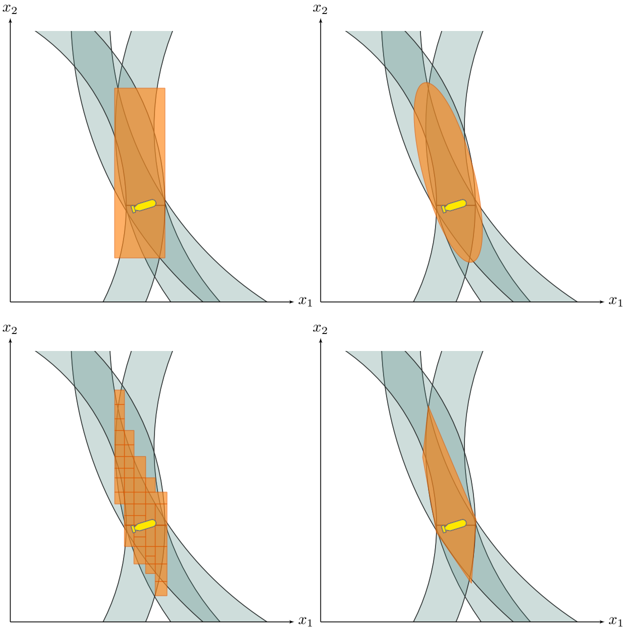
Wrappers

- box
- ellipse
- paving
- polygon



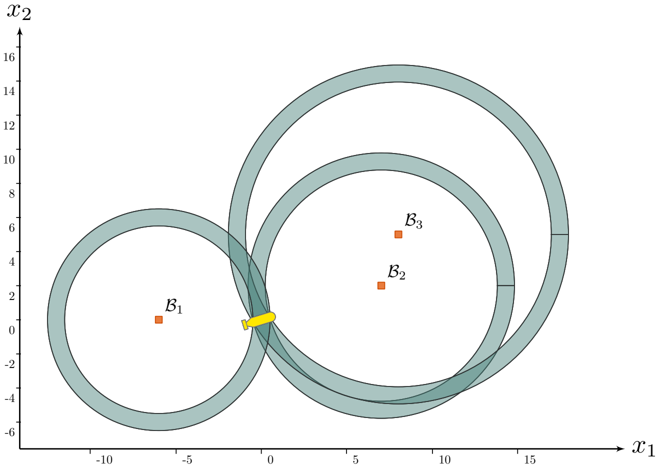
Wrappers

- 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**:

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

Constraints:

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

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

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:

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

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

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

Contractors to apply constraints

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

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

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

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

Contractors to apply constraints

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

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

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

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

Contractor programming

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

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

Constraints:

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

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

Contractor programming

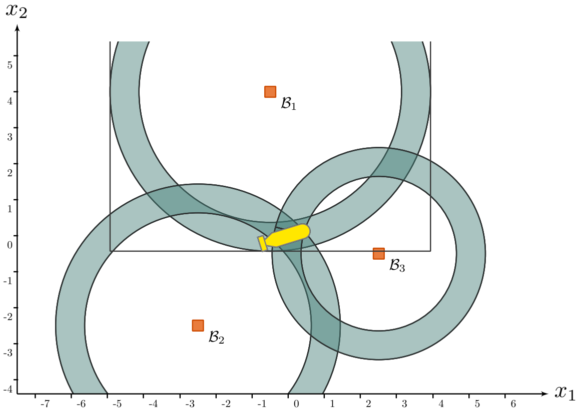
Now: problem to be solved with a **set of contractors**:

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

Initializations:

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

Fixed point propagations

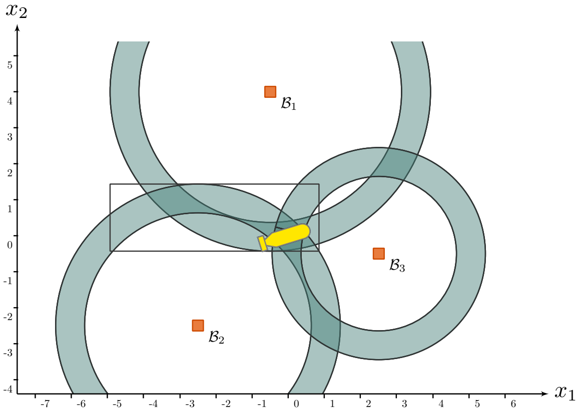


Applying $\mathcal{C}_g^{(1)}$.

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

Vincent Drevelle *Thesis*, 2011

Fixed point propagations

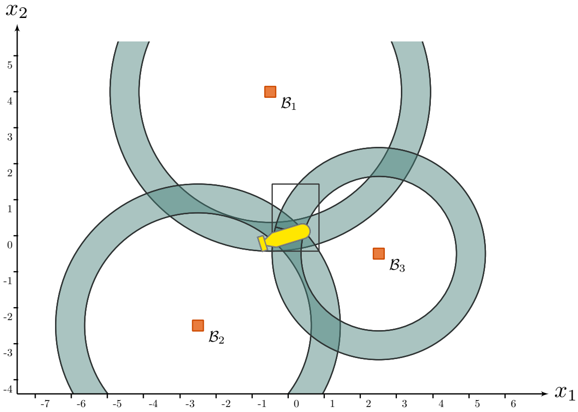


Applying $\mathcal{C}_g^{(2)}$.

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

Vincent Drevelle *Thesis*, 2011

Fixed point propagations

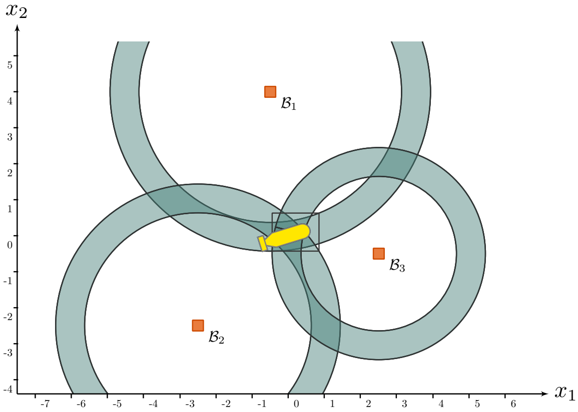


Applying $\mathcal{C}_g^{(3)}$.

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

Vincent Drevelle *Thesis*, 2011

Fixed point propagations

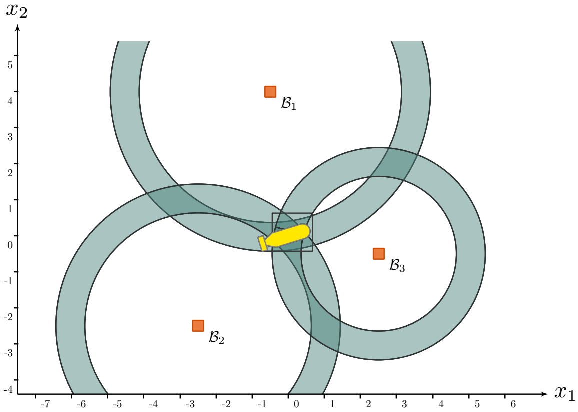


Applying $\mathcal{C}_g^{(1)}$.

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

Vincent Drevelle *Thesis*, 2011

Fixed point propagations

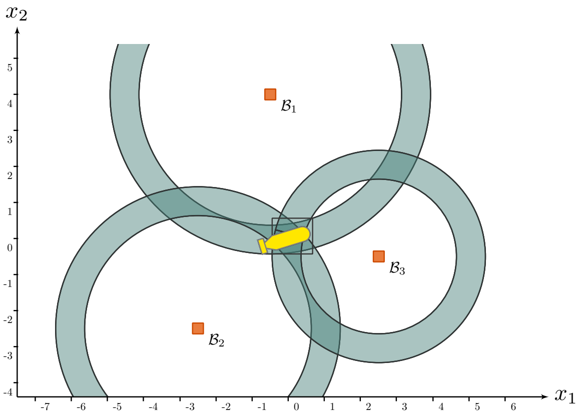


Applying $\mathcal{C}_g^{(2)}$.

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

Vincent Drevelle *Thesis*, 2011

Fixed point propagations

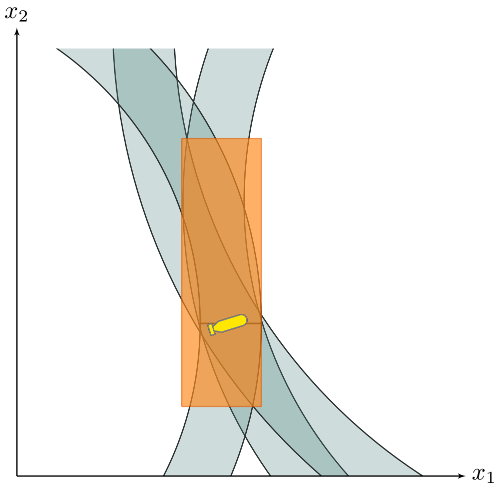


Fixed point reached.

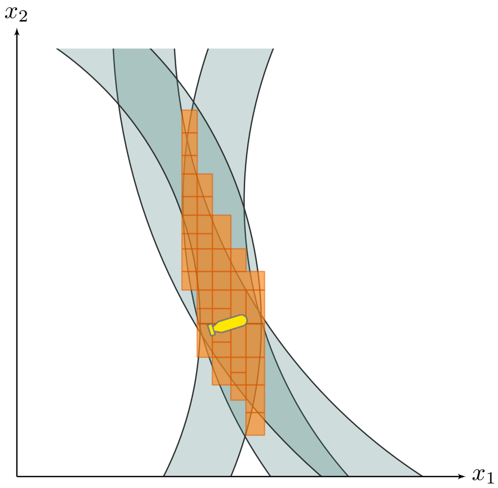
Study of robust set estimation methods for a high integrity multi-sensor localization.

Vincent Drevelle *Thesis*, 2011

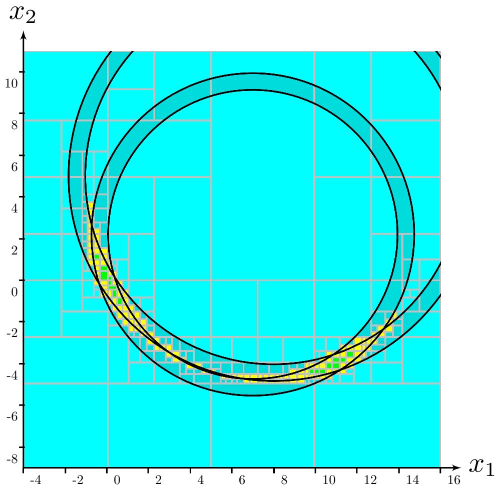
Sub-pavings



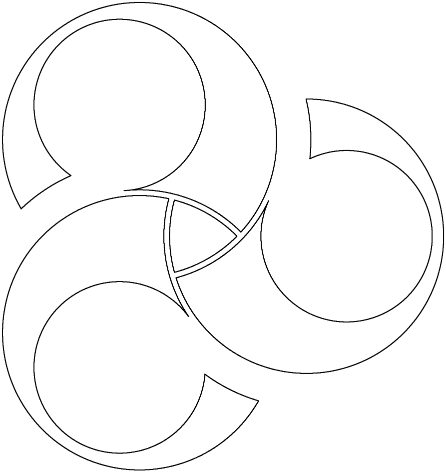
Sub-pavings



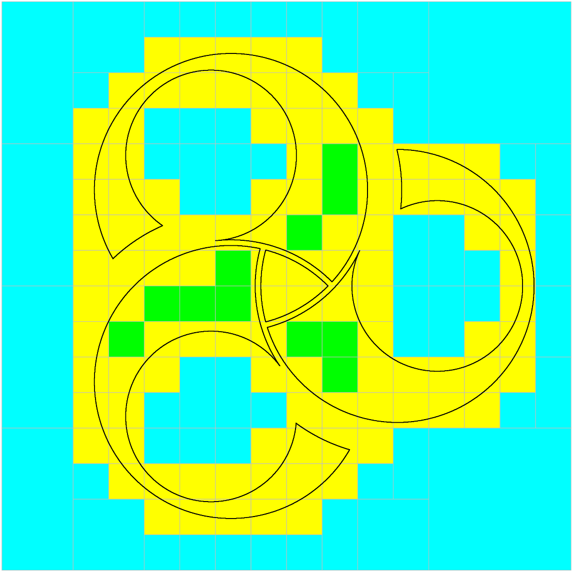
Sub-pavings



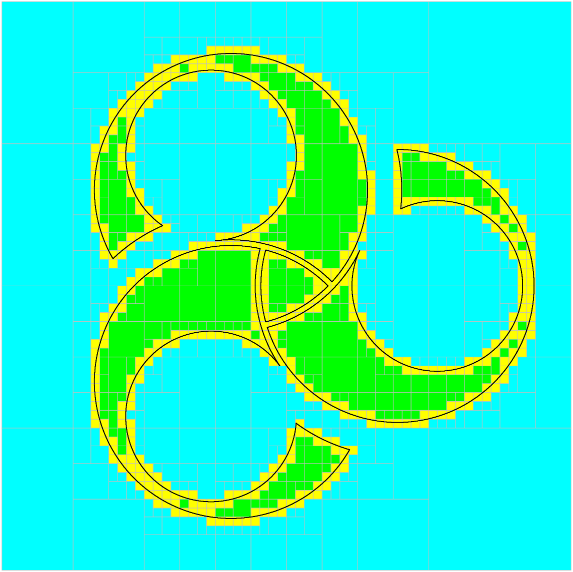
Sub-pavings: precision



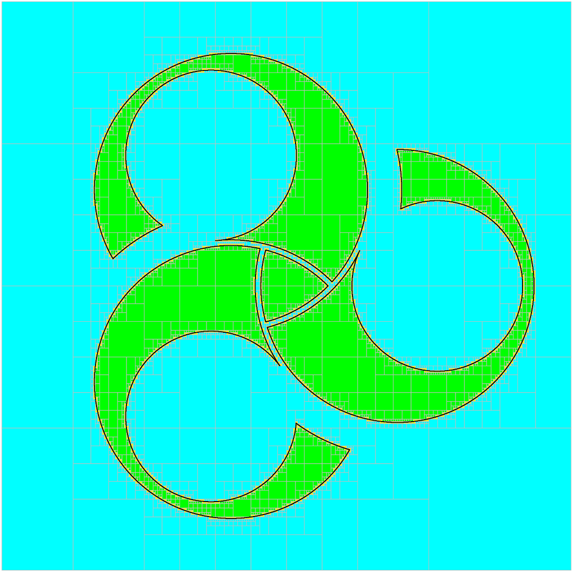
Sub-pavings: precision



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