Set-membership methods for mobile robotics

Simon Rohou

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

JNRR, Vittel
16$^{\text{th}}$ October 2019

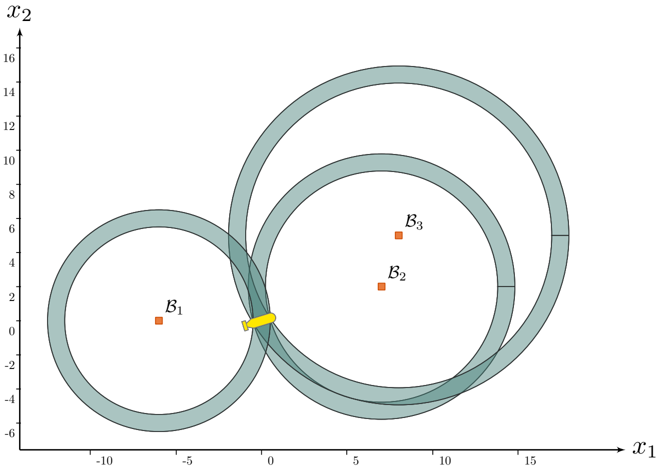
Mobile robotics

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

Special thanks to DGA-TN Brest (formerly GESMA)

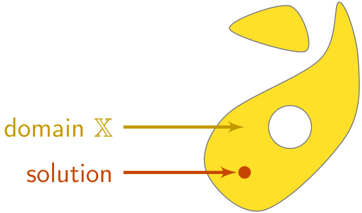
Uncertainties as sets

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



Constraint programming: overall concept

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



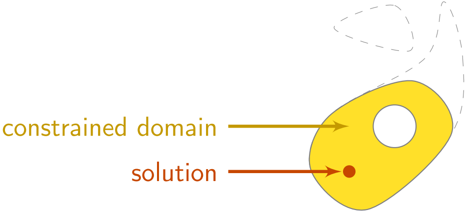
Constraint network:

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

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, ...



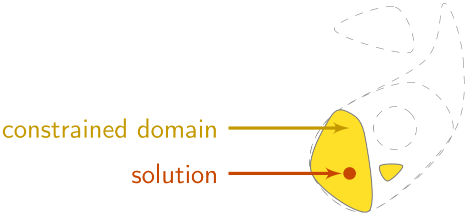
Constraint network:

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

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, ...



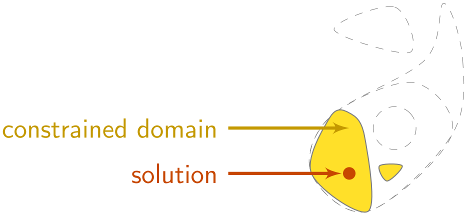
Constraint network:

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

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, ...



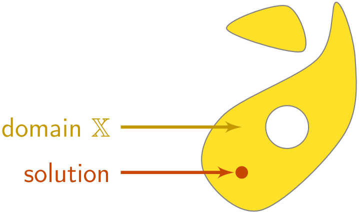
Constraint network:

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

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, ...



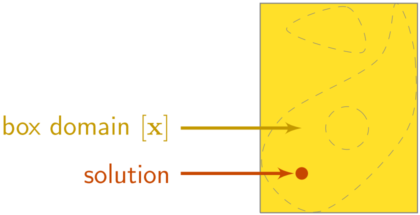
Constraint network:

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

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



Constraint network:

Variables: $\mathbf{x}$

Constraints:

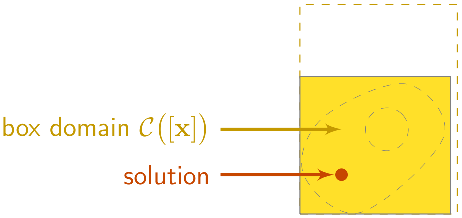
1. $\mathcal{L}_1(\mathbf{x})$
2. $\mathcal{L}_2(\mathbf{x})$
3. $\dots$

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:

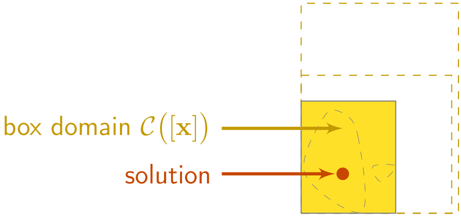
1. $\mathcal{L}_1(\mathbf{x})$
2. $\mathcal{L}_2(\mathbf{x})$
3. ...

Domains: $[\mathbf{x}]$

Contractor Programming
Chabert, Jaulin *Artifical Intelligence*, 2009

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**, $\mathcal{C}_{\mathcal{L}}([\mathbf{x}])$



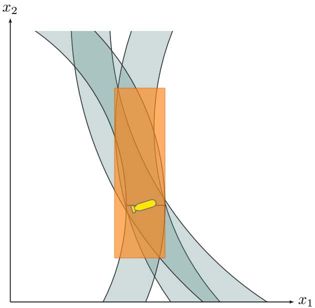
Constraint network:

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

Contractor Programming
Chabert, Jaulin *Artifical Intelligence*, 2009

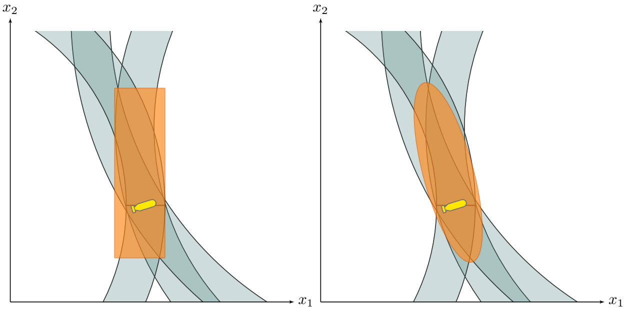
Wrappers

- box



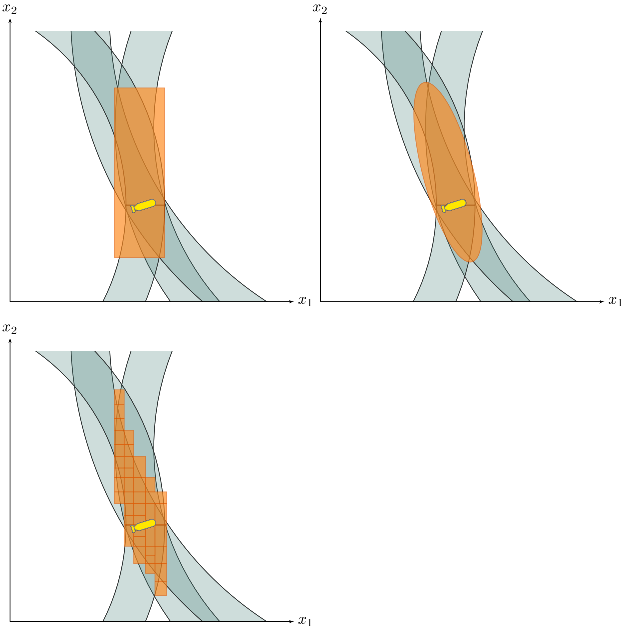
Wrappers

- box
- ellipse



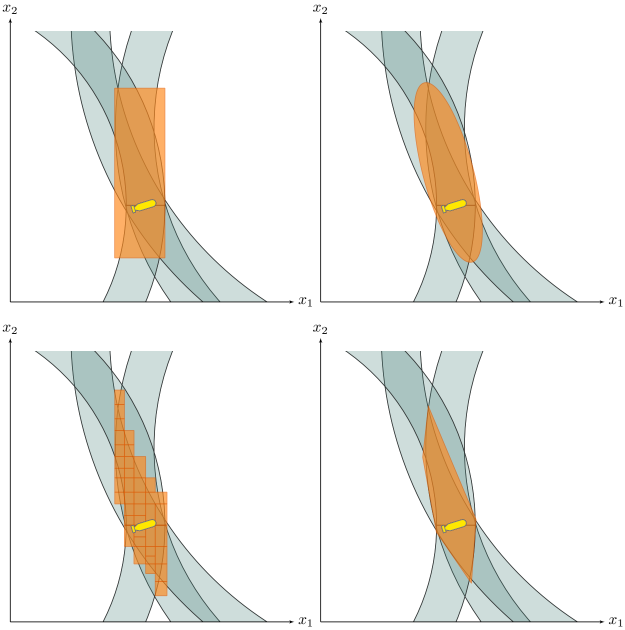
Wrappers

- box
- ellipse
- paving



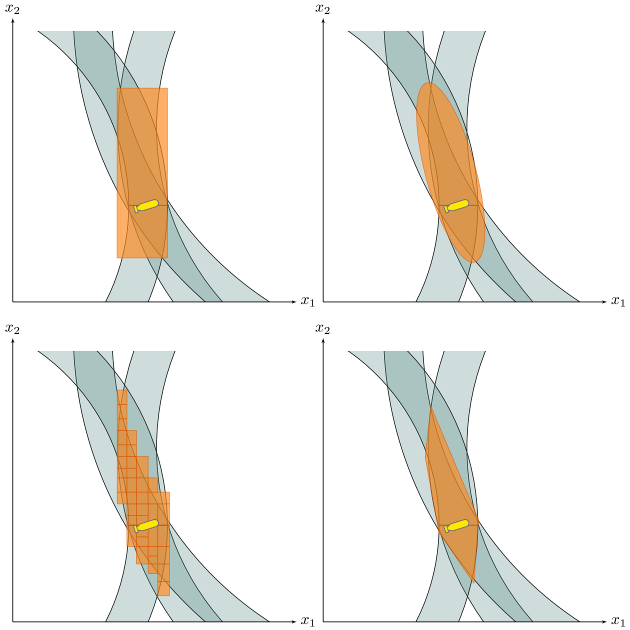
Wrappers

- box
- ellipse
- paving
- polygon



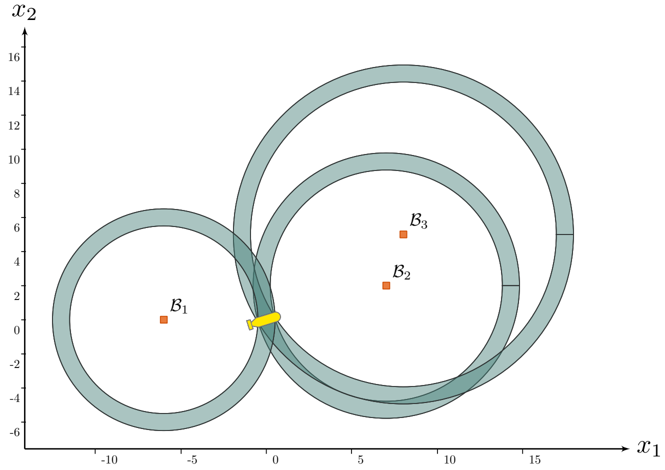
Wrappers

- 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

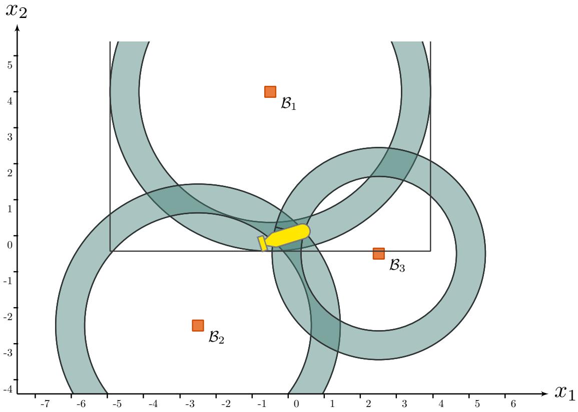
Observation constraint, links a measurement $\rho^{(k)}$ to the state $\mathbf{x}$:

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

Problem synthesized as a **constraint network**:

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

Fixed point propagations

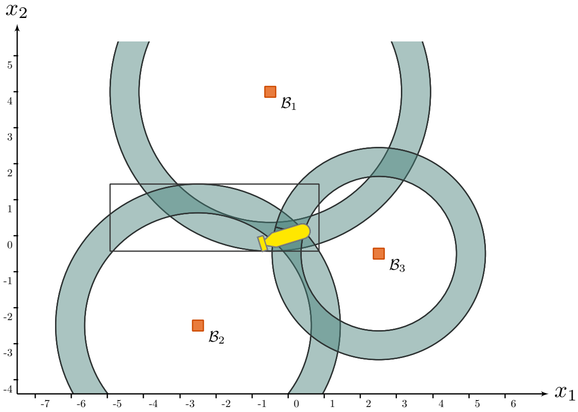


Applying $\mathcal{L}_g^{(1)}$

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

Vincent Drevelle *Thesis*, 2011

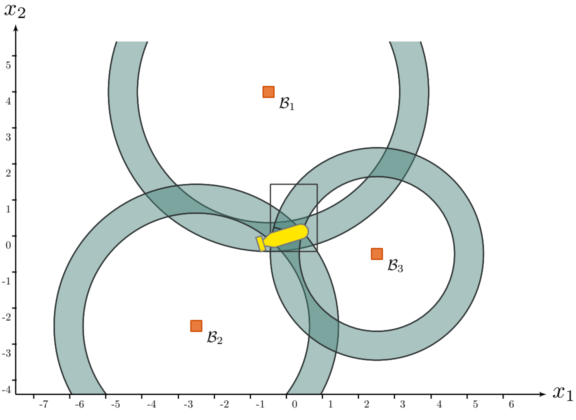
Fixed point propagations



Applying $\mathcal{L}_g^{(2)}$

Study of robust set estimation methods for a high integrity multi-sensor localization.
Vincent Drevelle *Thesis*, 2011

Fixed point propagations

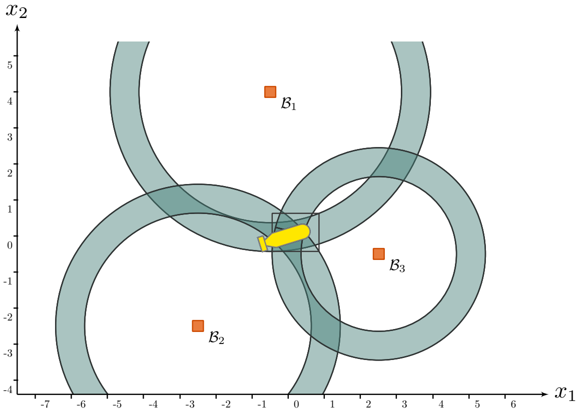


Applying $\mathcal{L}_g^{(3)}$

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

Vincent Drevelle *Thesis*, 2011

Fixed point propagations

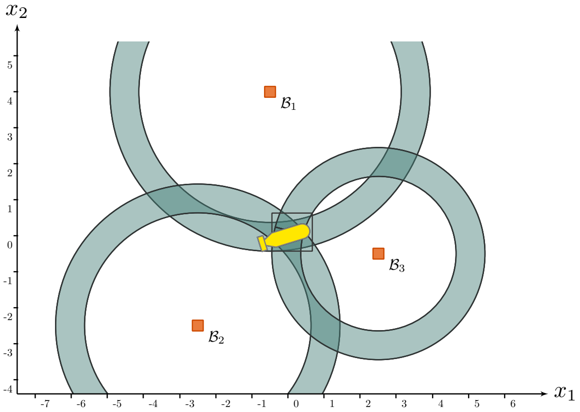


Applying $\mathcal{L}_g^{(1)}$

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

Vincent Drevelle *Thesis*, 2011

Fixed point propagations

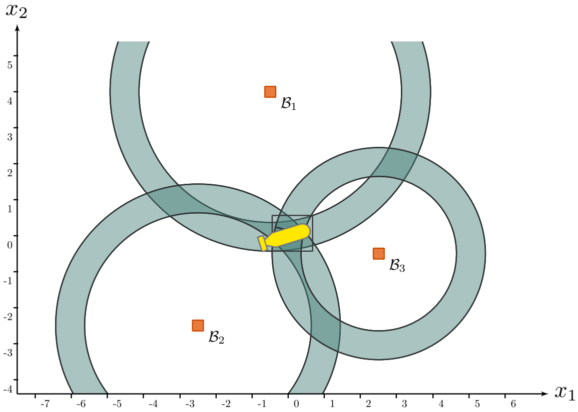


Applying $\mathcal{L}_g^{(2)}$

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

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

Further assets:

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

Sets from sensor data

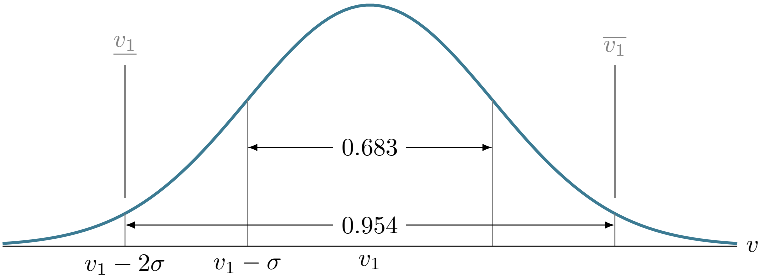
Sets from sensor data

Video

Sets from sensor data

Uncertainties:

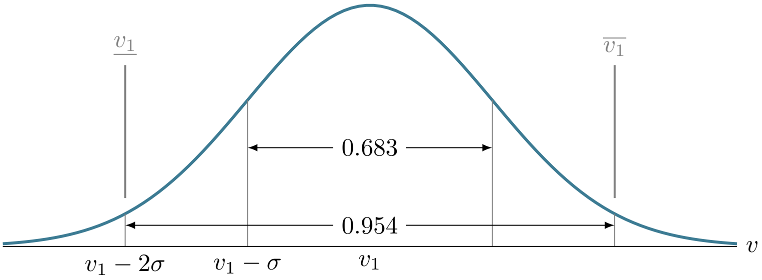
- datasheets $\Longrightarrow$ standard deviation σ for each sensor
- 95% confidence rate: $v_1^* \in [v_1] = [v_1 - 2\sigma, v_1 + 2\sigma]$



Sets from sensor data

Uncertainties:

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

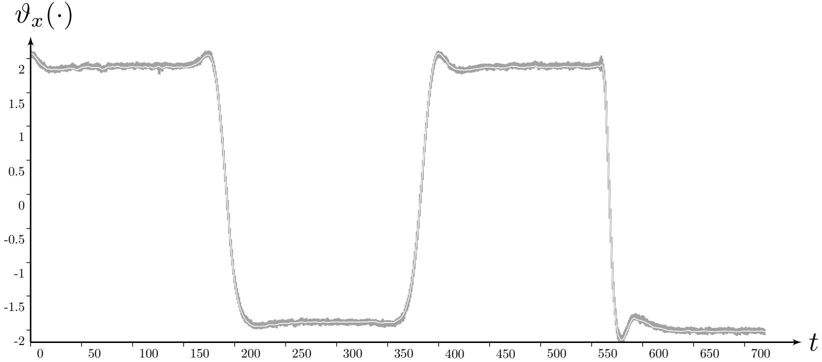


- uncertainties then reliably propagated in the system

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

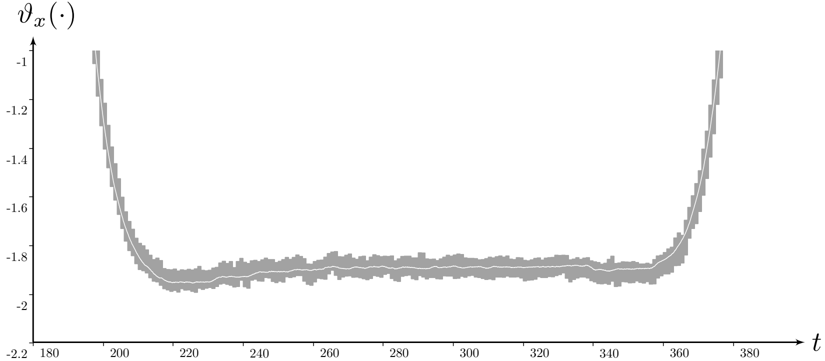
Example: velocity sensing

East velocity given by DVL + IMU:



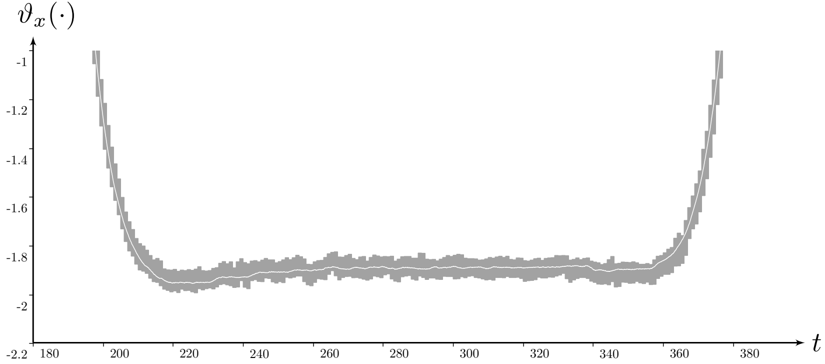
Example: velocity sensing

East velocity given by DVL + IMU (zoom):



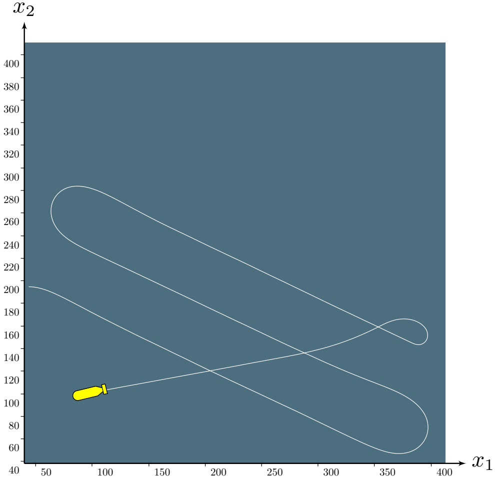
Example: velocity sensing

East velocity given by DVL + IMU (zoom):



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

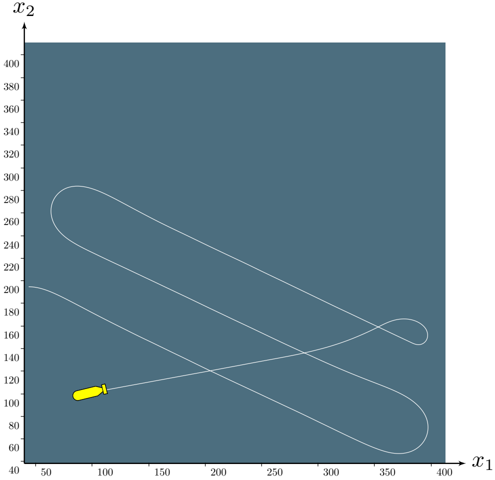
Dynamic state estimation



State estimation:

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

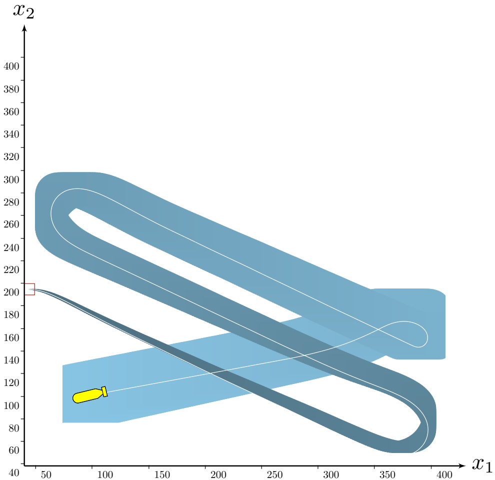
Dynamic state estimation



State estimation:

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

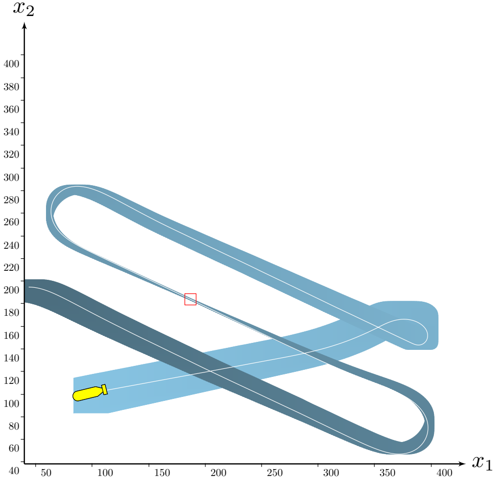
Dynamic state estimation



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



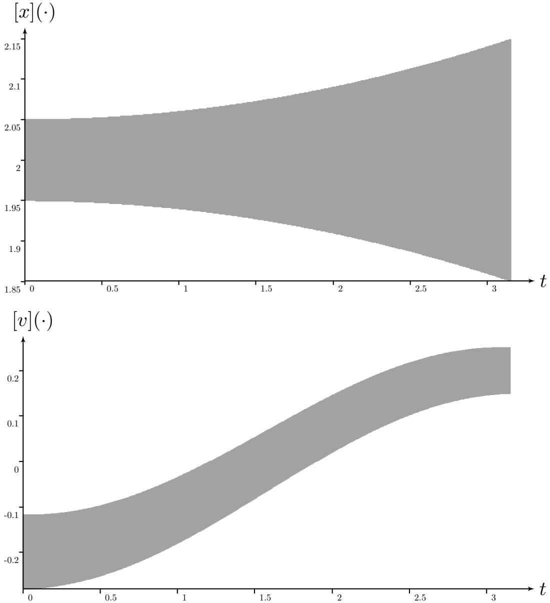
State estimation:

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

Derivative constraint

Differential constraint:

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



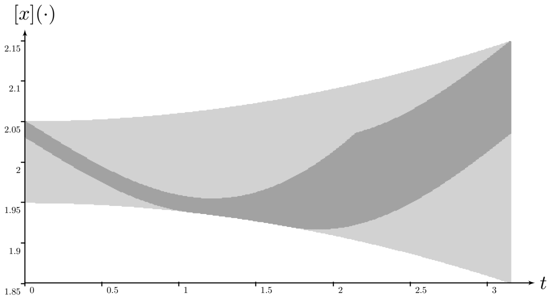
Derivative constraint

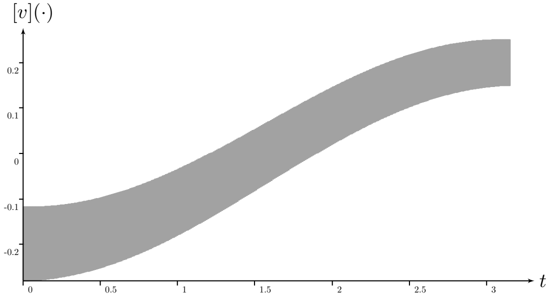
Differential constraint:

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

Contractor programming:

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





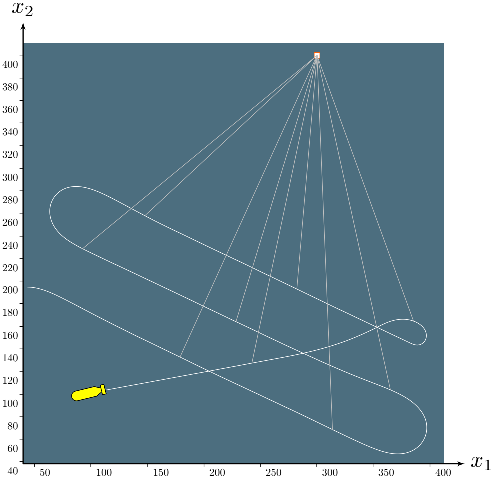
Guaranteed computation of robot trajectories

Rohou, Jaulin, Mihaylova, Le Bars, Veres

Robotics and Autonomous Systems, 2017

Dynamic state estimation

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

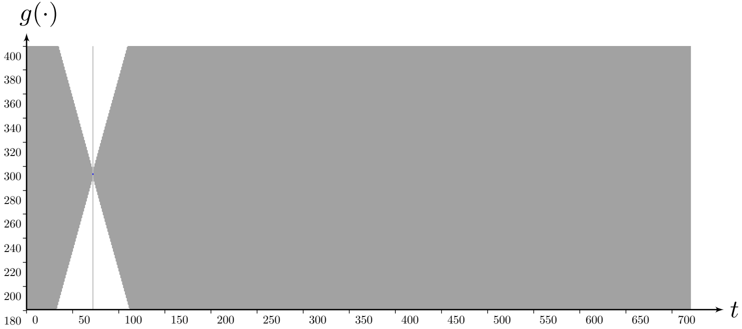


Non-linear state estimation:

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

Exteroceptive measurements

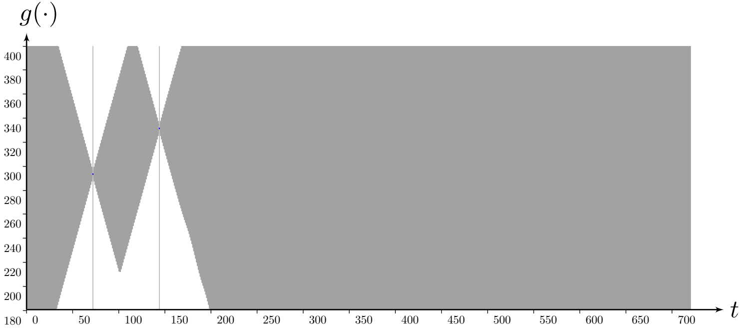
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



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

Exteroceptive measurements

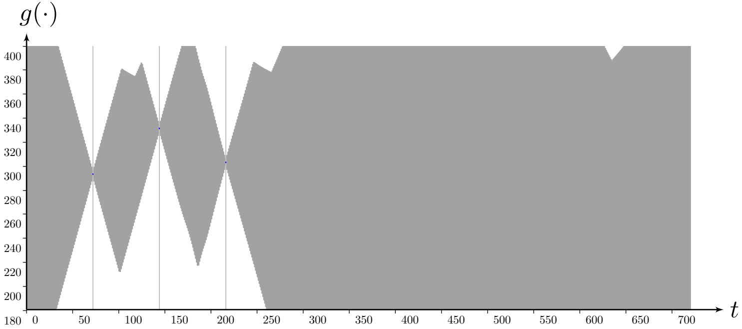
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



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

Exteroceptive measurements

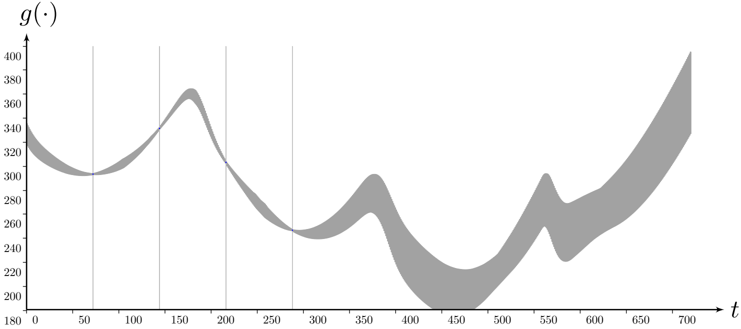
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



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

Exteroceptive measurements

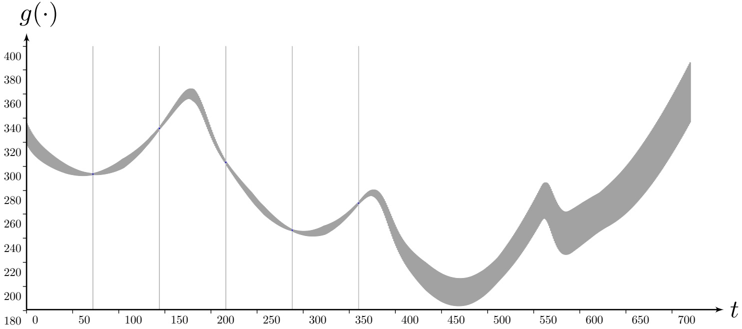
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



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

Exteroceptive measurements

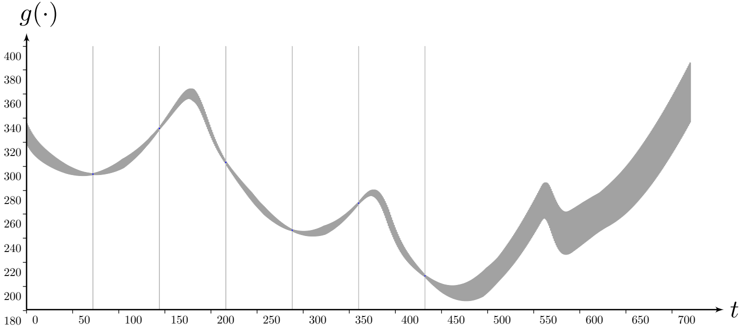
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



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

Exteroceptive measurements

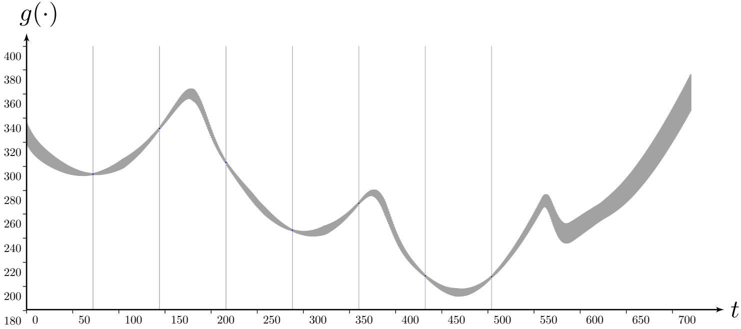
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



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

Exteroceptive measurements

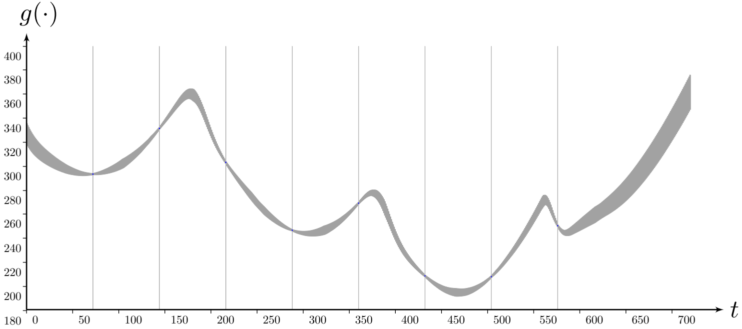
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



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

Exteroceptive measurements

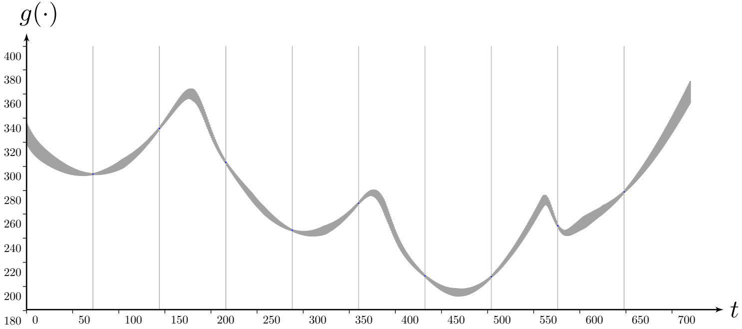
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



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

Exteroceptive measurements

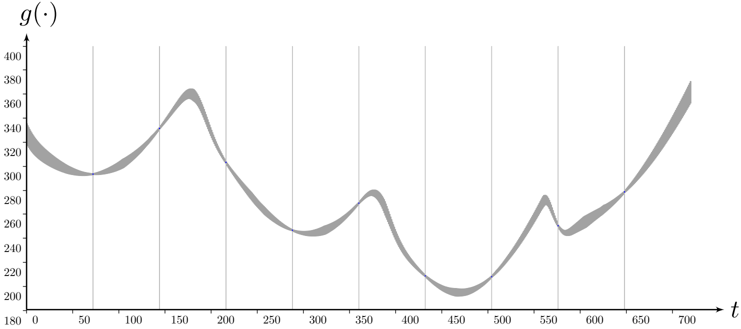
Creating another tube $[g](\cdot)$ that will be **constrained by measurements**.



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

Exteroceptive measurements

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



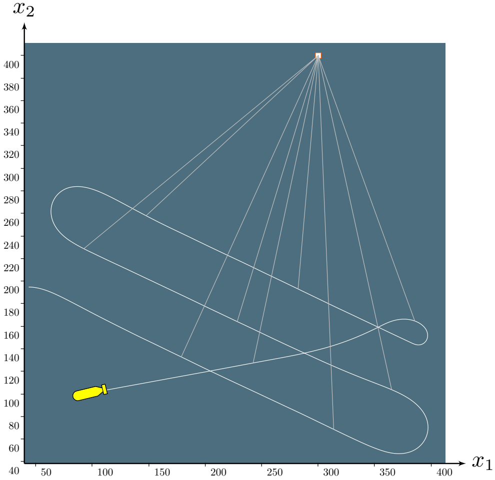
Observation tube, considering 9 range-only measurements from the beacon.

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

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

Dynamic state estimation

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

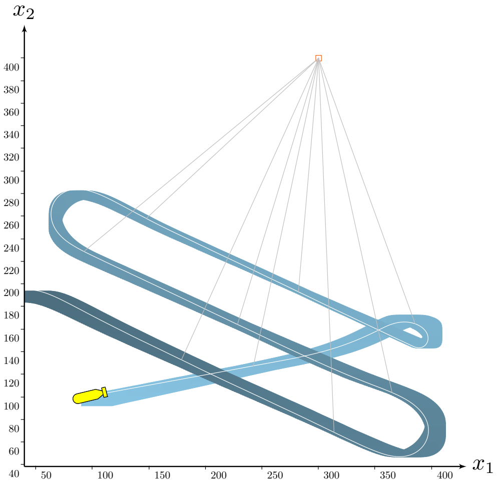


State estimation:

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

Dynamic state estimation

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



State estimation:

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

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...