Set-membership state estimation by solving data association

Simon Rohou, Benoît Desrochers, Luc Jaulin

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

Contredo, ENPC
22nd January 2020

ICRA 2020

ICRA Robotics conference: May 31 – June 4

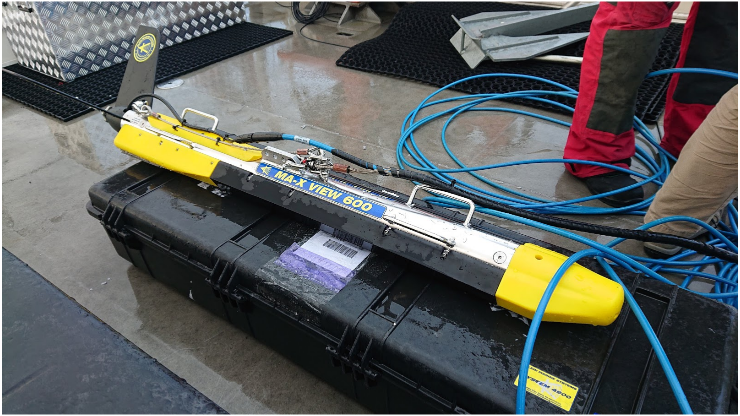
- **tutorial session of Tubex** (1 day)
- **paper presentation**

Program

DAY 1	DAY 2	DAY 3	DAY 4	DAY 5
SUNDAY, MAY 31	MONDAY, JUNE 1	TUESDAY, JUNE 2	WEDNESDAY, JUNE 3	THURSDAY, JUNE 4
Workshop & Tutorial sessions	Main Conferences Exhibition Competitions	Main Conferences Exhibition Competitions	Main Conferences Exhibition Competitions	Workshop & Tutorial sessions
Welcome Reception		Gala Evening	Farewell Reception	

Underwater robotics: sonar sensors

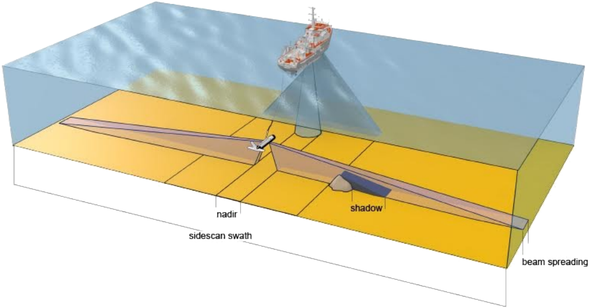
Side-scan sonars: to perceive objects on the seabed



A side-scan sonar Klein Max-View 600 during a demo in Brest.

Underwater robotics: sonar sensors

Side-scan sonars: to perceive objects on the seabed

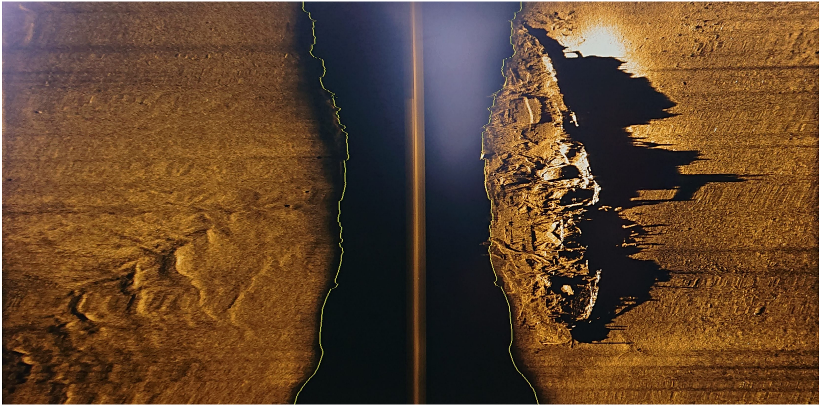


Schematic drawing illustrating the principles of a side-scan sonar.

Image from www.ga.gov.au

Underwater robotics: sonar sensors

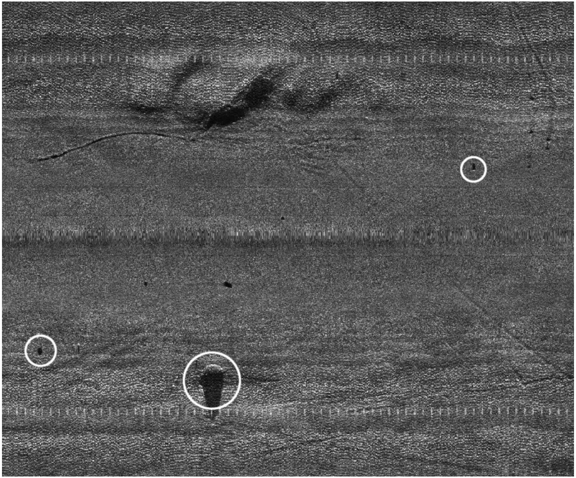
Side-scan sonars: to perceive objects on the seabed



Perception of a wreck with the Klein Max-View 600.

Underwater robotics: sonar sensors

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



Detection of unidentifiable/indistinguishable rocks on the seabed.

Localization with data association: assumptions

1. **the map is static and made of point landmarks**

Localization with data association: assumptions

1. **the map is static and made of point landmarks**
2. **the landmarks are indistinguishable**

Localization with data association: assumptions

1. **the map is static and made of point landmarks**

2. **the landmarks are indistinguishable**

3. **the position of each landmark is known (bounded)**

Localization with data association: assumptions

1. **the map is static and made of point landmarks**

2. **the landmarks are indistinguishable**

3. **the position of each landmark is known (bounded)**

4. **the initial pose of the robot is not known**

Localization with data association: assumptions

1. **the map is static and made of point landmarks**
   static: does not change over time, *points*: such as rocks

2. **the landmarks are indistinguishable**

3. **the position of each landmark is known (bounded)**

4. **the initial pose of the robot is not known**

Localization with data association: assumptions

1. **the map is static and made of point landmarks**
   static: does not change over time, *points*: such as rocks

2. **the landmarks are indistinguishable**
   all the rocks on the seabed look alike

3. **the position of each landmark is known (bounded)**

4. **the initial pose of the robot is not known**

Localization with data association: assumptions

1. **the map is static and made of point landmarks**
   static: does not change over time, *points*: such as rocks

2. **the landmarks are indistinguishable**
   all the rocks on the seabed look alike

3. **the position of each landmark is known (bounded)**
   a previous mapping of the area has been done by a surface vehicle

4. **the initial pose of the robot is not known**

Localization with data association: assumptions

1. **the map is static and made of point landmarks**
   static: does not change over time, *points*: such as rocks

2. **the landmarks are indistinguishable**
   all the rocks on the seabed look alike

3. **the position of each landmark is known (bounded)**
   a previous mapping of the area has been done by a surface vehicle

4. **the initial pose of the robot is not known**
   no satellite receivers underwater and strong drift during the dive

Localization with data association: assumptions

1. **the map is static and made of point landmarks**
   static: does not change over time, *points*: such as rocks

2. **the landmarks are indistinguishable**
   all the rocks on the seabed look alike

3. **the position of each landmark is known (bounded)**
   a previous mapping of the area has been done by a surface vehicle

4. **the initial pose of the robot is not known**
   no satellite receivers underwater and strong drift during the dive

state estimation $\rightleftarrows$ data association

State estimation with landmark perception

Classical localization problem:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) & \text{(evolution equation)} \\ \mathbf{g}\big(\mathbf{x}(t_i)\big) \in [\mathbf{y}](t_i) & \text{(observation equation)} \end{cases}$$

with:

- $\mathbf{x}$, unknown state vector
- $\mathbf{y}$, an output measurement vector
  related to the perception of a landmark
- $\mathbf{u}$, an input measurement vector

It is also known that $\forall t, \mathbf{u}(t) \in [\mathbf{u}](t)$ and $\mathbf{y}(t) \in [\mathbf{y}](t)$.

State estimation with landmark perception

Classical localization problem:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) & \text{(evolution equation)} \\ \mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}(t_i)\big) = \mathbf{0}\ ,\ \mathbf{y}(t_i) \in [\mathbf{y}](t_i) & \text{(implicit form)} \end{cases}$$

with:

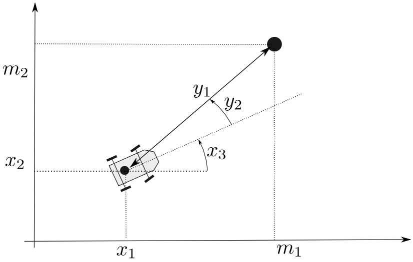
- $\mathbf{x}$, unknown state vector
- $\mathbf{y}$, an output measurement vector
  related to the perception of a landmark
- $\mathbf{u}$, an input measurement vector

It is also known that $\forall t, \mathbf{u}(t) \in [\mathbf{u}](t)$ and $\mathbf{y}(t) \in [\mathbf{y}](t)$.

State estimation with landmark perception: example

Example:

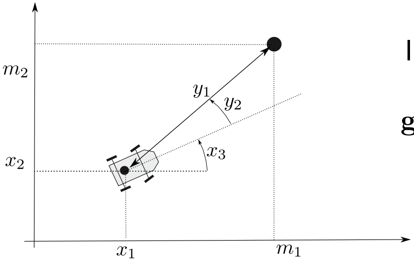
- a robot at position $(x_1, x_2)^\intercal$ with a heading x_3
- a landmark $\mathbf{m}$ located at $(4, 5)$
- the corresponding measurement vector is composed of
  - the distance y_1
  - the bearing y_2



State estimation with landmark perception: example

Example:

- a robot at position $(x_1, x_2)^\intercal$ with a heading x_3
- a landmark $\mathbf{m}$ located at $(4, 5)$
- the corresponding measurement vector is composed of
  - the distance y_1
  - the bearing y_2



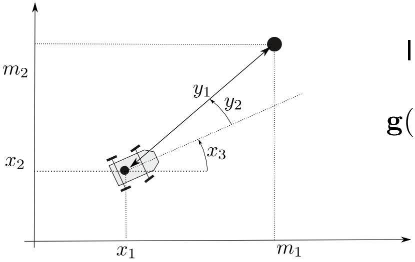
In such case, we have:

$$\mathbf{g}(\mathbf{x},\mathbf{y}) = \begin{pmatrix} x_1 + y_1 \cdot \cos(x_3 + y_2) - 4 \\ x_2 + y_1 \cdot \sin(x_3 + y_2) - 5 \end{pmatrix}$$

State estimation with landmark perception: example

Example:

- a robot at position $(x_1, x_2)^\intercal$ with a heading x_3
- a landmark $\mathbf{m}$ located at $(4, 5)$
- the corresponding measurement vector is composed of
  - the distance y_1
  - the bearing y_2



In such case, we have:

$$\mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{m}) = \begin{pmatrix} x_1 + y_1 \cdot \cos(x_3 + y_2) - m_1 \\ x_2 + y_1 \cdot \sin(x_3 + y_2) - m_2 \end{pmatrix}$$

State estimation with landmark perception

In the general case we have:

$$\left\{\begin{array}{l}\dot{\mathbf{x}}(t)=\mathbf{f}\big(\mathbf{x}(t),\mathbf{u}(t)\big)\\ \mathbf{g}\big(\mathbf{x}(t_i),\mathbf{y}(t_i),\mathbf{m}(t_i)\big)=\mathbf{0},\\ \qquad\qquad \mathbf{y}(t_i)\in[\mathbf{y}](t_i),\ \mathbf{m}(t_i)\in[\mathbf{m}](t_i)\end{array}\right.$$

with $[\mathbf{m}](t_i)$ the bounded position of the beacon perceived at time t_i.

State estimation with landmark perception

In the general case we have:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) \\ \mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}(t_i), \mathbf{m}(t_i)\big) = \mathbf{0}, \\ \qquad\qquad \mathbf{y}(t_i) \in [\mathbf{y}](t_i),\ \mathbf{m}(t_i) \in [\mathbf{m}](t_i) \end{cases}$$

with $[\mathbf{m}](t_i)$ the bounded position of the beacon perceived at time t_i.

Problem: when several landmarks $\mathbf{m}_1,\dots,\mathbf{m}_l$ can be observed,

- data may not be associated,
- we do not know to which landmark $\mathbf{m}(t_i)$ the measurement $\mathbf{y}(t_i)$ refers.

State estimation with landmark perception

In the general case we have:

$$\left\{\begin{array}{l}\dot{\mathbf{x}}(t)=\mathbf{f}\big(\mathbf{x}(t),\mathbf{u}(t)\big)\\ \mathbf{g}\big(\mathbf{x}(t_i),\mathbf{y}(t_i),\mathbf{m}(t_i)\big)=\mathbf{0},\\ \qquad\qquad \mathbf{y}(t_i)\in[\mathbf{y}](t_i),\ \mathbf{m}(t_i)\in[\mathbf{m}](t_i)\end{array}\right.$$

with $[\mathbf{m}](t_i)$ the bounded position of the beacon perceived at time t_i.

Problem: when several landmarks $\mathbf{m}_1,\dots,\mathbf{m}_l$ can be observed,

- data may not be associated,
- we do not know to which landmark $\mathbf{m}(t_i)$ the measurement $\mathbf{y}(t_i)$ refers.

The map constraint has now the form:

$$\big(\mathbf{m}(t_i)\in[\mathbf{m}_1]\big)\vee\dots\vee\big(\mathbf{m}(t_i)\in[\mathbf{m}_\ell]\big)$$

State estimation with landmark perception

In the general case we have:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) \\ \mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}(t_i), \mathbf{m}(t_i)\big) = \mathbf{0}, \\ \qquad\qquad \mathbf{y}(t_i) \in [\mathbf{y}](t_i),\ \mathbf{m}(t_i) \in [\mathbf{m}](t_i) \end{cases}$$

with $[\mathbf{m}](t_i)$ the bounded position of the beacon perceived at time t_i.

Problem: when several landmarks $\mathbf{m}_1$,...,$\mathbf{m}_l$ can be observed,

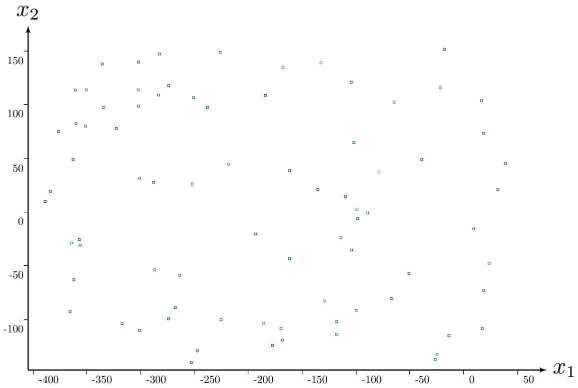
- data may not be associated,
- we do not know to which landmark $\mathbf{m}(t_i)$ the measurement $\mathbf{y}(t_i)$ refers.

The map constraint has now the form:

$$\big(\mathbf{m}(t_i) \in [\mathbf{m}_1]\big) \vee \cdots \vee \big(\mathbf{m}(t_i) \in [\mathbf{m}_\ell]\big) \Leftrightarrow \mathbf{m}(t_i) \in \mathbb{M} = \{[\mathbf{m}_1], \ldots, [\mathbf{m}_\ell]\}$$

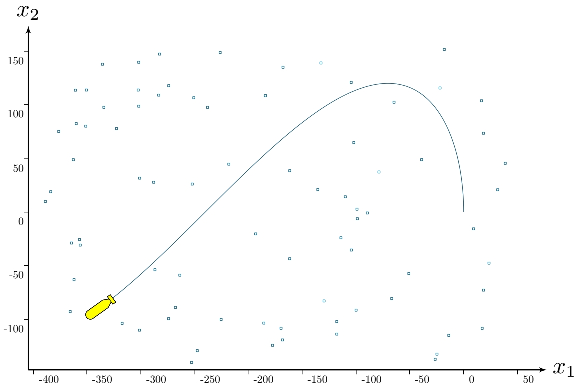
Localization with data association: formalization

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



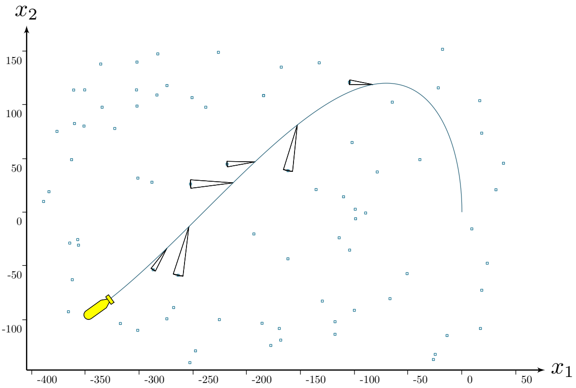
Localization with data association: formalization

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



Localization with data association: formalization

$$\begin{cases} \mathbf{m}(t_i) \in \mathbb{M} & \text{(association constraint)} \\ \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) & \text{(evolution equation)} \\ \mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}(t_i), \mathbf{m}(t_i)\big) = \mathbf{0} & \text{(observation equation)} \end{cases}$$



Localization with data association: formalization

Provides a test case with **heterogeneous constraints**:

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

Localization with data association: formalization

Provides a test case with **heterogeneous constraints**:

$$\left\{ \begin{array}{ll} \mathbf{m}(t_i) \in \mathbb{M} & \rightarrow \text{discrete constraint} \\ \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) & \rightarrow \text{differential constraint} \\ & \end{array} \right.$$

Localization with data association: formalization

Provides a test case with **heterogeneous constraints**:

$$\left\{\begin{array}{ll}\mathbf{m}(t_i) \in \mathbb{M} & \rightarrow \text{discrete constraint} \\ \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) & \rightarrow \text{differential constraint} \\ \mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}(t_i), \mathbf{m}(t_i)\big) = \mathbf{0} & \rightarrow \text{continuous constraint}\end{array}\right.$$

Decomposition

We recall the problem:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big), \\ \mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}(t_i), \mathbf{m}(t_i)\big) = \mathbf{0}, \\ \mathbf{y}(t_i) \in [\mathbf{y}](t_i),\ \mathbf{m}(t_i) \in \mathbb{M}, \end{cases} \tag{1}$$

Decomposition

We recall the problem:

$$\left\{\begin{array}{l}\dot{\mathbf{x}}(t)=\mathbf{f}\big(\mathbf{x}(t),\mathbf{u}(t)\big),\\ \mathbf{g}\big(\mathbf{x}(t_i),\mathbf{y}(t_i),\mathbf{m}(t_i)\big)=\mathbf{0},\\ \mathbf{y}(t_i)\in[\mathbf{y}](t_i),\ \mathbf{m}(t_i)\in\mathbb{M},\end{array}\right. \tag{1}$$

with:

$$\mathbf{g}\big(\mathbf{x}(t_i),\mathbf{y}(t_i),\mathbf{m}(t_i)\big)=\left(\begin{array}{c}p_x(t_i)\\ p_y(t_i)\end{array}\right)+\rho(t_i)\cdot\left(\begin{array}{c}\cos(\psi(t_i)+\varphi(t_i))\\ \sin(\psi(t_i)+\varphi(t_i))\end{array}\right)-\left(\begin{array}{c}m_x(t_i)\\ m_y(t_i)\end{array}\right)$$

Decomposition

We recall the problem:

$$\left\{\begin{array}{l}\dot{\mathbf{x}}(t)=\mathbf{f}\big(\mathbf{x}(t),\mathbf{u}(t)\big),\\ \mathbf{g}\big(\mathbf{x}(t_i),\mathbf{y}(t_i),\mathbf{m}(t_i)\big)=\mathbf{0},\\ \mathbf{y}(t_i)\in[\mathbf{y}](t_i),\ \ \mathbf{m}(t_i)\in\mathbb{M},\end{array}\right. \tag{1}$$

with:

$$\mathbf{g}\big(\mathbf{x}(t_i),\mathbf{y}(t_i),\mathbf{m}(t_i)\big)= \\ \begin{pmatrix}p_x(t_i)\\ p_y(t_i)\end{pmatrix}+\rho(t_i)\cdot\begin{pmatrix}\cos(\psi(t_i)+\varphi(t_i))\\ \sin(\psi(t_i)+\varphi(t_i))\end{pmatrix}-\begin{pmatrix}m_x(t_i)\\ m_y(t_i)\end{pmatrix}$$

These equations can be broken down into:

$$\left\{\begin{array}{rl}(i) & \dot{\mathbf{x}}(\cdot)=\mathbf{f}\big(\mathbf{x}(\cdot),\mathbf{u}(\cdot)\big)\\ (ii) & \mathbf{m}(t_i)\in\mathbb{M}\\ (iii) & \mathbf{a}_i=\mathbf{m}(t_i)-\mathbf{p}(t_i)\\ (iv) & \alpha_i=\psi(t_i)+\varphi(t_i)\\ (v) & \mathbf{a}_i=\rho\cdot\begin{pmatrix}\cos\alpha_i\\ \sin\alpha_i\end{pmatrix}\end{array}\right. \tag{2}$$

Decomposition

We recall the problem:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big), \\ \mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}(t_i), \mathbf{m}(t_i)\big) = \mathbf{0}, \\ \mathbf{y}(t_i) \in [\mathbf{y}](t_i),\ \ \mathbf{m}(t_i) \in \mathbb{M}, \end{cases} \tag{1}$$

with:

$$\mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}(t_i), \mathbf{m}(t_i)\big) = \begin{pmatrix} p_x(t_i) \\ p_y(t_i) \end{pmatrix} + \rho(t_i) \cdot \begin{pmatrix} \cos(\psi(t_i) + \varphi(t_i)) \\ \sin(\psi(t_i) + \varphi(t_i)) \end{pmatrix} - \begin{pmatrix} m_x(t_i) \\ m_y(t_i) \end{pmatrix}$$

These equations can be broken down into:

$$\begin{cases} (i) & \dot{\mathbf{x}}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot), \mathbf{u}(\cdot)\big) & \rightarrow \mathcal{L}_{\frac{d}{dt}} \\ (ii) & \mathbf{m}(t_i) \in \mathbb{M} \\ (iii) & \mathbf{a}_i = \mathbf{m}(t_i) - \mathbf{p}(t_i) \\ (iv) & \alpha_i = \psi(t_i) + \varphi(t_i) \\ (v) & \mathbf{a}_i = \rho \cdot \begin{pmatrix} \cos \alpha_i \\ \sin \alpha_i \end{pmatrix} \end{cases} \tag{2}$$

Decomposition

We recall the problem:

$$\left\{\begin{array}{l}\dot{\mathbf{x}}(t)=\mathbf{f}\big(\mathbf{x}(t),\mathbf{u}(t)\big),\\ \mathbf{g}\big(\mathbf{x}(t_i),\mathbf{y}(t_i),\mathbf{m}(t_i)\big)=\mathbf{0},\\ \mathbf{y}(t_i)\in[\mathbf{y}](t_i),\ \ \mathbf{m}(t_i)\in\mathbb{M},\end{array}\right. \tag{1}$$

with:

$$\mathbf{g}\big(\mathbf{x}(t_i),\mathbf{y}(t_i),\mathbf{m}(t_i)\big)=\\ \begin{pmatrix}p_x(t_i)\\ p_y(t_i)\end{pmatrix}+\rho(t_i)\cdot\begin{pmatrix}\cos(\psi(t_i)+\varphi(t_i))\\ \sin(\psi(t_i)+\varphi(t_i))\end{pmatrix}-\begin{pmatrix}m_x(t_i)\\ m_y(t_i)\end{pmatrix}$$

These equations can be broken down into:

$$\left\{\begin{array}{rll}(i) & \dot{\mathbf{x}}(\cdot)=\mathbf{f}\big(\mathbf{x}(\cdot),\mathbf{u}(\cdot)\big) & \rightarrow\mathcal{L}_{\frac{d}{dt}}\\ (ii) & \mathbf{m}(t_i)\in\mathbb{M} & \\ (iii) & \mathbf{a}_i=\mathbf{m}(t_i)-\mathbf{p}(t_i) & \rightarrow\mathcal{L}_{\text{algeb}}\\ (iv) & \alpha_i=\psi(t_i)+\varphi(t_i) & \rightarrow\mathcal{L}_{\text{algeb}}\\ (v) & \mathbf{a}_i=\rho\cdot\begin{pmatrix}\cos\alpha_i\\ \sin\alpha_i\end{pmatrix} & \end{array}\right. \tag{2}$$

Decomposition

We recall the problem:

$$\left\{\begin{array}{l}\dot{\mathbf{x}}(t)=\mathbf{f}\big(\mathbf{x}(t),\mathbf{u}(t)\big),\\ \mathbf{g}\big(\mathbf{x}(t_i),\mathbf{y}(t_i),\mathbf{m}(t_i)\big)=\mathbf{0},\\ \mathbf{y}(t_i)\in[\mathbf{y}](t_i),\ \ \mathbf{m}(t_i)\in\mathbb{M},\end{array}\right. \tag{1}$$

with:

$$\mathbf{g}\big(\mathbf{x}(t_i),\mathbf{y}(t_i),\mathbf{m}(t_i)\big)=\begin{pmatrix}p_x(t_i)\\p_y(t_i)\end{pmatrix}+\rho(t_i)\cdot\begin{pmatrix}\cos(\psi(t_i)+\varphi(t_i))\\\sin(\psi(t_i)+\varphi(t_i))\end{pmatrix}-\begin{pmatrix}m_x(t_i)\\m_y(t_i)\end{pmatrix}$$

These equations can be broken down into:

$$\left\{\begin{array}{rll}(i) & \dot{\mathbf{x}}(\cdot)=\mathbf{f}\big(\mathbf{x}(\cdot),\mathbf{u}(\cdot)\big) & \rightarrow\mathcal{L}_{\frac{d}{dt}}\\ (ii) & \mathbf{m}(t_i)\in\mathbb{M} & \\ (iii) & \mathbf{a}_i=\mathbf{m}(t_i)-\mathbf{p}(t_i) & \rightarrow\mathcal{L}_{\text{algeb}}\\ (iv) & \alpha_i=\psi(t_i)+\varphi(t_i) & \rightarrow\mathcal{L}_{\text{algeb}}\\ (v) & \mathbf{a}_i=\rho\cdot\begin{pmatrix}\cos\alpha_i\\\sin\alpha_i\end{pmatrix} & \rightarrow\mathcal{L}_{\text{polar}}\end{array}\right. \tag{2}$$

Decomposition

We recall the problem:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big), \\ \mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}(t_i), \mathbf{m}(t_i)\big) = \mathbf{0}, \\ \mathbf{y}(t_i) \in [\mathbf{y}](t_i),\ \ \mathbf{m}(t_i) \in \mathbb{M}, \end{cases} \tag{1}$$

with:

$$\mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}(t_i), \mathbf{m}(t_i)\big) = \\ \begin{pmatrix} p_x(t_i) \\ p_y(t_i) \end{pmatrix} + \rho(t_i) \cdot \begin{pmatrix} \cos(\psi(t_i) + \varphi(t_i)) \\ \sin(\psi(t_i) + \varphi(t_i)) \end{pmatrix} - \begin{pmatrix} m_x(t_i) \\ m_y(t_i) \end{pmatrix}$$

These equations can be broken down into:

$$\begin{cases} (i) & \dot{\mathbf{x}}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot), \mathbf{u}(\cdot)\big) & \rightarrow \mathcal{L}_{\frac{d}{dt}} \\ (ii) & \mathbf{m}(t_i) \in \mathbb{M} & \rightarrow \mathcal{L}_{\text{constel.}} \\ (iii) & \mathbf{a}_i = \mathbf{m}(t_i) - \mathbf{p}(t_i) & \rightarrow \mathcal{L}_{\text{algeb}} \\ (iv) & \alpha_i = \psi(t_i) + \varphi(t_i) & \rightarrow \mathcal{L}_{\text{algeb}} \\ (v) & \mathbf{a}_i = \rho \cdot \begin{pmatrix} \cos\alpha_i \\ \sin\alpha_i \end{pmatrix} & \rightarrow \mathcal{L}_{\text{polar}} \end{cases} \tag{2}$$

Association constraint: constellation contractor

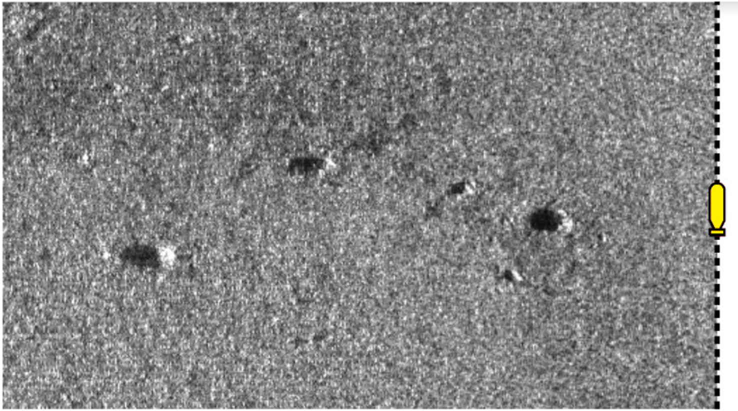
Let us consider a constellation of ℓ points $\mathbb{M} = \{\mathbf{m}_1, \dots, \mathbf{m}_\ell\}$ of $\mathbb{R}^d$ and a box $[\mathbf{x}] \in \mathbb{IR}^d$. We want to compute the smallest box $\mathcal{C}([\mathbf{x}])$ containing $\mathbb{M} \cap [\mathbf{x}]$, or equivalently:

$$\mathcal{C}([\mathbf{x}]) = \bigsqcup_i \big([\mathbf{x}] \cap \{\mathbf{m}_i\} \big), \tag{3}$$

where $\bigsqcup$, called *squared union*, returns the smallest box enclosing the union of its arguments.

Constraint network on data

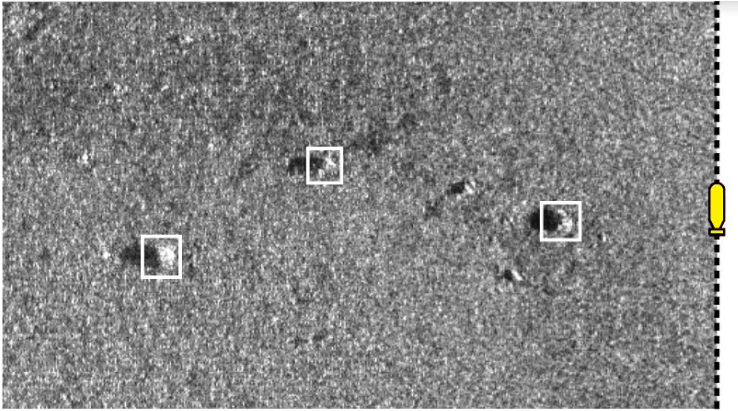
Constraint $\mathbf{m}(t_i) \in \mathbb{M}$:
An observation $\mathbf{m}(t_i)$ is related to one of the known seamarks $\mathbb{M}$.



Perception of the seabed with a side-scan sonar.

Constraint network on data

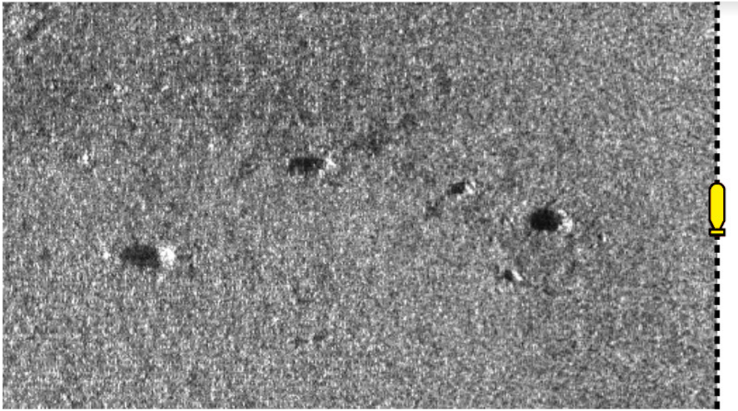
Constraint $\mathbf{m}(t_i) \in \mathbb{M}$:
An observation $\mathbf{m}(t_i)$ is related to one of the known seamarks $\mathbb{M}$.



Seamarks are already known with some uncertainty.

Constraint network on data

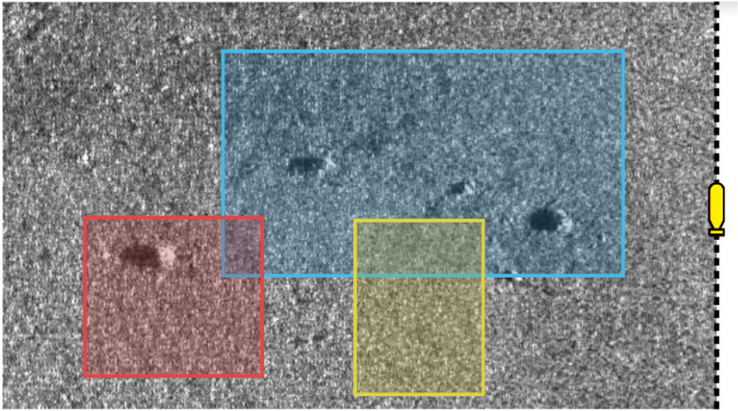
Constraint $\mathbf{m}(t_i) \in \mathbb{M}$:
An observation $\mathbf{m}(t_i)$ is related to one of the known seamarks $\mathbb{M}$.



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

Constraint network on data

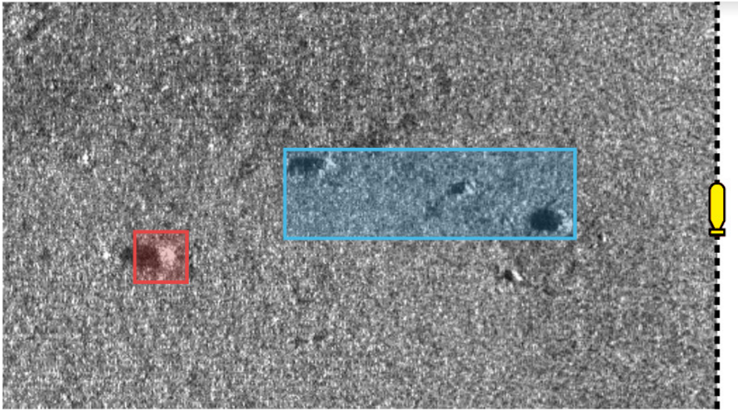
Constraint $\mathbf{m}(t_i) \in \mathbb{M}$:
An observation $\mathbf{m}(t_i)$ is related to one of the known seamarks $\mathbb{M}$.



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

Constraint network on data

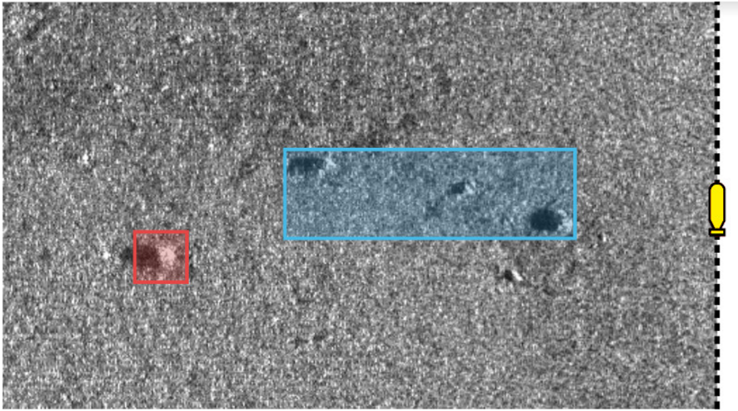
Constraint $\mathbf{m}(t_i) \in \mathbb{M}$:
An observation $\mathbf{m}(t_i)$ is related to one of the known seamarks $\mathbb{M}$.



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

Constraint network on data

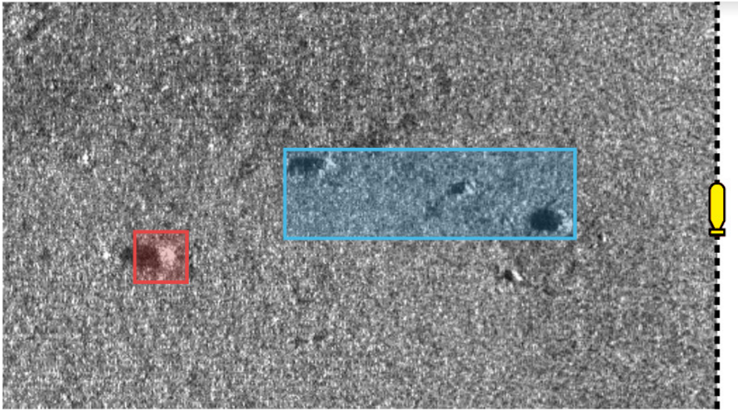
Constraint $\mathbf{m}(t_i) \in \mathbb{M}$:
An observation $\mathbf{m}(t_i)$ is related to one of the known seamarks $\mathbb{M}$.



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

Constraint network on data

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.

Application

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

Special thanks to DGA-TN Brest (formerly GESMA)

Application

Video

Results on actual data

Map: 133 objects. 54 detections in sonar images.

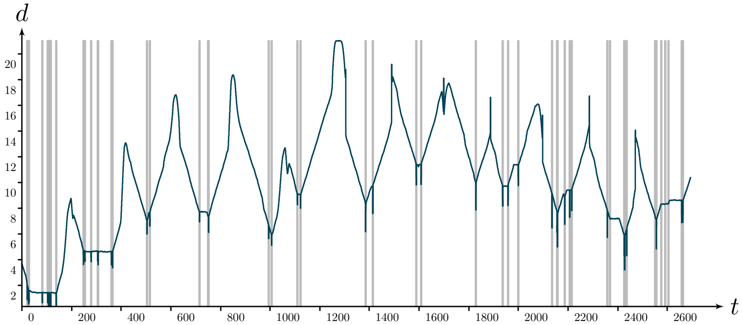
Table: Numerical results of the iterative localization algorithm.

#	time(s)	#min	#max	#ok
1	0.278	133	133	0
2	0.271	14	64	0
3	0.268	5	52	0
4	0.266	1	34	2
5	0.271	1	16	39
6	0.267	1	4	48
7	0.266	1	3	49
8	0.266	1	3	50
9	0.266	1	2	51

- #min: minimal number of objects included in the $[\mathbf{m}](t_i)$
- #max: maximal number of objects included in the $[\mathbf{m}](t_i)$
- #ok: number of correct associations

Results on actual data

The initial position of the robot is **not known before the contractions**, and is finally estimated with an **error of 3.6m** in the worst case:



$d = w([\mathbf{p}])$ when reaching a contracting fixed point. Computation time < 2.5s.

d: diameter of each box $[x_1](t) \times [x_2](t)$, *i.e.* localization error in the very worst case.