Nonlinear interval observers

Luc Jaulin, ENSIETA, Brest

www.ensieta.fr/jaulin/

GREYC, Caen, March 11 2010

1 Interval approach

1.1 Interval calculus

If $\diamond \in \{+, -, ., /, \max, \min\}$, we have

$$[x] \diamond [y] = [\{x \diamond y \mid x \in [x], y \in [y]\}] .$$

For instance,

$$\begin{aligned} [-1, 3] + [2, 5] &= [1, 8], \\ [-1, 3].[2, 5] &= [-5, 15], \\ [-1, 3]/[2, 5] &= [-\tfrac{1}{2}, \tfrac{3}{2}], \end{aligned}$$

If $f \in \{\cos, \sin, \text{sqr}, \text{sqrt}, \log, \exp, \dots\}$

$$f([x]) = [\{f(x) \mid x \in [x]\}] .$$

For instance,

$$\begin{aligned}
\sin([0,\pi]) &= [0,1], \\
\text{sqr}([-1,3]) &= [-1,3]^2 = [0,9], \\
\text{abs}([-7,1]) &= [0,7], \\
\text{sqrt}([-10,4]) &= \sqrt{[-10,4]} = [0,2], \\
\log([-2,-1]) &= \emptyset.
\end{aligned}$$

1.2 Constraint projection

Consider three variables x, y, z such that

$$\begin{aligned} x &\in [-\infty, 5], \\ y &\in [-\infty, 4], \\ z &\in [6, \infty], \\ z &= x + y. \end{aligned}$$

The values < 2 for x, < 1 for y and > 9 for z are inconsistent.

1.3 Method for projection

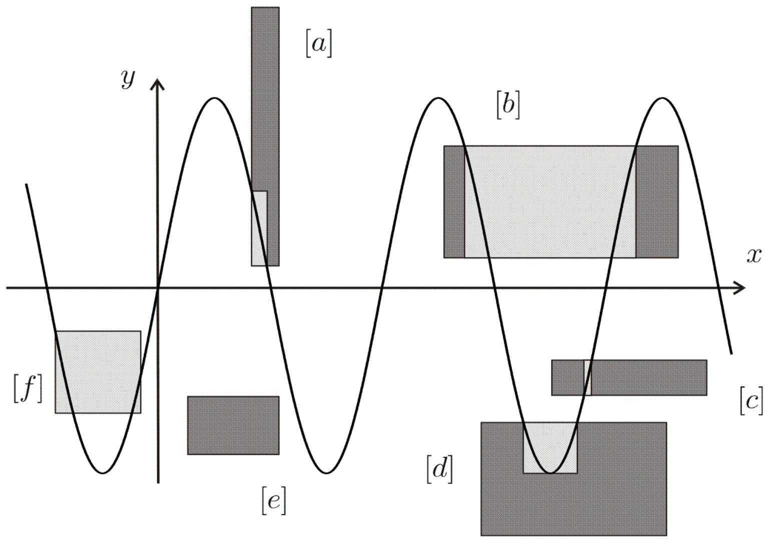
Since $x \in [-\infty, 5], y \in [-\infty, 4], z \in [6, \infty]$ and $z = x + y$, we have

$$\begin{aligned}
z = x + y \Rightarrow \quad z \in \ & [6, \infty] \cap ([-\infty, 5] + [-\infty, 4]) \\
& = [6, \infty] \cap [-\infty, 9] = [6, 9]. \\
x = z - y \Rightarrow \quad x \in \ & [-\infty, 5] \cap ([6, \infty] - [-\infty, 4]) \\
& = [-\infty, 5] \cap [2, \infty] = [2, 5]. \\
y = z - x \Rightarrow \quad y \in \ & [-\infty, 4] \cap ([6, \infty] - [-\infty, 5]) \\
& = [-\infty, 4] \cap [1, \infty] = [1, 4].
\end{aligned}$$

For the constraint

$$y = \sin x,\ \ x \in [x], y \in [y]$$

the problem is a more complicated.



1.4 Interval solver

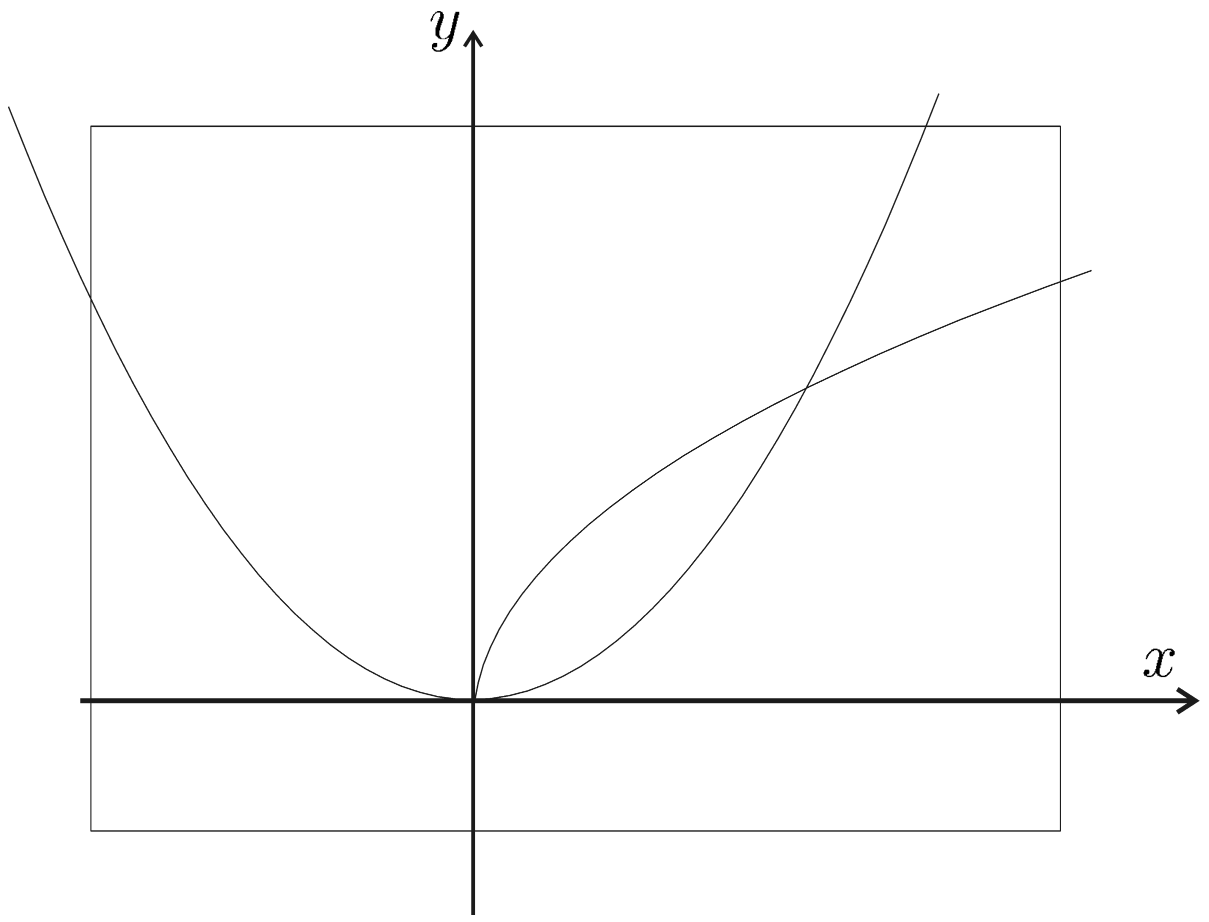
Example. Consider the system.

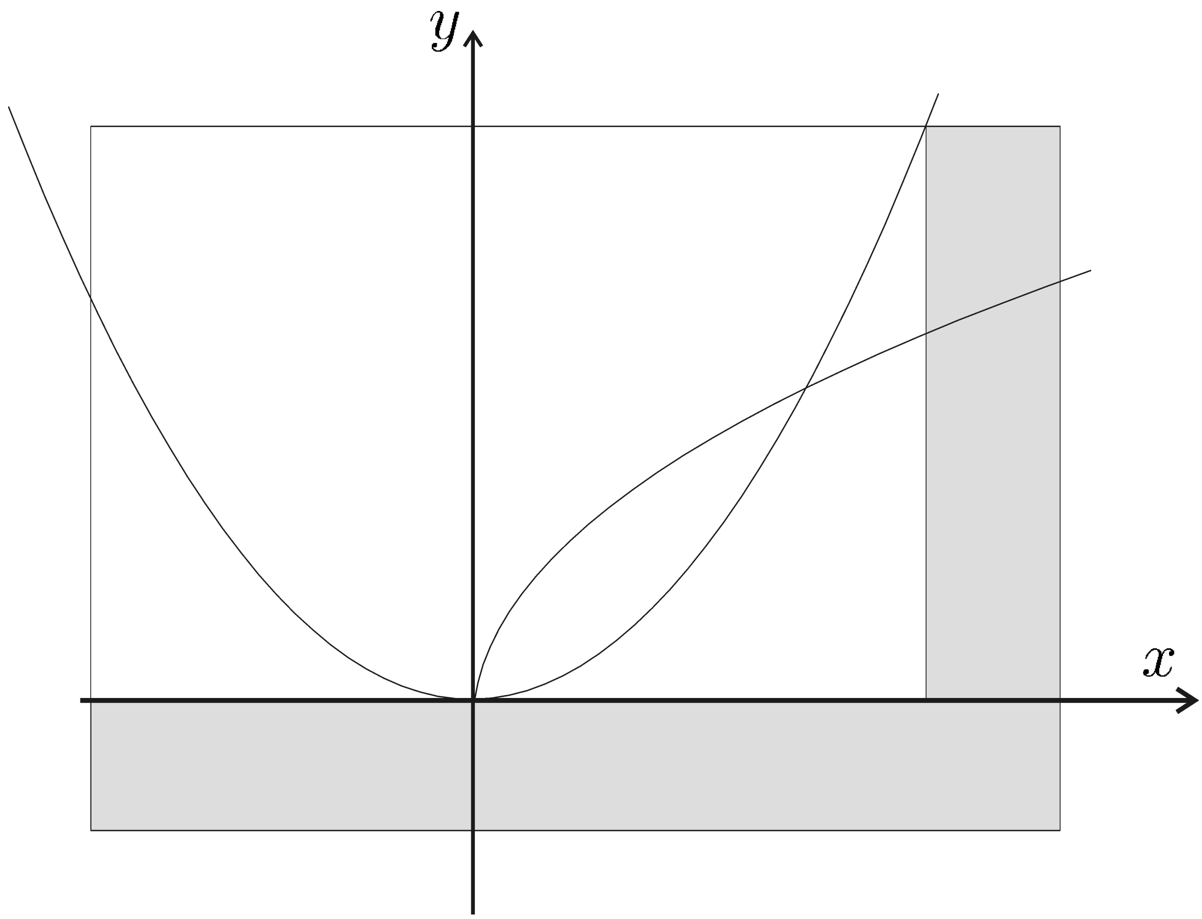
$$\begin{aligned} y &= x^2 \\ y &= \sqrt{x}. \end{aligned}$$

We build two contractors

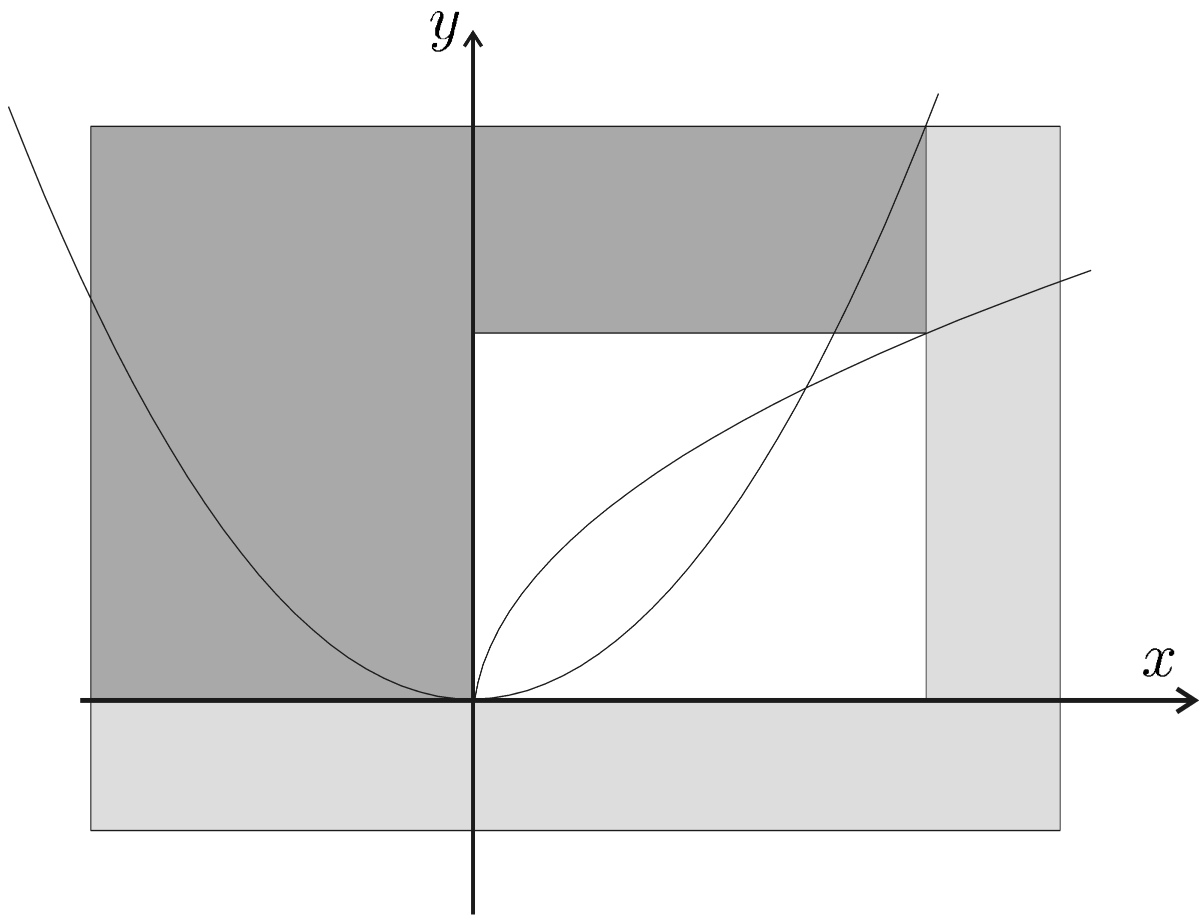
$$\mathcal{C}_1 : \begin{cases} [y] = [y] \cap [x]^2 \\ [x] = [x] \cap \sqrt{[y]} \end{cases} \quad \text{associated to } y = x^2$$

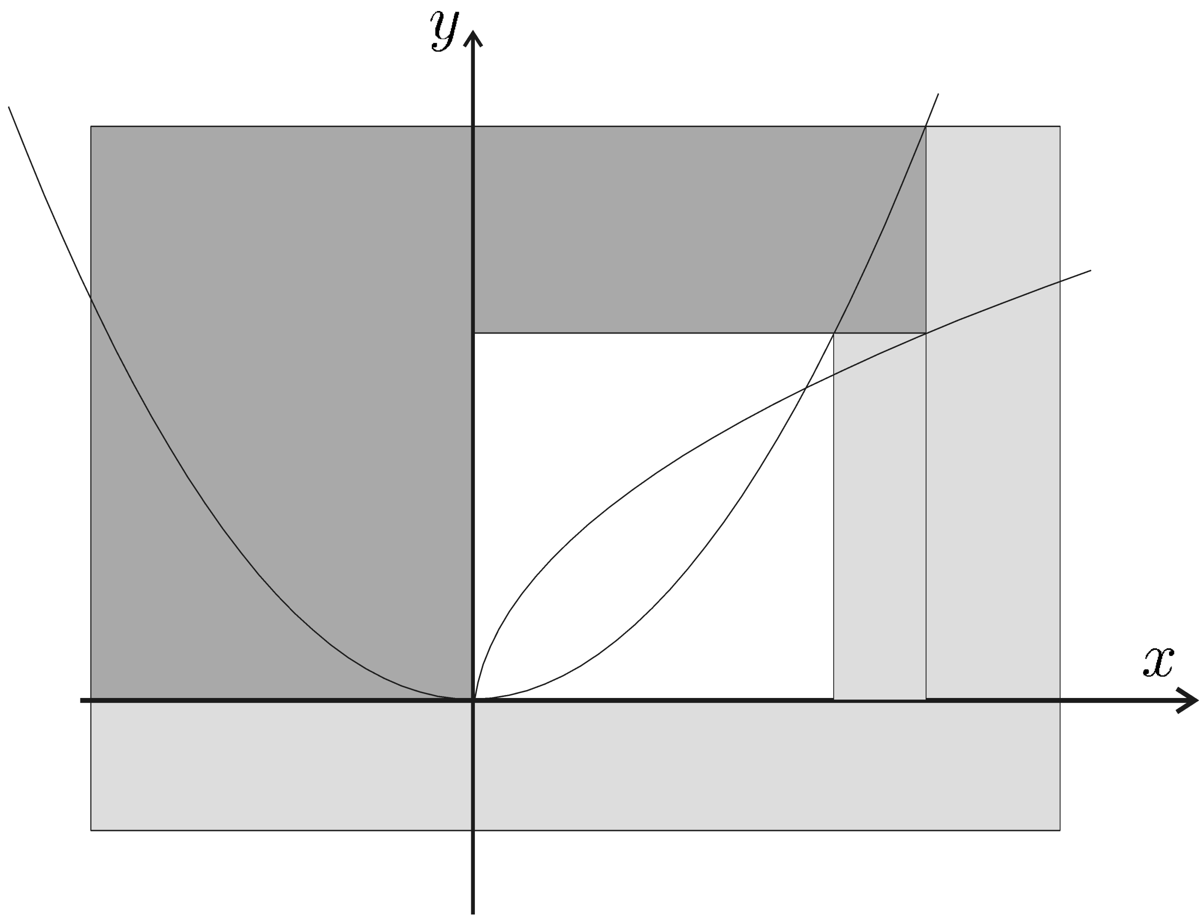
$$\mathcal{C}_2 : \begin{cases} [y] = [y] \cap \sqrt{[x]} \\ [x] = [x] \cap [y]^2 \end{cases} \quad \text{associated to } y = \sqrt{x}$$

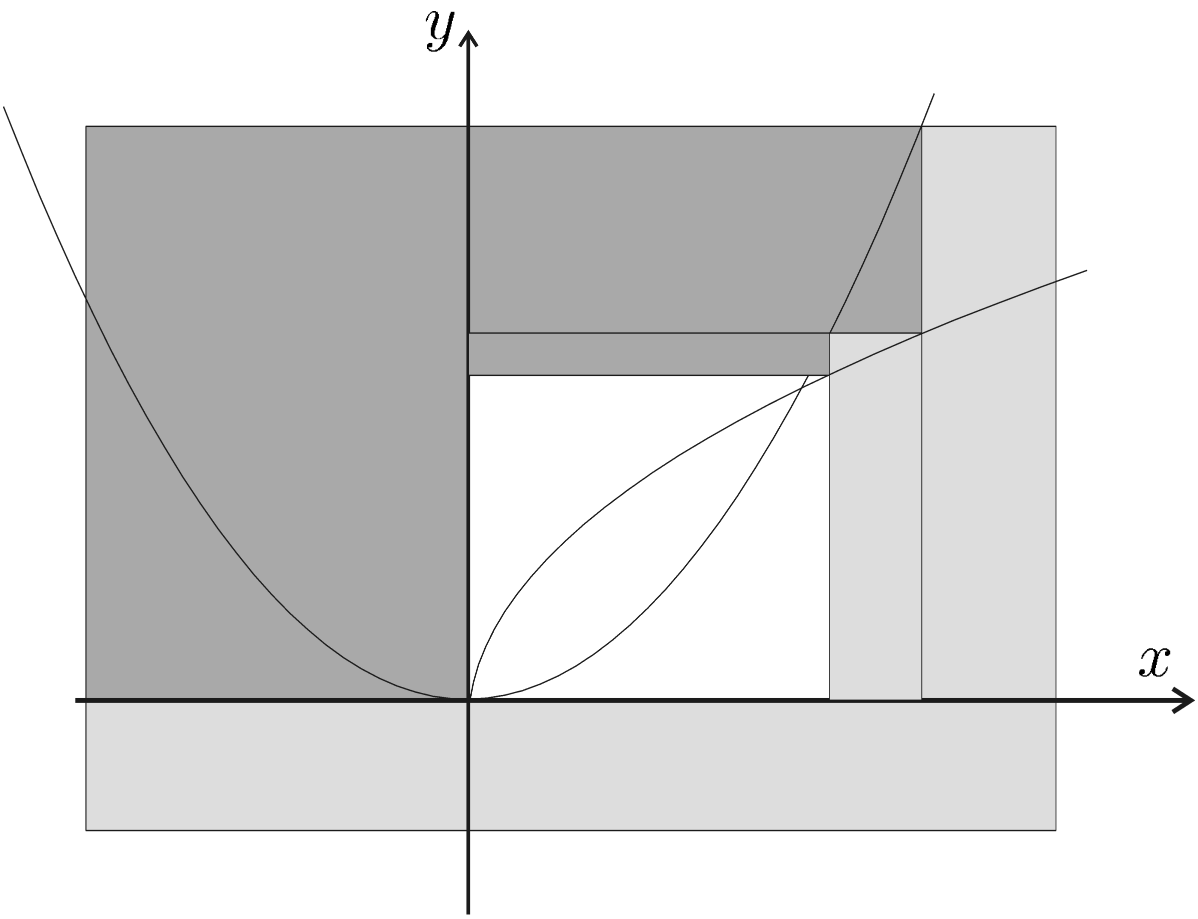

y
x



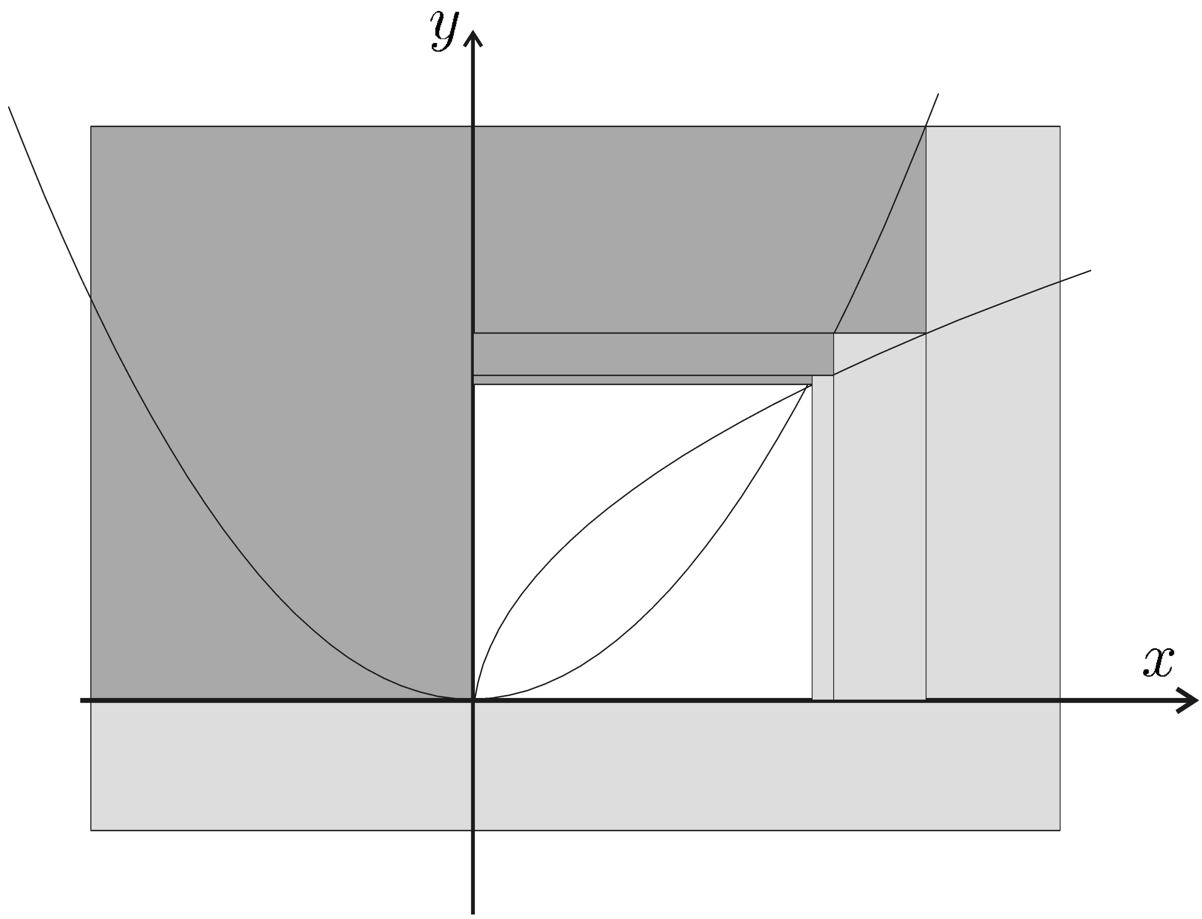
y
x

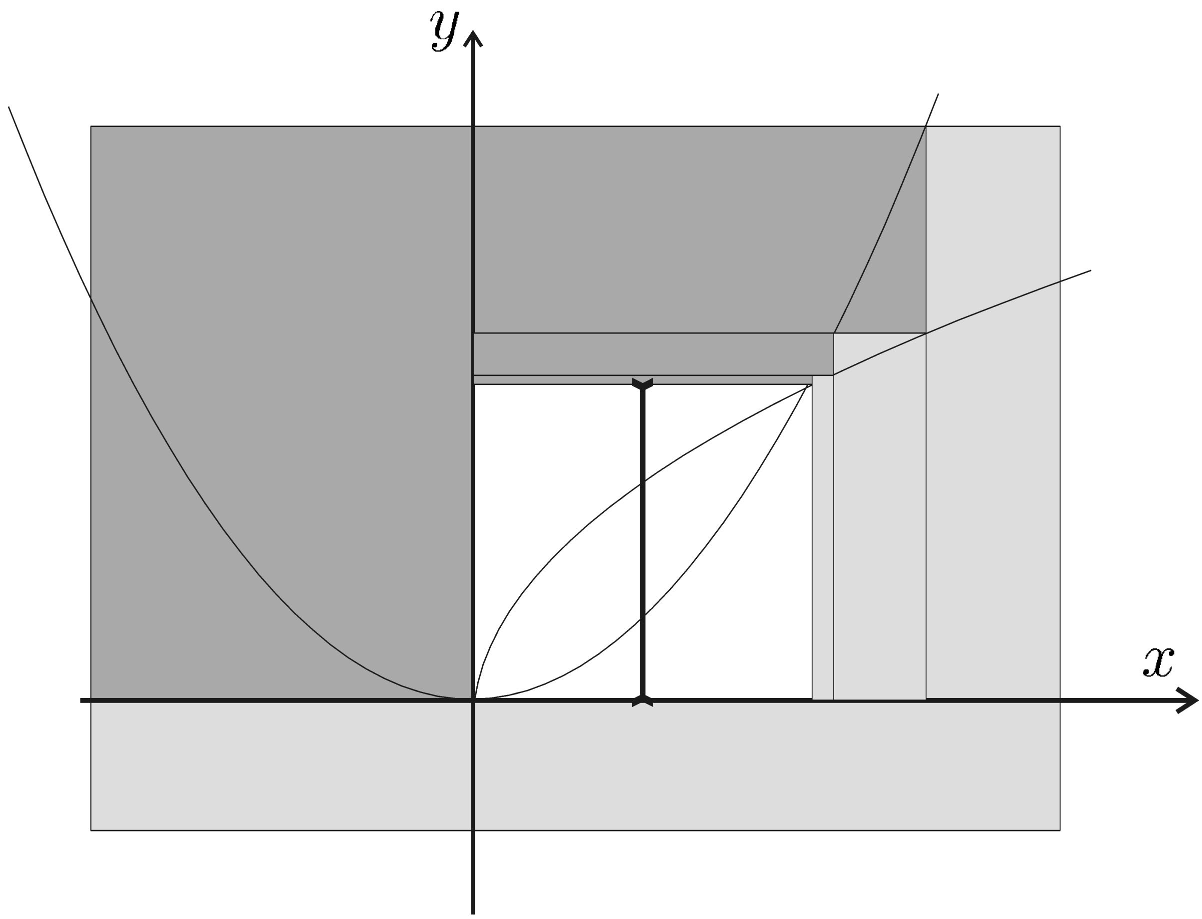

y
x

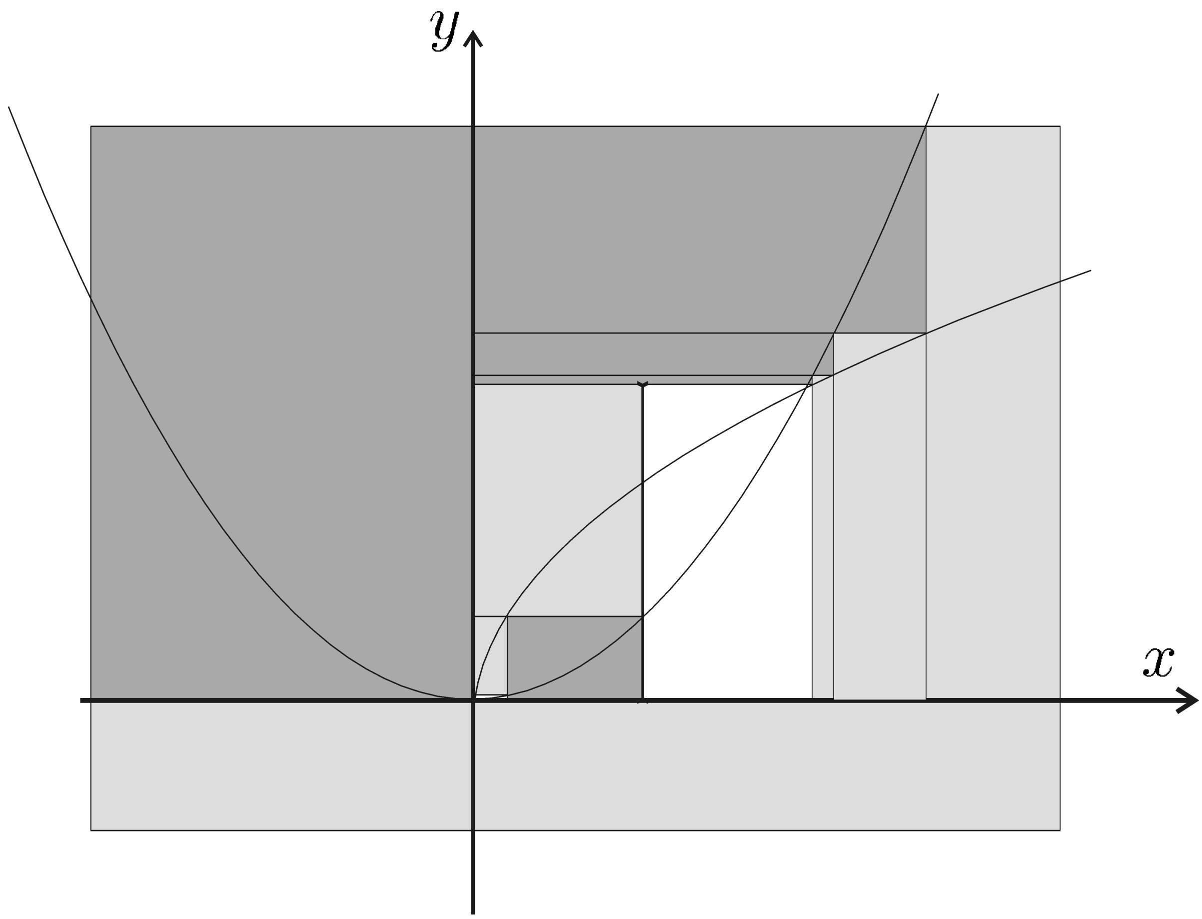

y
x

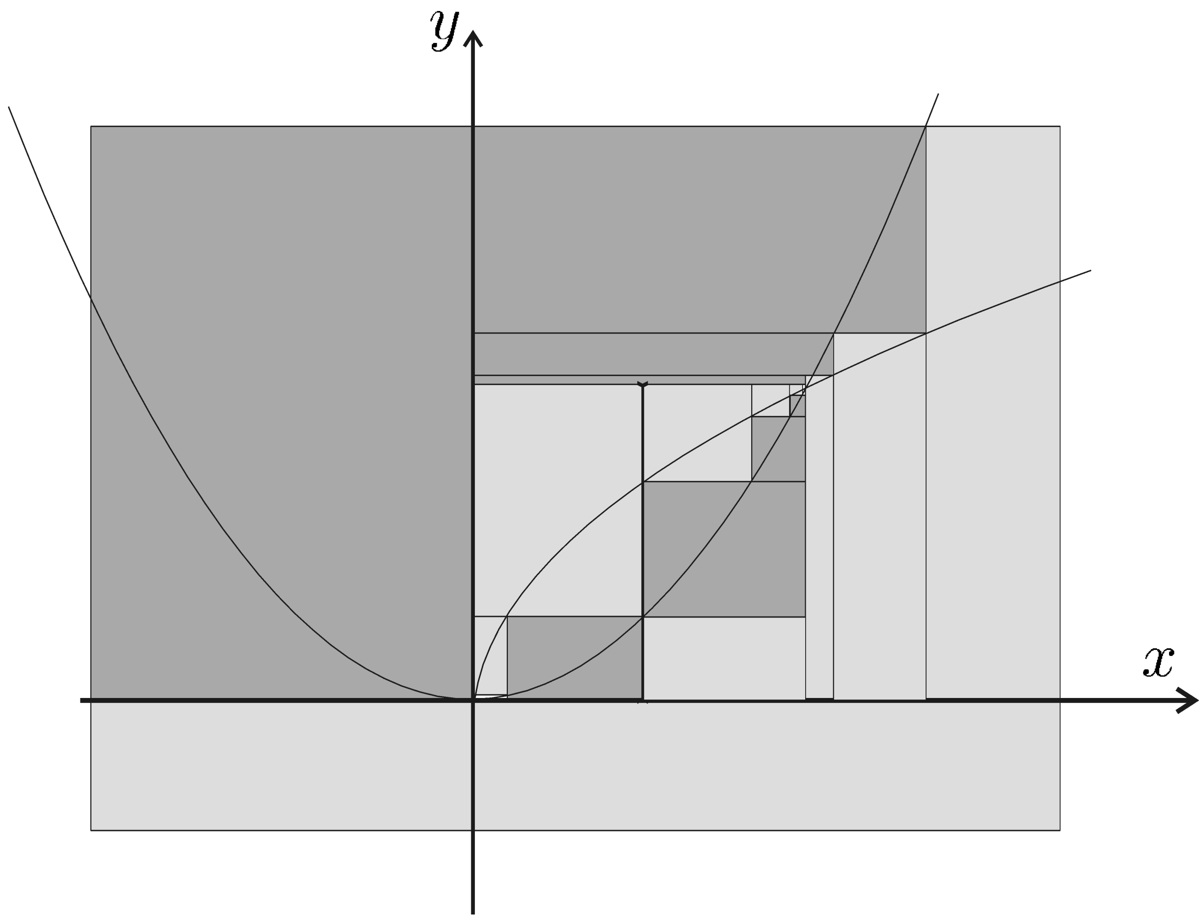


y
x


y
x


y
x


y
x


y
x

1.5 Decomposition

For non-primitive constraints, a decomposition is required. For instance

$$x + x \sin(y) \le 0,$$
$$x \in [-1, 1], y \in [-1, 1]$$

is decomposed into

$$\left\{ \begin{array}{l} a = \sin(y) \\ b = x * a \\ c = a + b \end{array} \right. , \qquad \begin{array}{ll} x \in [-1, 1] & a \in]-\infty, \infty[\\ y \in [-1, 1] & b \in]-\infty, \infty[\\ & c \in]-\infty, 0[\end{array}$$

1.6 QUIMPER

Quimper : QUick Interval Modeling and Programming in a bounded-ERror context.

Quimper is an interpreted language for set computation.

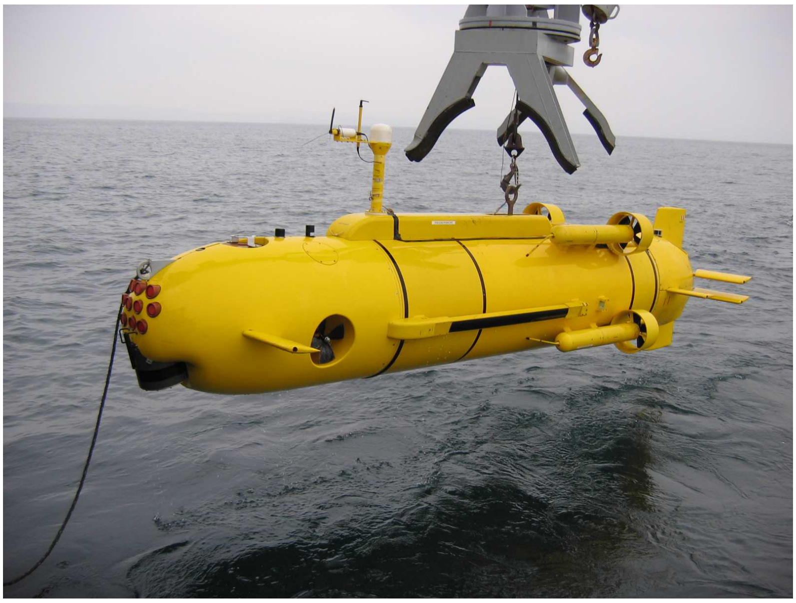
A Quimper program can be described by a list of contractors.

`http://ibex-lib.org/`

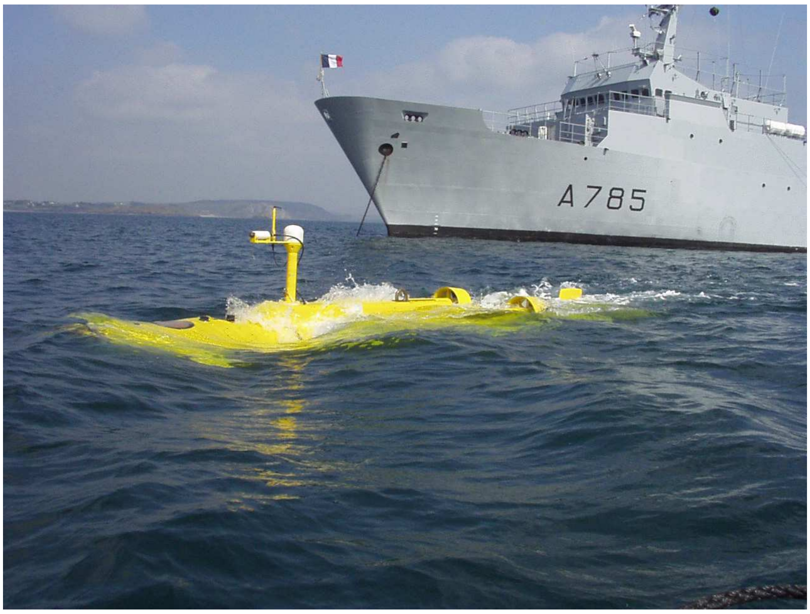
Show Setdémo

Show Proj2d

2 Redermor



The *Redermor*, GESMA



The *Redermor* at the surface

Show simulation

Why choosing an interval constraint approach for SLAM ?

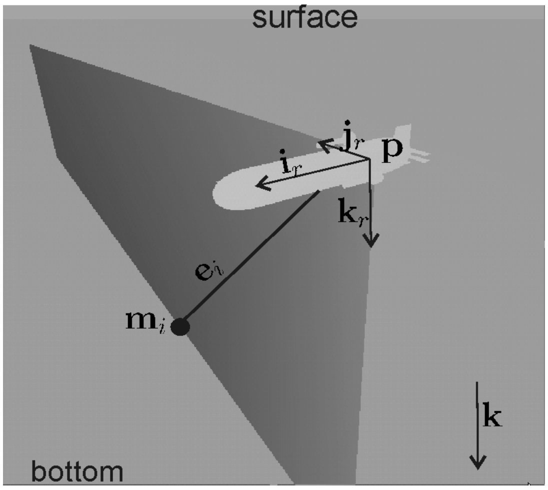
1) A reliable method is needed.
2) The model is nonlinear.
3) The pdf of the noises are unknown.
4) Reliable error bounds are provided by the sensors.
5) A huge number of redundant data are available.

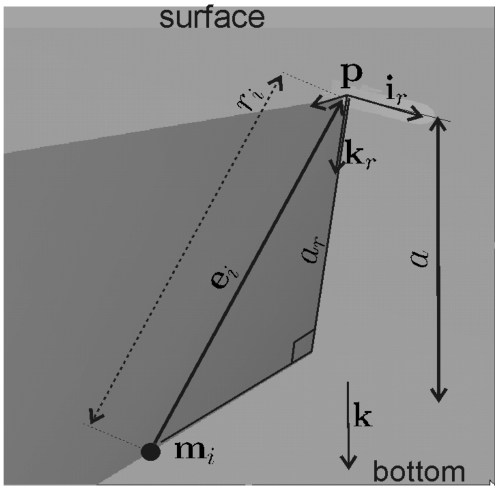
2.1 Sensors

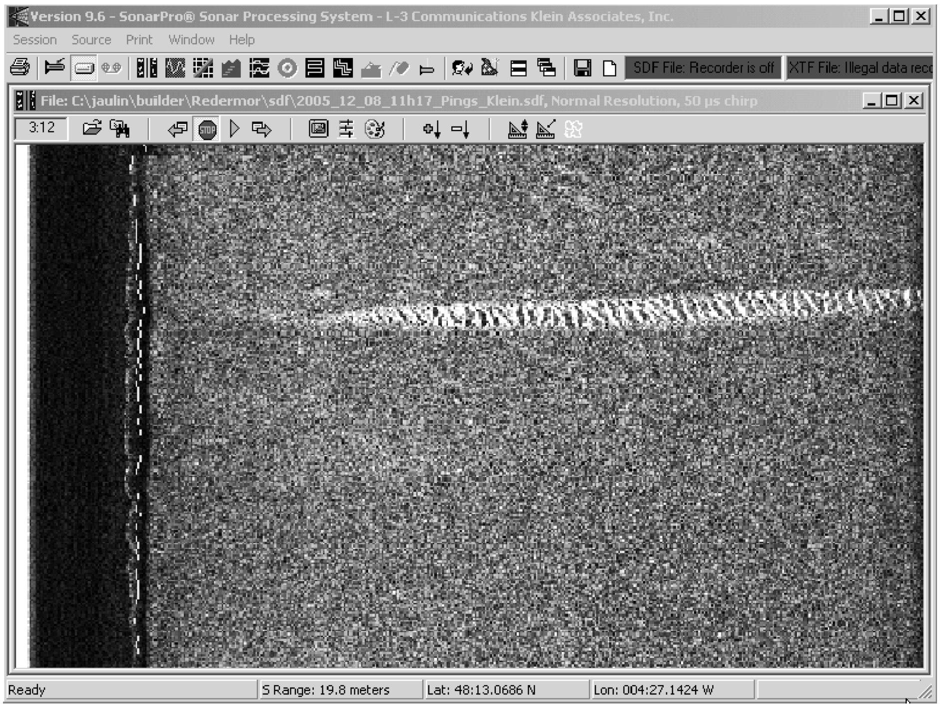
A GPS (Global positioning system) at the surface only.

$$t_0 = 6000 \text{ s}, \quad \ell^0 = (-4.4582279^\circ, 48.2129206^\circ) \pm 2.5m$$
$$t_f = 12000 \text{ s}, \quad \ell^f = (-4.4546607^\circ, 48.2191297^\circ) \pm 2.5m$$

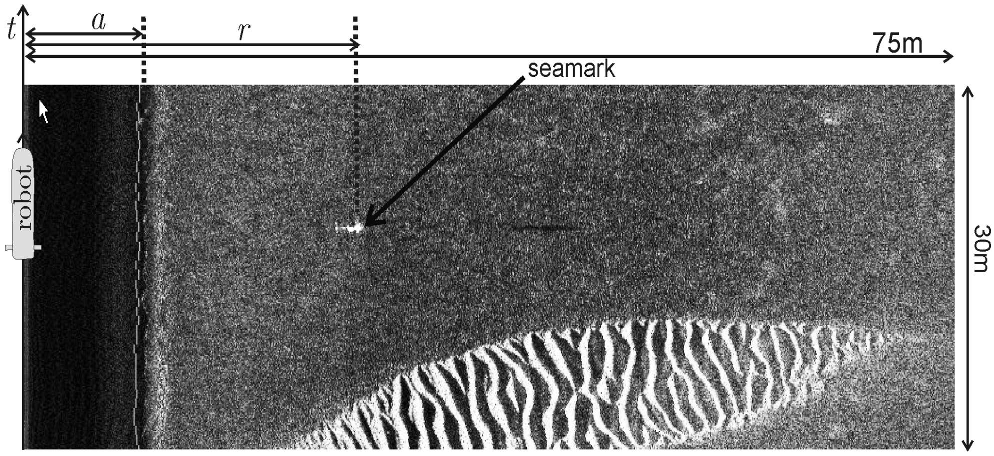
A sonar (KLEIN 5400 side scan sonar). Gives the distance r between the robot to the detected object.







Screenshot of SonarPro



Detection of a mine using SonarPro

A Loch-Doppler. Returns the speed of the robot $\mathbf{v}_r$ and the altitude a of the robot $\pm$ 10cm.

A Gyrocompass (Octans III from IXSEA). Returns the roll ϕ, the pitch θ and the head ψ.

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} \in \begin{pmatrix} \tilde{\phi} \\ \tilde{\theta} \\ \tilde{\psi} \end{pmatrix} + \begin{pmatrix} 1.75 \times 10^{-4}.\,[-1,1] \\ 1.75 \times 10^{-4}.\,[-1,1] \\ 5.27 \times 10^{-3}.\,[-1,1] \end{pmatrix}.$$

2.2 Data

For each time $t \in \{6000.0, 6000.1, 6000.2, \ldots, 11999.4\}$, we get intervals for

$$\phi(t), \theta(t), \psi(t), v_r^x(t), v_r^y(t), v_r^z(t), a(t).$$

Six mines have been detected by the sonar:

i	0	1	2	3	4	5	6	7	8	9	10	11
$\tau(i)$	7054	7092	7374	7748	9038	9688	10024	10817	11172	11232	11279	11688
$\sigma(i)$	1	2	1	0	1	5	4	3	3	4	5	1
$\tilde{r}(i)$	52.42	12.47	54.40	52.68	27.73	26.98	37.90	36.71	37.37	31.03	33.51	15.05

2.3 Constraints satisfaction problem

$$t \in \{6000.0, 6000.1, 6000.2, \ldots, 11999.4\},$$

$$i \in \{0, 1, \ldots, 11\},$$

$$\begin{pmatrix} p_x(t) \\ p_y(t) \end{pmatrix} = 111120 \begin{pmatrix} 0 & 1 \\ \cos\left(\ell_y(t) * \frac{\pi}{180}\right) & 0 \end{pmatrix} \begin{pmatrix} \ell_x(t) - \ell_x^0 \\ \ell_y(t) - \ell_y^0 \end{pmatrix},$$

$$\mathbf{p}(t) = (p_x(t), p_y(t), p_z(t)),$$

$$\mathbf{R}_\psi(t) = \begin{pmatrix} \cos\psi(t) & -\sin\psi(t) & 0 \\ \sin\psi(t) & \cos\psi(t) & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{R}_\theta(t) = \begin{pmatrix} \cos\theta(t) & 0 & \sin\theta(t) \\ 0 & 1 & 0 \\ -\sin\theta(t) & 0 & \cos\theta(t) \end{pmatrix},$$

$$\mathbf{R}_{\varphi}(t) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi(t) & -\sin\varphi(t) \\ 0 & \sin\varphi(t) & \cos\varphi(t) \end{pmatrix},$$

$$\mathbf{R}(t) = \mathbf{R}_{\psi}(t).\mathbf{R}_{\theta}(t).\mathbf{R}_{\varphi}(t),$$

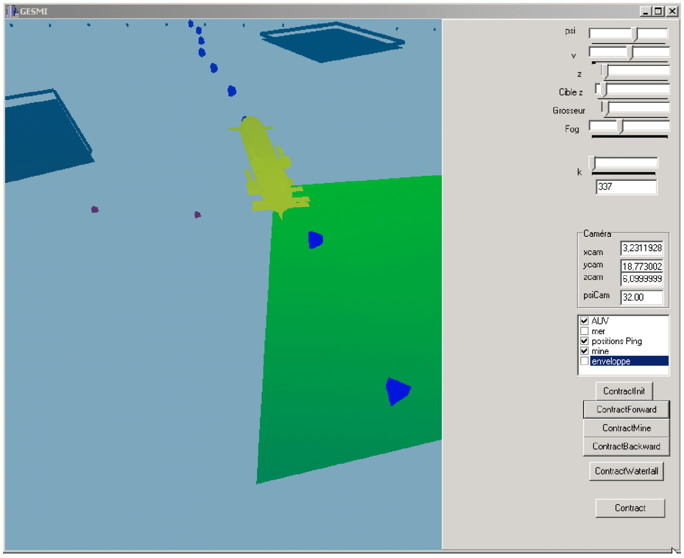
$$\dot{\mathbf{p}}(t) = \mathbf{R}(t).\mathbf{v}_r(t)$$

$$||\mathbf{m}(\sigma(i)) - \mathbf{p}(\tau(i))|| \ = r(i),$$

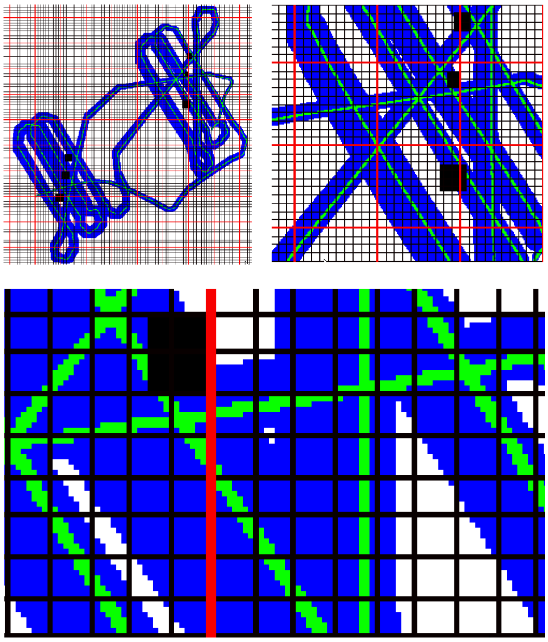
$$\mathbf{R}^{\mathsf{T}}(\tau(i))\,(\mathbf{m}(\sigma(i)) - \mathbf{p}(\tau(i))) \in [0] \times [0,\infty]^{\times 2},$$

$$m_z(\sigma(i)) - p_z(\tau(i)) - a(\tau(i)) \in [-0.5, 0.5].$$

2.4 GESMI



GESMI (Guaranteed Estimation of Sea Mines with Intervals)



Trajectory reconstructed by GESMI

```
//--------------------------------------------------
Constants
N = 59996; // Number of time steps
Variables
  R[N-1][3][3], // rotation matrices
  p[N][3], // positions
  v[N-1][3], // speed vectors
  phi[N-1],theta[N-1],psi[N-1]; // Euler angles
  px[N],py[N]; // for display only
//--------------------------------------------------
```

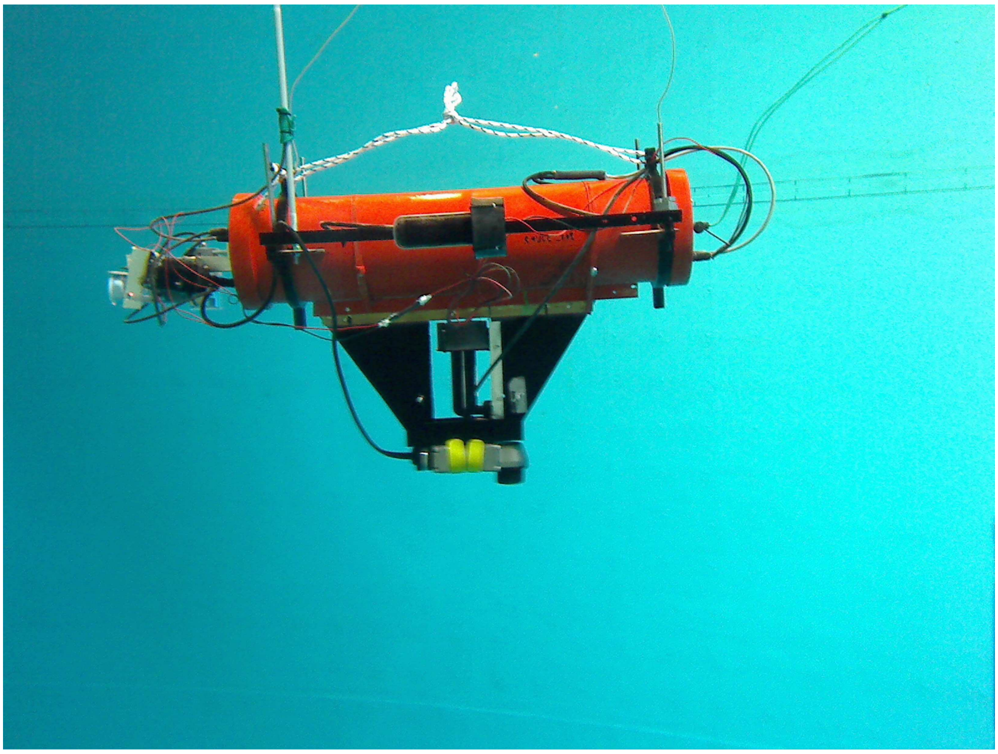
```
function R[3][3]=euler(phi,theta,psi)
  cphi = cos(phi);
  sphi = sin(phi);
  ctheta = cos(theta);
  stheta = sin(theta);
  cpsi = cos(psi);
  spsi = sin(psi);
  R[1][1]=ctheta*cpsi;
  R[1][2]=-cphi*spsi+stheta*cpsi*sphi;
  R[1][3]=spsi*sphi+stheta*cpsi*cphi;
  R[2][1]=ctheta*spsi;
  R[2][2]=cpsi*cphi+stheta*spsi*sphi;
  R[2][3]=-cpsi*sphi+stheta*cphi*spsi;
  R[3][1]=-stheta;
  R[3][2]=ctheta*sphi;
  R[3][3]=ctheta*cphi;
end
```

```
contractor-list rotation
  for k=1:N-1;
    R[k]=euler(phi[k],theta[k],psi[k]);
  end
end
//---------------------------------------------------
contractor-list statequ
  for k=1:N-1;
    p[k+1]=p[k]+0.1*R[k]*v[k];
  end
end
//---------------------------------------------------
contractor init
  inter k=1:N-1;
    rotation(k)
  end
end
```

```
contractor fwd
  inter k=1:N-1;
    statequ(k)
  end
end
//-------------------------
contractor bwd
  inter k=1:N-1;
    statequ(N-k)
  end
end
```

```
main
  p[1] :=read("gps_init.dat");
  v :=read("Quimper_v.dat");
  phi :=read("Quimper_phi.dat");
  theta :=read("Quimper_theta.dat");
  psi :=read("Quimper_psi.dat");
  init;
  fwd;
  bwd;
  column(p,px,1);
  column(p,py,2);
  print("--- Robot positions: ---");
  newplot("gesmi.dat");
  plot(px,py,color(rgb(1,1,1),rgb(0,0,0)));
end
```

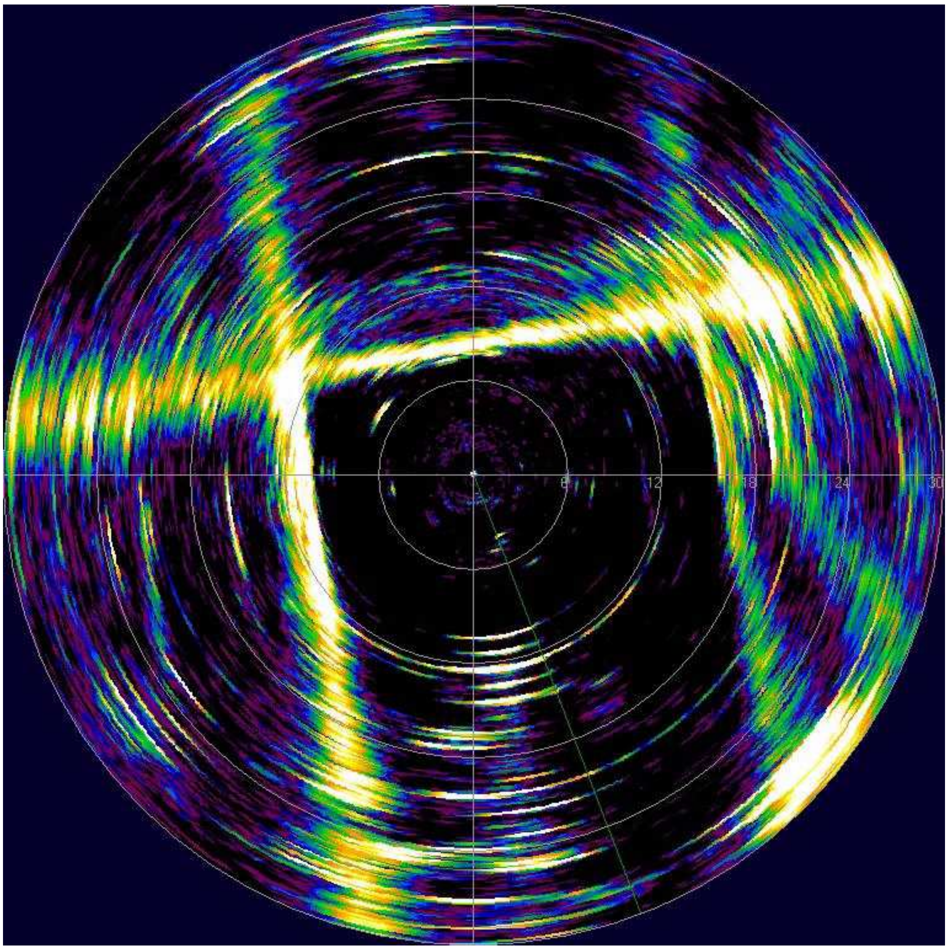
3 SAUC'ISSE



Robot SAUC'ISSE

Montrer une vidéo

3.1 Localization with sonar


6
12
18
24
30

3.2 Set-membership approach

$$\begin{cases} \mathbf{x}(k+1) &= \mathbf{f}_k(\mathbf{x}(k), \mathbf{n}(k)) \\ \mathbf{y}(k) &= \mathbf{g}_k(\mathbf{x}(k)), \end{cases}$$

with $\mathbf{n}(k) \in \mathbb{N}(k)$ and $\mathbf{y}(k) \in \mathbb{Y}(k)$.

Without outliers

$$\mathbb{X}(k+\mathbf{1}) = \mathbf{f}_k\left(\mathbb{X}(k) \cap \mathbf{g}_k^{-1}\left(\mathbb{Y}(k)\right), \quad \mathbb{N}\left(k\right)\right).$$

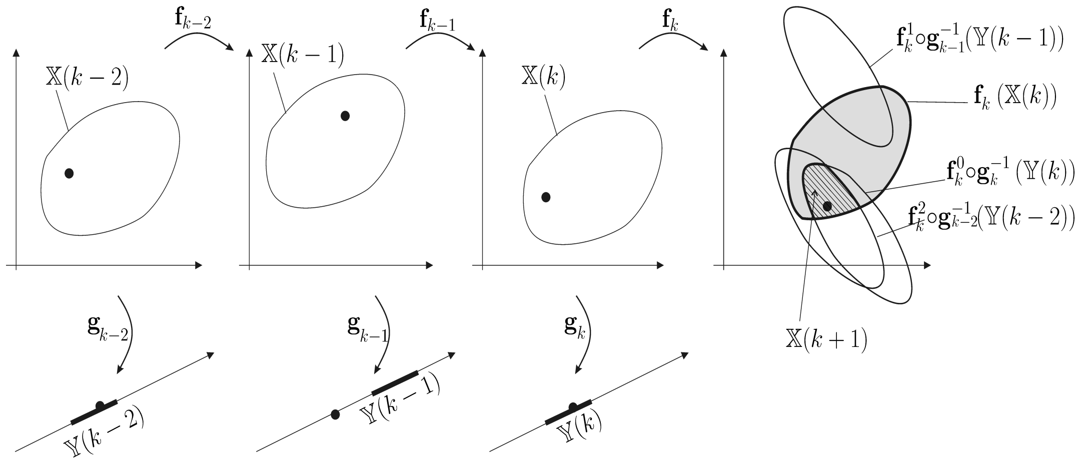
Define

$$\begin{cases} \mathbf{f}_{k:k}(\mathbb{X}) & \stackrel{\text{def}}{=} \mathbb{X} \\ \mathbf{f}_{k_1:k_2+1}(\mathbb{X}) & \stackrel{\text{def}}{=} \mathbf{f}_{k_2}(\mathbf{f}_{k_1:k_2}(\mathbb{X}), \mathbb{N}(k_2)), \ k_1 \leq k_2. \end{cases}$$

The set $\mathbf{f}_{k_1:k_2}(\mathbb{X})$ represents the set of all $\mathbf{x}(k_2)$, consistent with $\mathbf{x}(k_1) \in \mathbb{X}$.

Consider the set state estimator

$$\left\{ \begin{array}{lcl} \mathbb{X}(k) & = & \mathbf{f}_{0:k}\left(\mathbb{X}(0)\right) \quad \text{if } k < m, \ \text{(initialization step)} \\ \mathbb{X}(k) & = & \mathbf{f}_{k-m:k}\left(\mathbb{X}(k-m)\right) \cap \\ & & \bigcap_{i \in \{1,\dots,m\}}^{\{q\}} \mathbf{f}_{k-i:k} \circ \mathbf{g}_{k-i}^{-1}\left(\mathbb{Y}(k-i)\right) \quad \text{if } k \geq m \end{array} \right.$$



$\mathbf{f}_{k-2}$
$\mathbf{f}_{k-1}$
$\mathbf{f}_k$
$\mathbb{X}(k-2)$
$\mathbb{X}(k-1)$
$\mathbb{X}(k)$
$\mathbf{f}_k^1 \circ \mathbf{g}_{k-1}^{-1}(\mathbb{Y}(k-1))$
$\mathbf{f}_k(\mathbb{X}(k))$
$\mathbf{f}_k^0 \circ \mathbf{g}_k^{-1}(\mathbb{Y}(k))$
$\mathbf{f}_k^2 \circ \mathbf{g}_{k-2}^{-1}(\mathbb{Y}(k-2))$
$\mathbb{X}(k+1)$
$\mathbf{g}_{k-2}$
$\mathbf{g}_{k-1}$
$\mathbf{g}_k$
$\mathbb{Y}(k-2)$
$\mathbb{Y}(k-1)$
$\mathbb{Y}(k)$

We assume that all errors are time independent.

If (i) within any time window of length m we have less than q outliers and that (ii) $\mathbb{X}(0)$ contains $\mathbf{x}(0)$, then $\mathbb{X}(k)$ encloses $\mathbf{x}(k)$.

What is the probability of this assumption ?

Theorem. Consider the sequence of sets $\mathbb{X}(0), \mathbb{X}(1), \ldots$ built by the set observer. We have

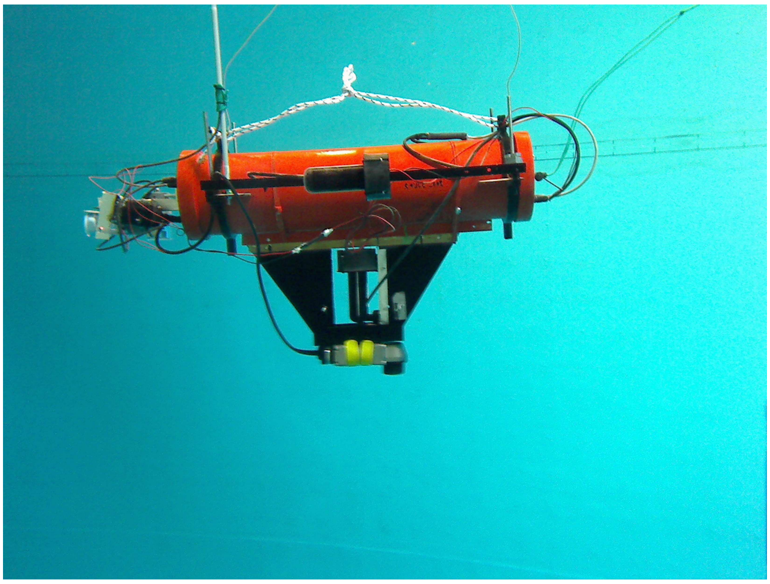
$$\Pr\left(\mathbf{x}\left(k\right) \in \mathbb{X}(k)\right) \geq \alpha \; * \; \Pr\left(\mathbf{x}\left(k-1\right) \in \mathbb{X}(k-1)\right)$$

where

$$\alpha = \sqrt[m]{\sum_{i=m-q}^{m} \frac{m!\ \pi_y^i . \left(1-\pi_y\right)^{m-i}}{i!\left(m-i\right)!}}$$

with an equality if $\mathbb{N}\left(k\right)$ are singletons.

3.3 Application to localization



Sauc'isse robot inside a swimming pool

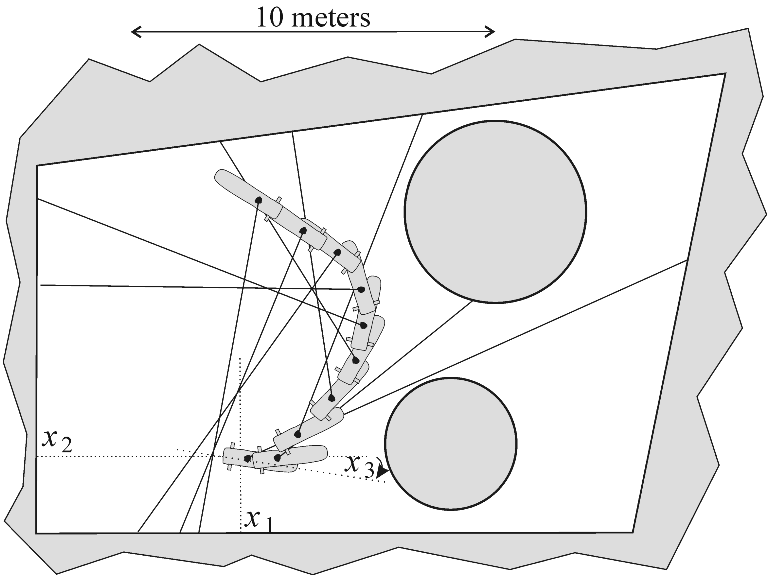
The robot evolution is described by

$$\begin{cases} \dot{x}_1 &= x_4 \cos x_3 \\ \dot{x}_2 &= x_4 \sin x_3 \\ \dot{x}_3 &= u_2 - u_1 \\ \dot{x}_4 &= u_1 + u_2 - x_4, \end{cases}$$

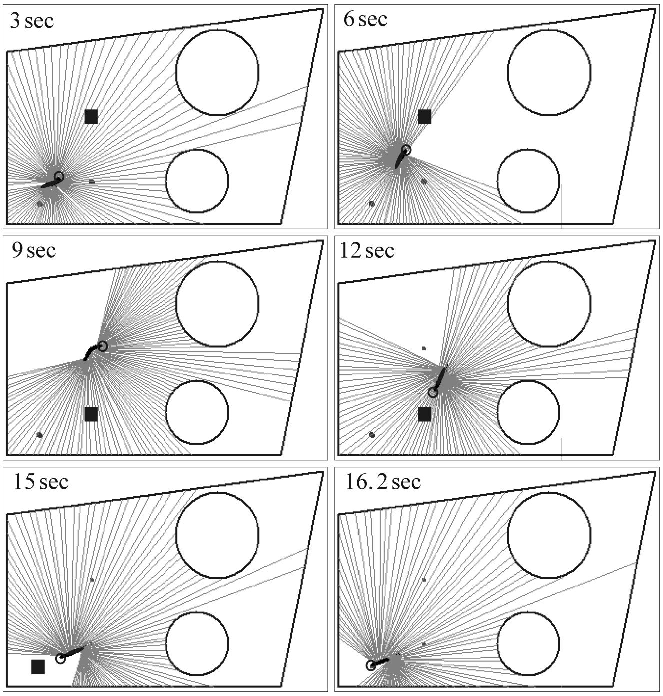
where x_1, x_2 are the coordinates of the robot center, x_3 is its orientation and x_4 is its speed. The inputs u_1 and u_2 are the accelerations provided by the propellers.

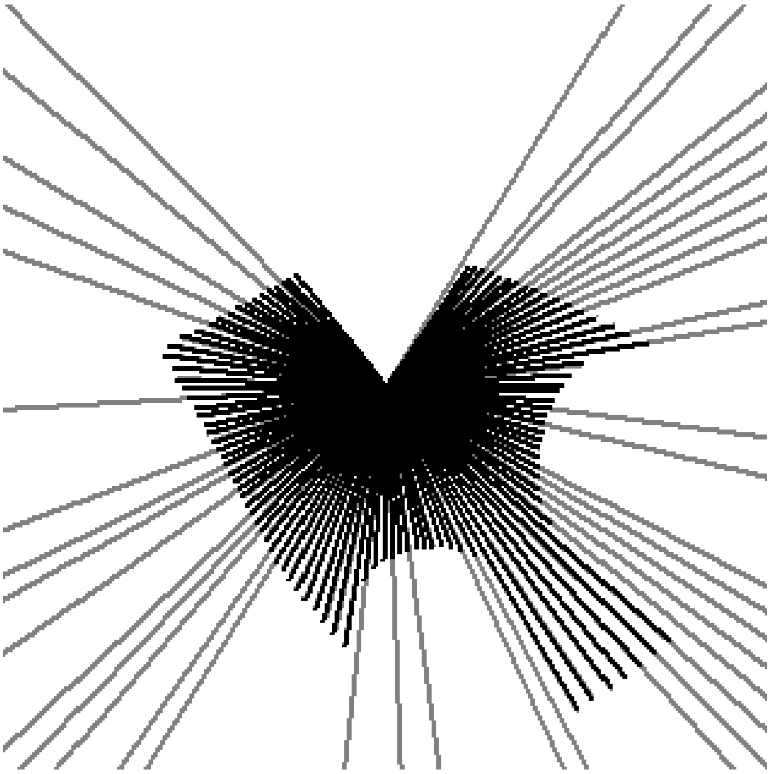
The system can be discretized by $\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k)$, where,

$$\mathbf{f}_k \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} x_1 + \delta.x_4.\cos(x_3) \\ x_2 + \delta.x_4.\sin(x_3) \\ x_3 + \delta.(u_2(k) - u_1(k)) \\ x_4 + \delta.(u_1(k) + u_2(k) - x_4) \end{pmatrix}$$

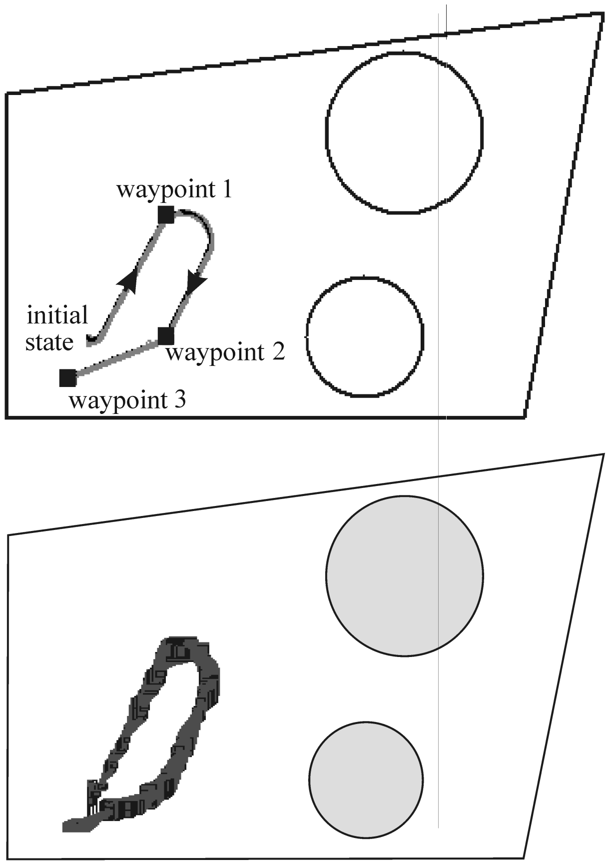


Underwater robot moving inside a pool


3 sec
6 sec
9 sec
12 sec
15 sec
16. 2 sec



Emission diagram at time $t = 9\,\mathsf{sec}$

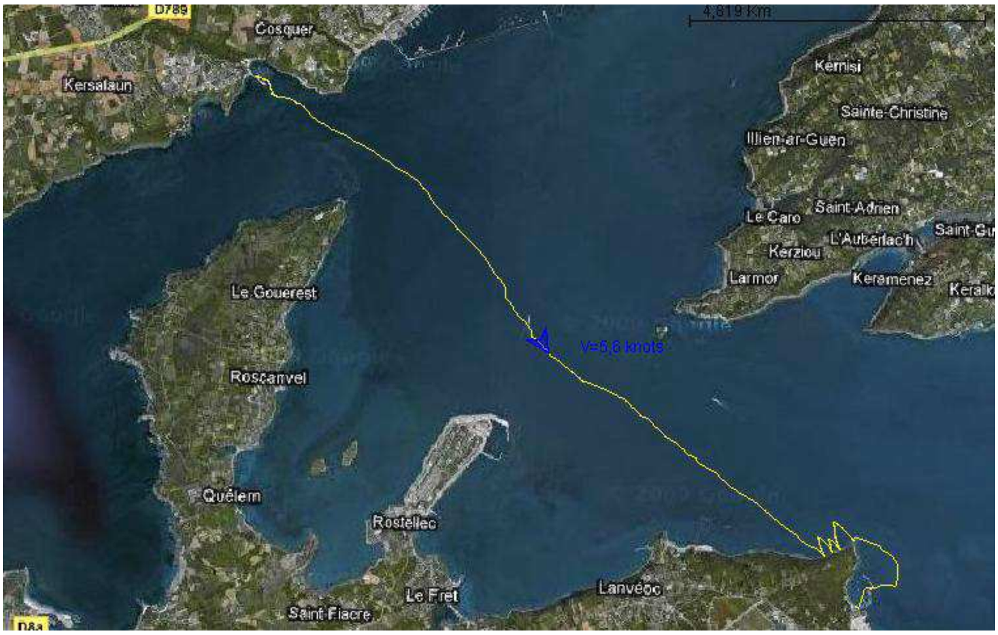

waypoint 1
initial state
waypoint 2
waypoint 3

t(sec)	$\mathsf{Pr}\left(\mathbf{x} \in \mathbb{X}\right)$	Outliers
3.0	$\geq$ 0.965	58
6.0	$\geq$ 0.932	50
9.0	$\geq$ 0.899	42
12.0	$\geq$ 0.869	51
15.0	$\geq$ 0.838	51
16.2	$\geq$ 0.827	49

Montrer la simu et la vidéo du concours

4 Breizh Spirit


BREIZH
ENSIETA
SPIRIT



D789
Cosquer
Kersalaun
4,819 Km
Kernisi
Sainte-Christine
Illien-ar-Guen
Saint-Adrien
Le Caro
L'Auberlac'h
Kerziou
Larmor
Keramenez
Le Gouerest
V=5,6 knots
Roscanvel
Quélem
Rostellec
Le Fret
Lanvéoc
Saint-Fiacre
D8a

Voilier autonome. La rade avant la transatlantique

Avant le grand bain, il y a le petit. Le voilier miniature autonome concocté à l'Ensieta a traversé avec succès la rade, en début de semaine. L'idée : réussir un jour une transatlantique.

Une partie de l'équipe : Kostia Poncin, Richard Leloup, Luc Jaulin, Bruno Auzier et Jan Sliwka. Manque Pierre-Henri Reilhac.

Lundi, Breizh-Spirit – c'est son nom – est parti de Saint-Anne-du-Portzic et a rejoint Lanvéoc, soit 12 km en deux heures environ. Il était tout seul, autonome, accompagné à distance, sur un semi-rigide, de ses « parents », une petite équipe d'étudiants et d'enseignants de l'Ensieta. Une traversée réalisée en collaboration avec l'École navale.

De beaucoup, Breizh-Spirit est sans doute resté inaperçu. Il ne fait qu'1,30 m de long pour 10 kg. Mais il a avancé vaillamment, à 3,1 nœuds de moyenne, au près, ce qui n'était pas la configuration idéale. En pointe, il a atteint 5,5 nœuds.

Premier test à la mer près de Porto

L'idée a pris corps en 2005. Luc Jaulin, professeur en automatique-robotique à l'Ensieta, était alors président du jury, à Toulouse, de la première Microtransat. L'objectif, pour une traversée de l'Atlantique, a été fixé à 2010.

Breiz-Spirit a lui-même mûri l'année passée. Richard Leloup, alors en première année, se souvient avoir fabriqué la coque durant les vacances de Noël. D'autres ont apporté leur pierre en électronique, informatique, mécanique, robotique et architecture navale, des compétences qui existent à l'école et que des projets, tels que Breiz-Spirit, permettent de mixer autour d'un objectif à atteindre.

Cet été, le mini-voilier a participé, près de Porto, à la « World robotic sailing championship », premier test à la mer pour lui ; l'occasion aussi de se comparer. Onze bateaux, fort divers, étaient au rendez-vous. Il y avait là aussi des Anglais, des Suisses, des Portugais et des Américains.

Une compétition en septembre 2010

L'équipe de l'Ensieta a en ligne de mire 2010 avec une compétition, en juin, probablement au Canada. Le départ de la fameuse transatlantique pourrait avoir lieu, en septembre, depuis l'Irlande. La traversée risque alors de durer cinq mois...

Pour l'heure, l'équipe de Breizh-Spirit va travailler à améliorer le mini-voilier, rendre plus robuste l'électronique, le gréement et les voiles. Étanchéifier la coque, implanter des panneaux solaires, se passer de la girouette sont aussi au programme. Il est prévu que les bateaux puissent communiquer chaque jour leur position à terre. Normalement, aucun voilier de cette future transat en autonomie ne doit dépasser les 4 m, des « Petits Poucet » comparés aux porte-conteneurs géants...

Montrer une vidéo

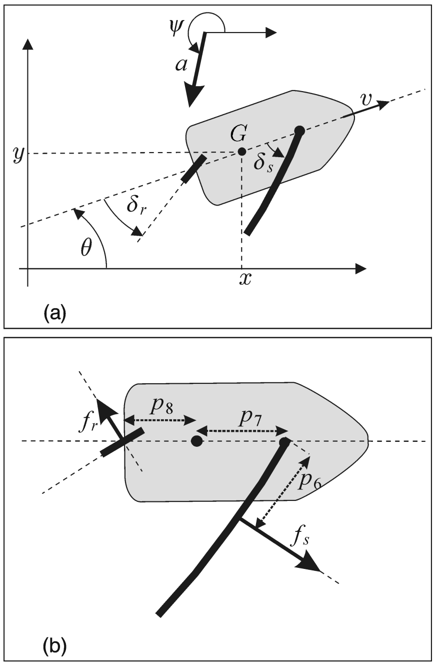
4.1 Sensors

- *Reliable sensors*: GPS, compass, gyrometers and accelerometers (low energy consumers, can be enclosed inside a waterproof tank, can survive for years).

- *Unreliable sensors:* Anemometers, weather vane, dynamometers (they are directly in contact with wind, wave, salt, . . .) and can fail down at any time.

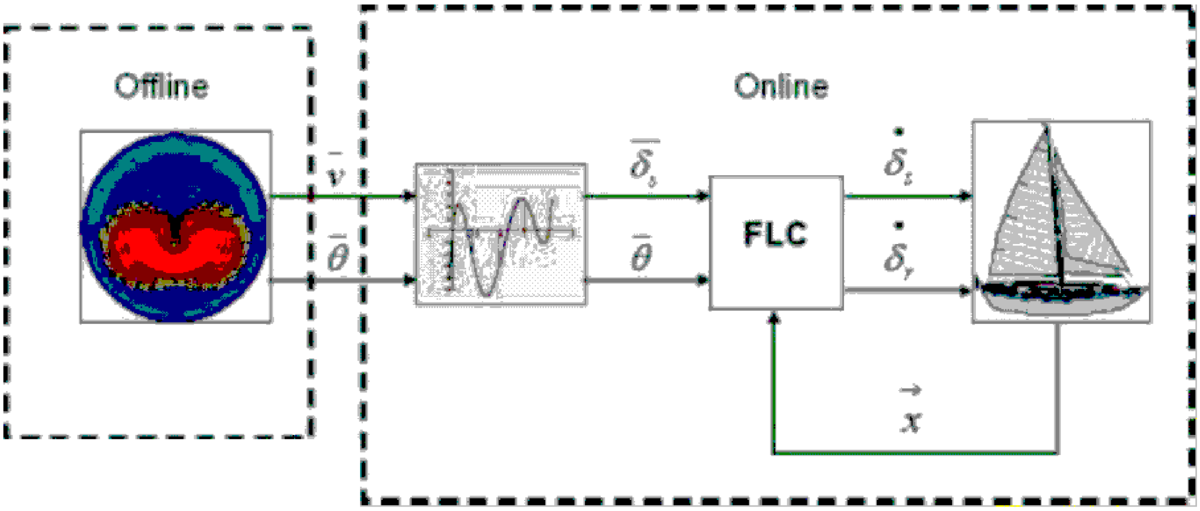
4.2 Normalized State equations

$$
\left\{
\begin{array}{rcl}
\dot{x} & = & v \cos\theta + a \cos\psi \\
\dot{y} & = & v \sin\theta + a \sin\psi \\
\dot{\theta} & = & \omega \\
\dot{v} & = & f_s . \sin\delta_s - f_r . \sin u_1 - v \\
\dot{\omega} & = & f_s . \left(1 - \cos\delta_s\right) - f_r . \cos u_1 - \omega \\
\dot{a} & = & 0 \\
\dot{\psi} & = & 0 \\
f_s & = & a \sin\left(\theta - \psi + \delta_s\right) \\
f_r & = & v \sin u_1 \\
\gamma & = & \cos\left(\theta - \psi\right) + \cos\left(u_2\right) \\
\delta_s & = & \left\{
\begin{array}{cc}
\pi - \theta + \psi & \text{if } \gamma \leq 0 \\
sign\left(\sin\left(\theta - \psi\right)\right) . u_2 & \text{otherwise.}
\end{array}
\right.
\end{array}
\right.
$$



ψ
a
v
G
y
δ_s
δ_r
θ
x
(a)
p_8
p_7
f_r
p_6
f_s
(b)

4.3 Control



4.4 Observer

To control the boat, we need to know where the wind comes from and what is its speed.

If the system

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

is *flat* with the flat output $\mathbf{y}$, then there exist two functions $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ such that for all t, we have

$$\begin{cases} \mathbf{x} &= \boldsymbol{\phi}\left(\mathbf{y},\dot{\mathbf{y}},\dots,\mathbf{y}^{(r-1)}\right) \\ \mathbf{u} &= \boldsymbol{\psi}\left(\mathbf{y},\dot{\mathbf{y}},\dots,\mathbf{y}^{(r-1)},\mathbf{y}^{(r)}\right). \end{cases}$$

To get $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$, we have to proceed as follows.

- The *derivation step* computes symbolically $\mathbf{y}, \dot{\mathbf{y}}, \ldots, \mathbf{y}^{(r)}$ with respect to $\mathbf{x}$ and $\mathbf{u}$. We get

$$\begin{pmatrix} \mathbf{y} \\ \dot{\mathbf{y}} \\ \vdots \\ \mathbf{y}^{(r)} \end{pmatrix} = \mathbf{h}\begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix}.$$

- The *resolution step* inverses symbolically the function $\mathbf{h}$. This operation is not easy.

Example. Consider the system

$$\begin{cases} \dot{x}_1 &=& x_1 + x_2 \\ \dot{x}_2 &=& x_2^2 + u \\ y &=& x_1. \end{cases}$$

Derivation step:

$$\begin{cases} y &=& x_1 \\ \dot{y} &=& \dot{x}_1 = x_1 + x_2 \\ \ddot{y} &=& \dot{x}_1 + \dot{x}_2 = x_1 + x_2 + x_2^2 + u. \end{cases}$$

Thus

$$\begin{pmatrix} y \\ \dot{y} \\ \ddot{y} \end{pmatrix} = \underbrace{\begin{pmatrix} x_1 \\ x_1 + x_2 \\ x_1 + x_2 + x_2^2 + u \end{pmatrix}}_{\mathbf{h}(\mathbf{x},u)}.$$

Resolution step:

$$\left\{\begin{array}{lll} x_1 & = & y \\ x_2 & = & \dot{y} - x_1 = \dot{y} - y \\ u & = & \ddot{y} - \left(x_1 + x_2 + x_2^2\right) = \ddot{y} - \dot{y} - (\dot{y} - y)^2 \,. \end{array}\right.$$

i.e.

$$\left(\begin{array}{c} \mathbf{x} \\ u \end{array}\right) = \underbrace{\left(\begin{array}{c} y \\ \dot{y} - y \\ \ddot{y} - \dot{y} - (\dot{y} - y)^2 \end{array}\right)}_{\mathbf{h}^{-1}(y,\dot{y},\ddot{y})}$$

As a consequence,

$$\begin{cases} \boldsymbol{\phi}\left(y,\dot{y}\right) & = & \begin{pmatrix} y \\ \dot{y}-y \end{pmatrix} \\ \psi\left(y,\dot{y},\ddot{y}\right) & = & \ddot{y}-\dot{y}-\left(\dot{y}-y\right)^{2}. \end{cases}$$

4.5 New approach

$$\underbrace{\begin{pmatrix} \mathbf{y} \\ \dot{\mathbf{y}} \\ \vdots \\ \mathbf{y}^{(r)} \end{pmatrix}}_{\mathbf{z}} = \mathbf{h}\underbrace{\begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix}}_{\mathbf{w}}$$

Classical approach. We invert symbolically $\mathbf{h}$ and then we compute $\mathbf{h}^{-1}(\hat{\mathbf{z}})$, where $\hat{\mathbf{z}}$ is a measure of $\mathbf{z}$.

Our approach: We compute $\mathbb{W} = [\mathbf{w}] \cap \mathbf{h}^{-1}([\mathbf{z}])$ for each t.

From the state equations of the sailboat, we get

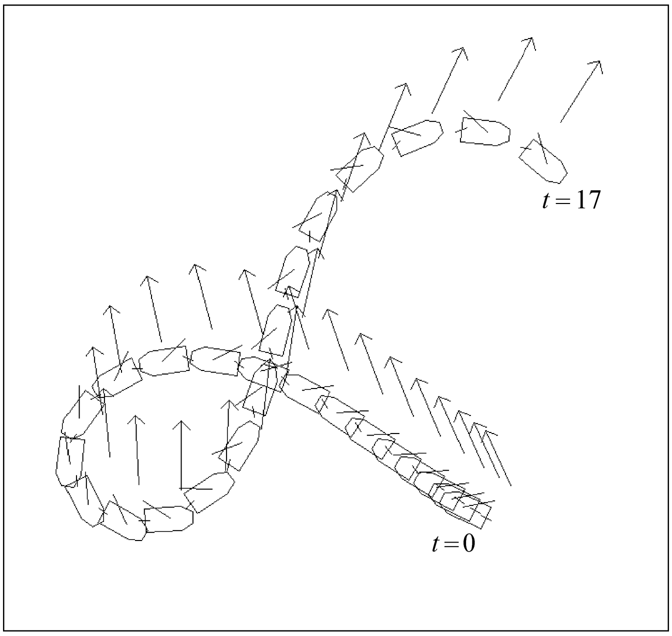
$$\underbrace{\begin{pmatrix} \theta \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{pmatrix}}_{\mathbf{z}} = \underbrace{\begin{pmatrix} \theta \\ v\sin\theta + a\sin\psi \\ v\sin\theta + a\sin\psi \\ \omega \\ (f_s.\sin\delta_s - f_r\sin u_1 - v)\cos\theta - \omega v\sin\theta \\ (f_s\sin\delta_s - f_r\sin u_1 - v)\sin\theta + \omega v\cos\theta \\ f_s(1-\cos\delta_s) - v\sin u_1\cos u_1 - \omega \end{pmatrix}}_{\mathbf{h}(\mathbf{w})}$$

with

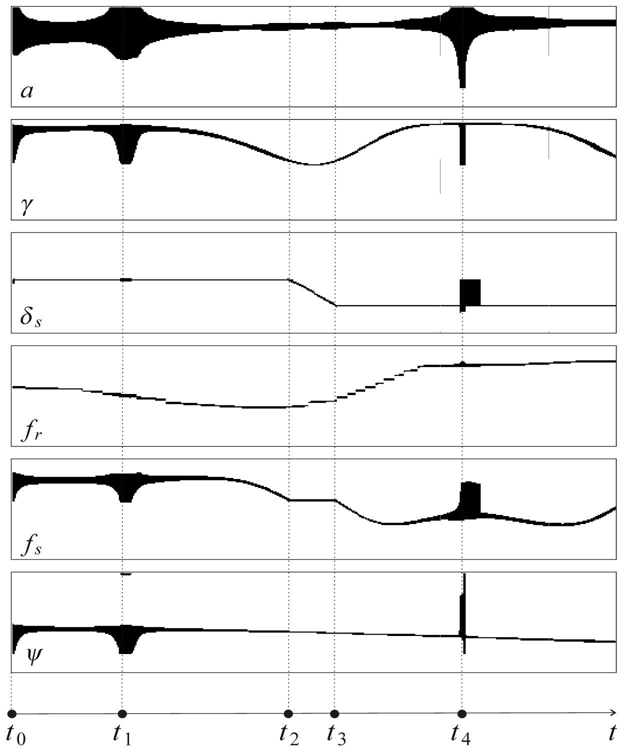
$$\mathbf{w} = \begin{pmatrix} \theta & v & \omega & a & \psi & u_1 & u_2 \end{pmatrix}^{\mathsf{T}}$$

and

$$\begin{cases} f_s(\mathbf{w}) &= a\sin(\theta - \psi + \delta_s) \\ f_r(\mathbf{w}) &= v\sin u_1 \\ \delta_s(\mathbf{w}) &= \begin{cases} \pi - \theta + \psi & \text{if } \gamma(\mathbf{x},t) \leq 0 \\ sign(\sin(\theta-\psi)).u_2 & \text{otherwise} \end{cases} \\ \gamma(\mathbf{w}) &= \cos(\theta-\psi) + \cos(u_2). \end{cases}$$



Simulated experiment



a
γ
δ_s
f_r
f_s
ψ
t_0
t_1
t_2
t_3
t_4
t