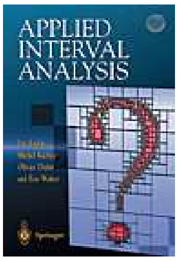
Interval constraint propagation; applications to robotics



Luc Jaulin,
http://www.ensieta.fr/jaulin/
ENSIETA, Brest, France

DyToComp'09, Poznan, June 4, 2009.

1 Contractors

Interval bisection algorithms bisect all boxes in all directions and become inefficient. Interval methods can still be useful if

- the solution set $\mathbb{X}$ is small (optimization problem, solving equations),
- contraction procedures are used as much as possible,
- bisections are used only as a last resort.

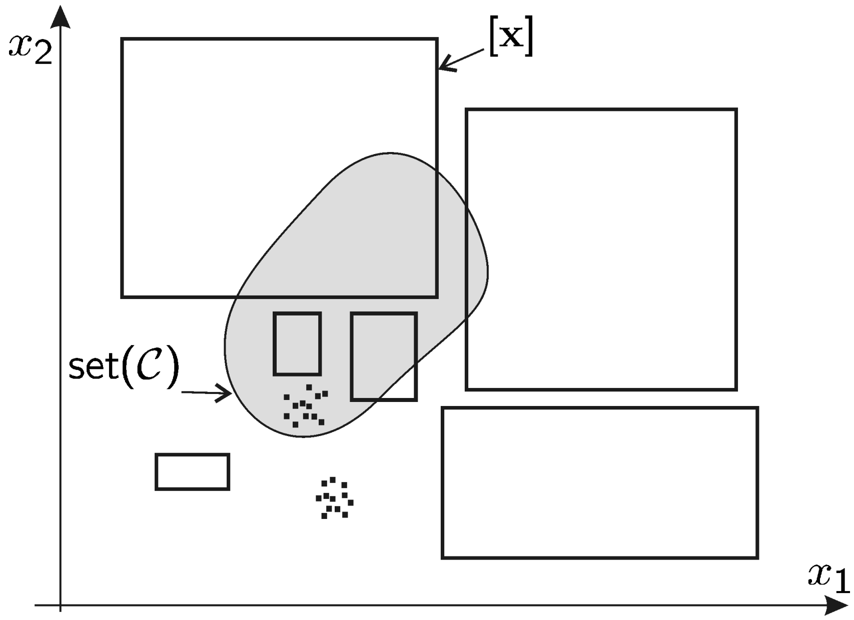
1.1 Definition

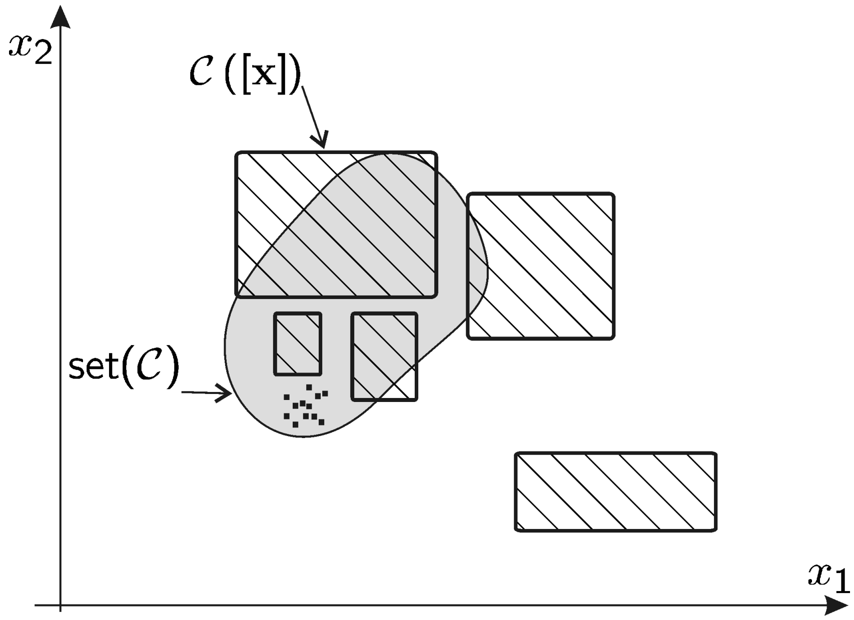
The operator $\mathcal{C} : \mathbb{IR}^n \to \mathbb{IR}^n$ is a *contractor* if

$$\left\{ \begin{array}{ll} \forall[\mathbf{x}] \in \mathbb{IR}^n,\ \mathcal{C}([\mathbf{x}]) \subset [\mathbf{x}] & \text{(contractance)}, \\ (\mathbf{x} \in [\mathbf{x}], \mathcal{C}(\{\mathbf{x}\}) = \{\mathbf{x}\}) \Rightarrow \mathbf{x} \in \mathcal{C}([\mathbf{x}]) & \text{(completeness)}. \end{array} \right.$$

We define

$$\operatorname{set}(\mathcal{C}) \overset{\text{def}}{=} \{\mathbf{x} \in \mathbb{R}^n, \mathcal{C}(\{\mathbf{x}\}) = \{\mathbf{x}\}\}.$$


x_2
$[\mathbf{x}]$
$\mathsf{set}(\mathcal{C})$
x_1


x_2
$\mathcal{C}([\mathbf{x}])$
set($\mathcal{C}$)
x_1

$\mathcal{C}_{\mathbb{X}}$ is *monotonic* if	$[\mathbf{x}] \subset [\mathbf{y}] \Rightarrow \mathcal{C}_{\mathbb{X}}([\mathbf{x}]) \subset \mathcal{C}_{\mathbb{X}}([\mathbf{y}])$
$\mathcal{C}_{\mathbb{X}}$ is *minimal* if	$\mathcal{C}_{\mathbb{X}}([\mathbf{x}]) = [[\mathbf{x}] \cap \mathbb{X}]$
$\mathcal{C}_{\mathbb{X}}$ is *thin* if	$\mathcal{C}_{\mathbb{X}}(\{\mathbf{x}\}) = \{\mathbf{x}\} \cap \mathbb{X}$
$\mathcal{C}_{\mathbb{X}}$ is *idempotent* if	$\mathcal{C}_{\mathbb{X}}(\mathcal{C}_{\mathbb{X}}([\mathbf{x}])) = \mathcal{C}_{\mathbb{X}}([\mathbf{x}])$.

$\mathcal{C}_{\mathbb{X}}$ is said to be *convergent* if

$$[\mathbf{x}](k) \to \mathbf{x} \quad \Rightarrow \quad \mathcal{C}_{\mathbb{X}}([\mathbf{x}]\,(k)) \to \{\mathbf{x}\} \cap \mathbb{X}.$$

1.2 Projection of constraints

Let x, y, z be 3 variables such that

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

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

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

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

The contractor associated with $z = x + y$ is.

Algorithm pplus(inout: $[z], [x], [y]$)	
1	$[z] := [z] \cap ([x] + [y])$;
2	$[x] := [x] \cap ([z] - [y])$;
3	$[y] := [y] \cap ([z] - [x])$.

The projection procedure developed for plus can be extended to other ternary constraints such as mult: $z = x * y$, or equivalently

$$\text{mult} \triangleq \left\{ (x, y, z) \in \mathbb{R}^3 \mid z = x * y \right\}.$$

The resulting projection procedure becomes

Algorithm pmult(inout: $[z], [x], [y]$)
1 $[z] := [z] \cap ([x] * [y])\,;$ 2 $[x] := [x] \cap ([z] * 1/[y])\,;$ 3 $[y] := [y] \cap ([z] * 1/[x])\,.$

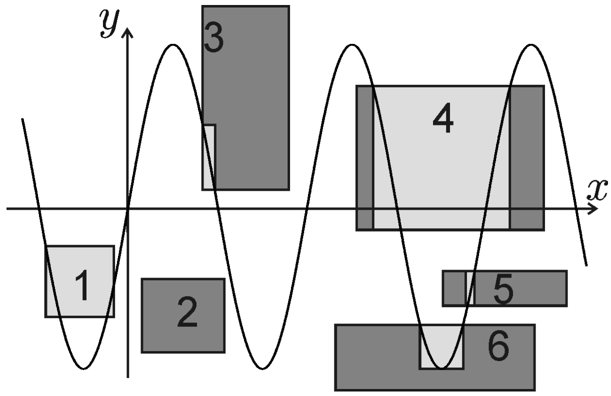
Consider the binary constraint

$$\mathsf{exp} \triangleq \{(x, y) \in \mathbb{R}^n | y = \mathsf{exp}\,(x)\}\,.$$

The associated contractor is

Algorithm pexp(inout: $[y], [x]$)
1 $\quad [y] := [y] \cap \mathsf{exp}\,([x])\,;$
2 $\quad [x] := [x] \cap \mathsf{log}\,([y])\,.$

Any constraint for which such a projection procedure is available will be called a *primitive constraint*.



Projection of the sine constraint

1.3 Constraint propagation

A CSP (Constraint Satisfaction Problem) is composed of

1) a set of variables $\mathcal{V} = \{x_1, \ldots, x_n\}$,

2) a set of constraints $\mathcal{C} = \{c_1, \ldots, c_m\}$ and

3) a set of interval domains $\{[x_1], \ldots, [x_n]\}$.

Principle of propagation techniques: contract $[\mathbf{x}] = [x_1] \times \cdots \times [x_n]$ as follows:

$$((((((([\mathbf{x}] \sqcap c_1) \sqcap c_2) \sqcap \ldots) \sqcap c_m) \sqcap c_1) \sqcap c_2) \ldots,$$

until a steady box is reached.

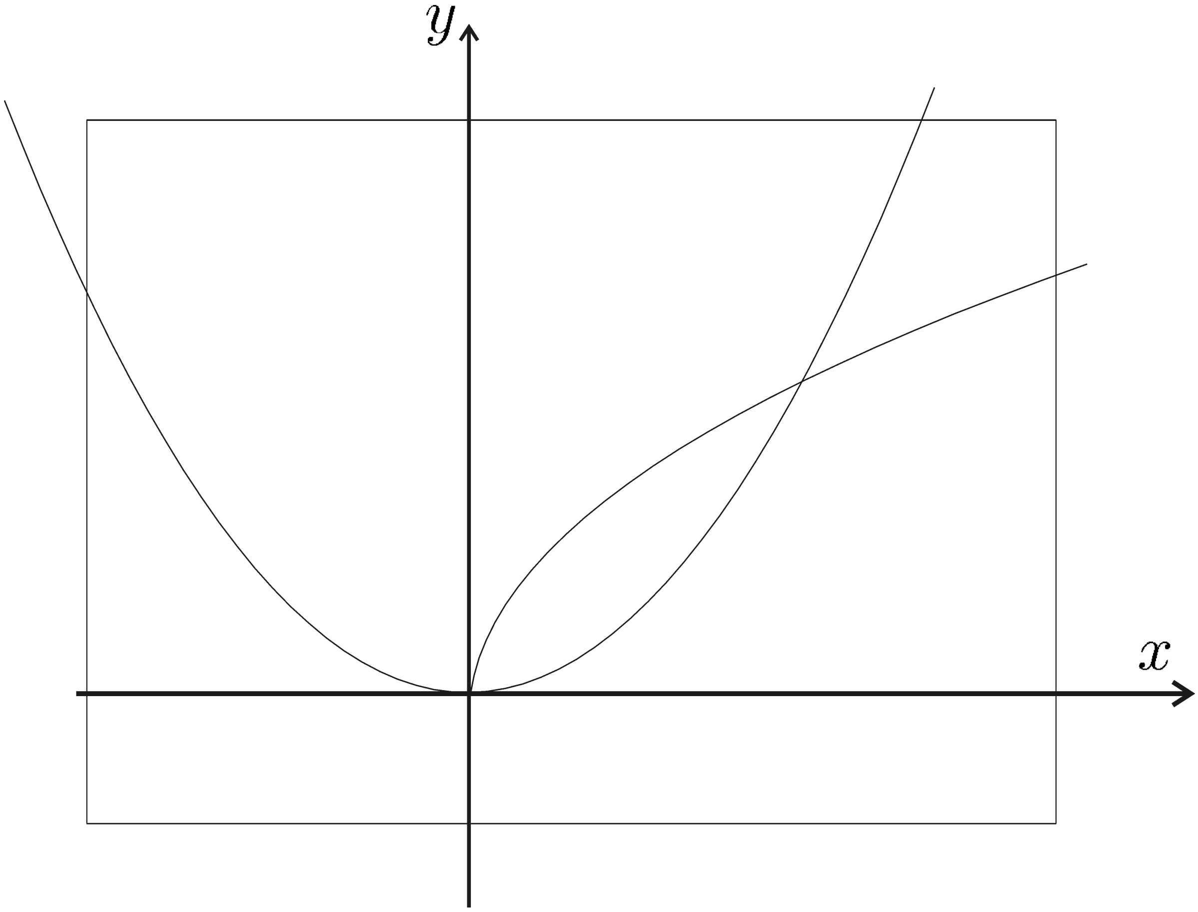
Example. Consider the system of two equations.

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

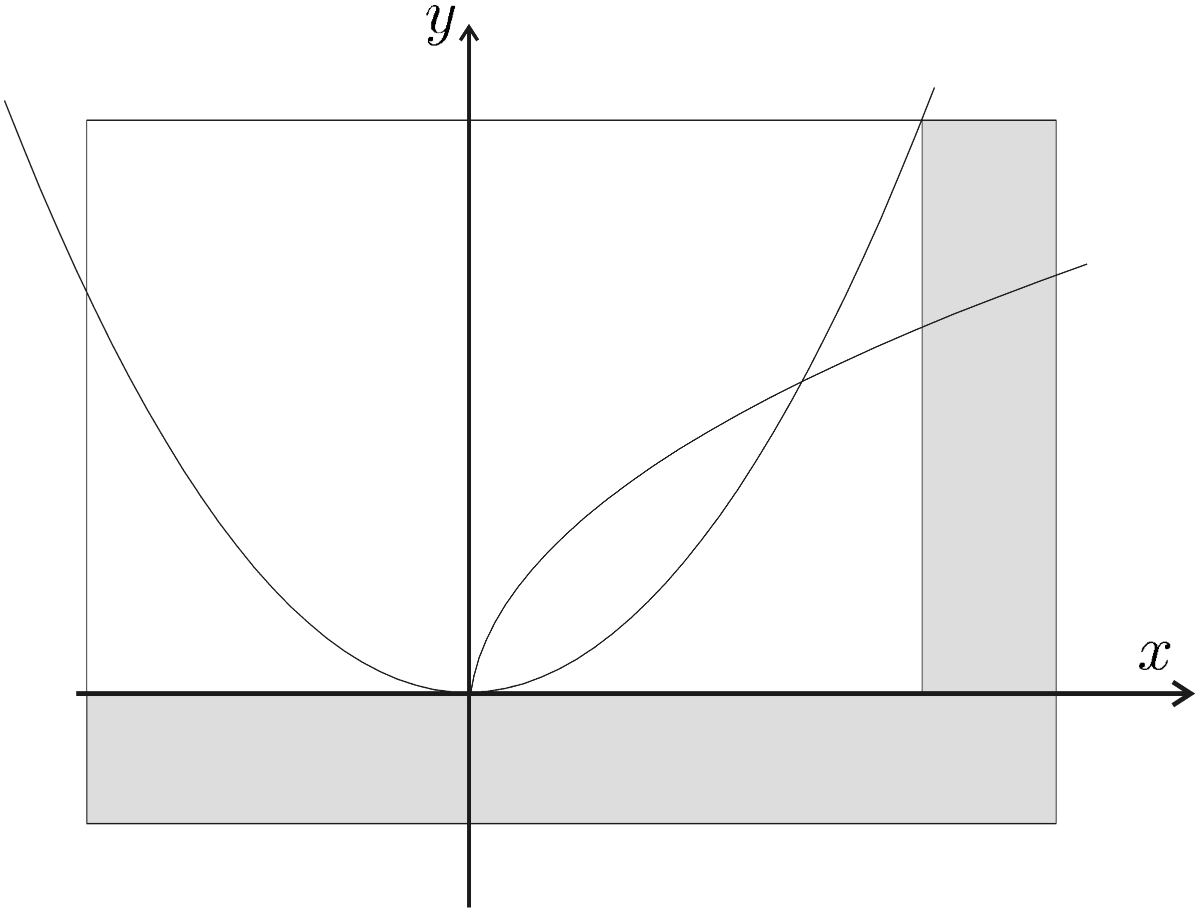
We can build two contractors

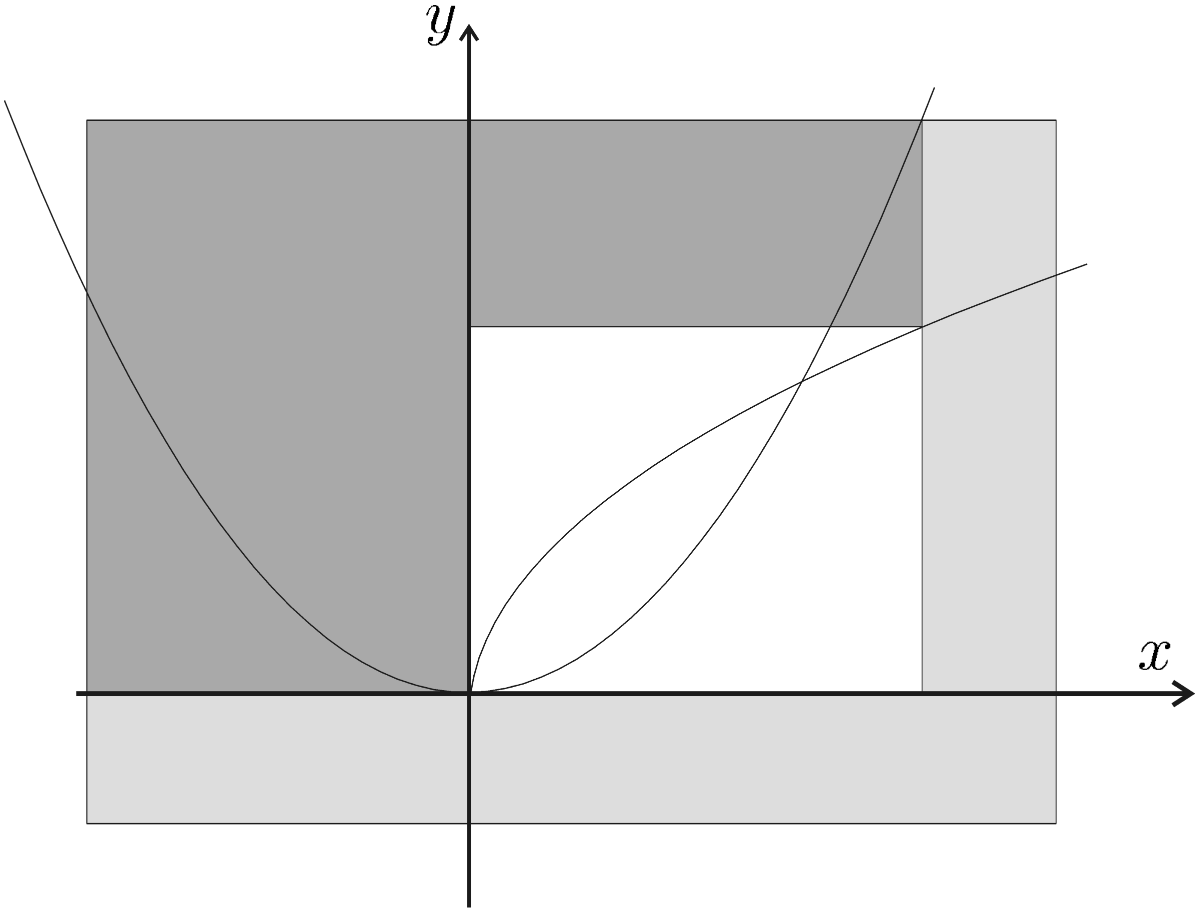
$$\mathcal{C}_1 : \begin{cases} [y] = [y] \cap [x]^2 \\ [x] = [x] \cap \sqrt{[y]} \end{cases} \quad \text{associated to } y = x^2$$

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

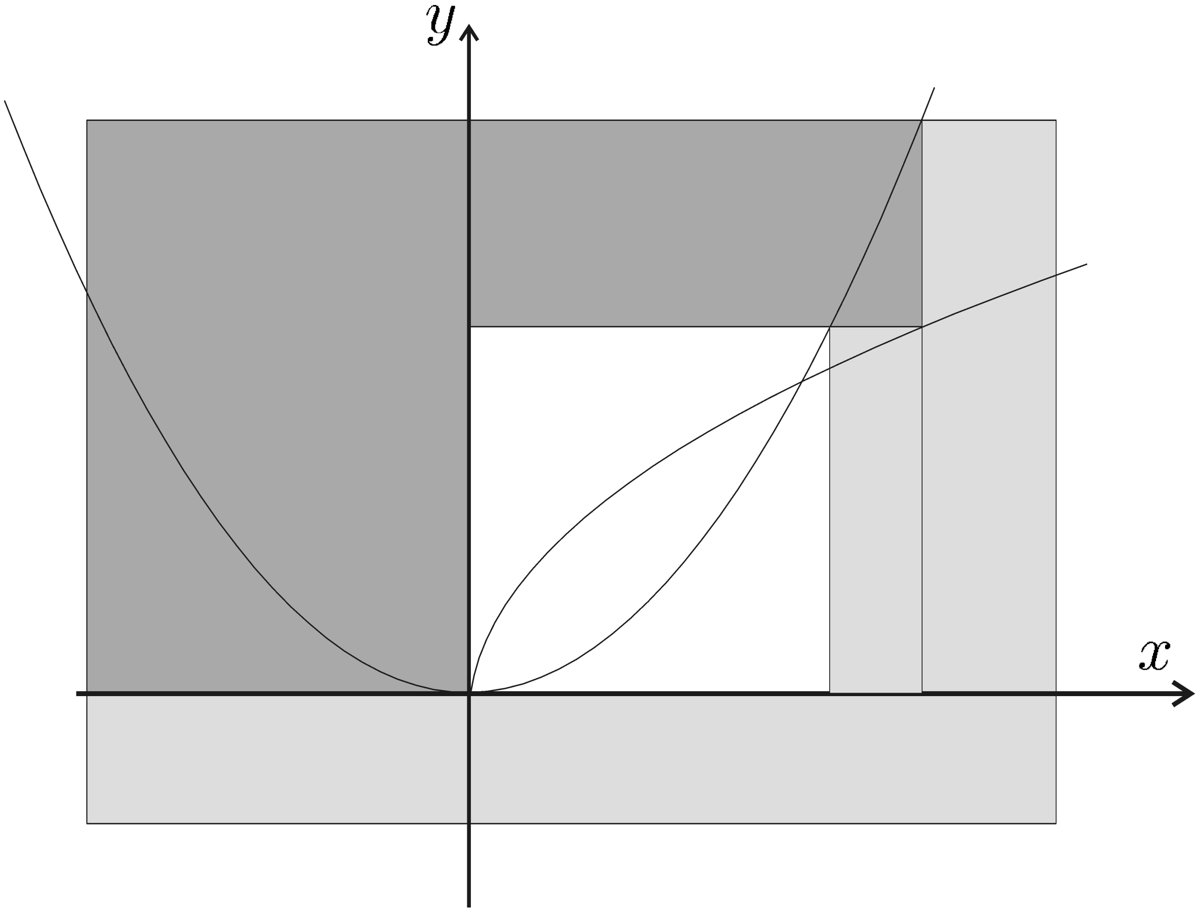


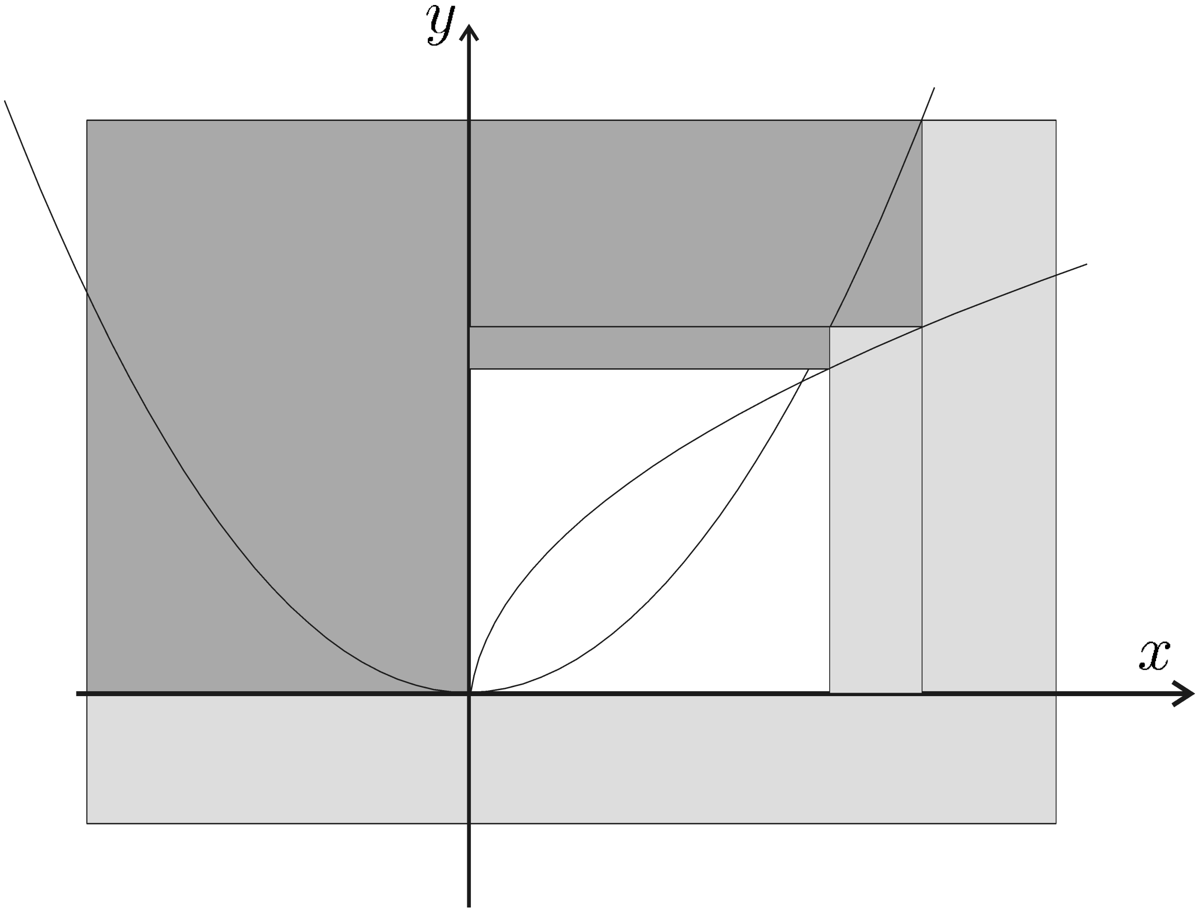
y
x


y
x

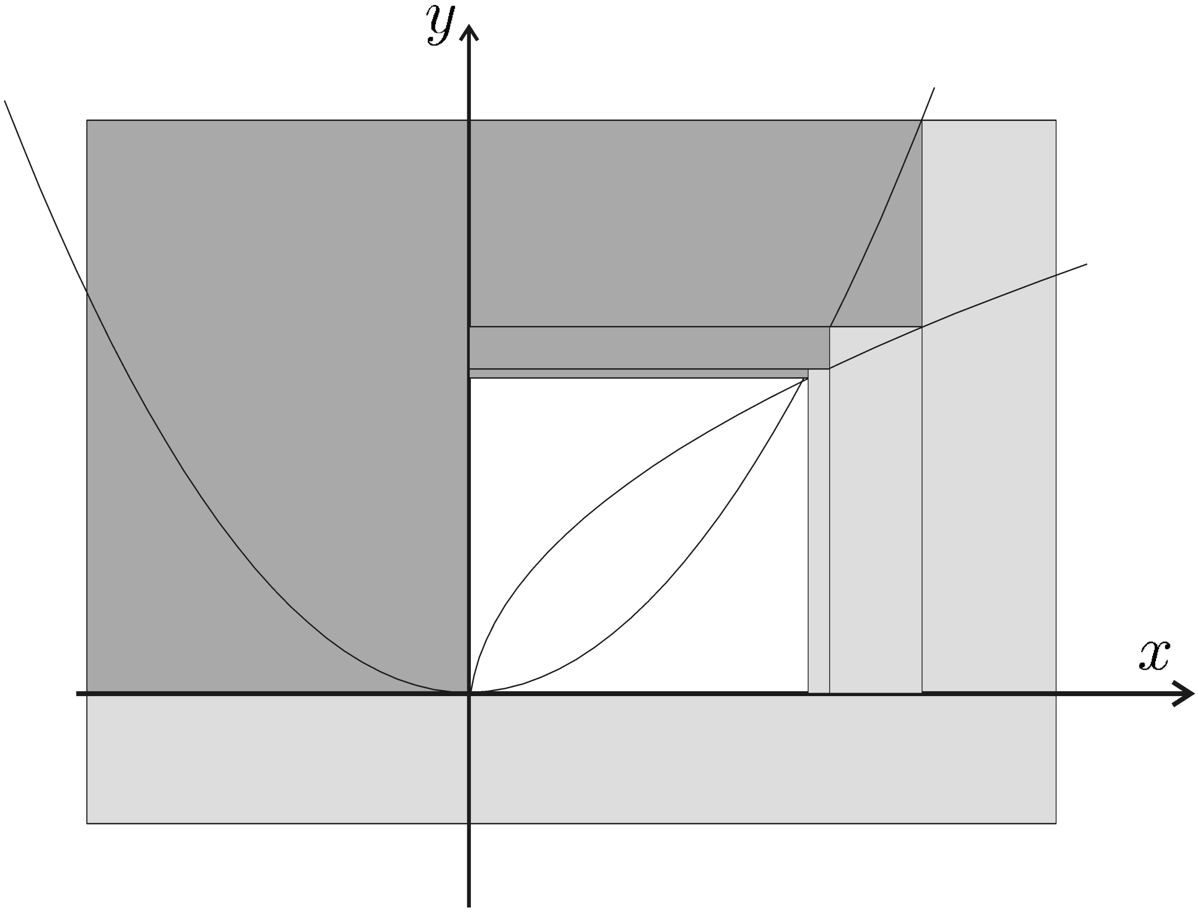


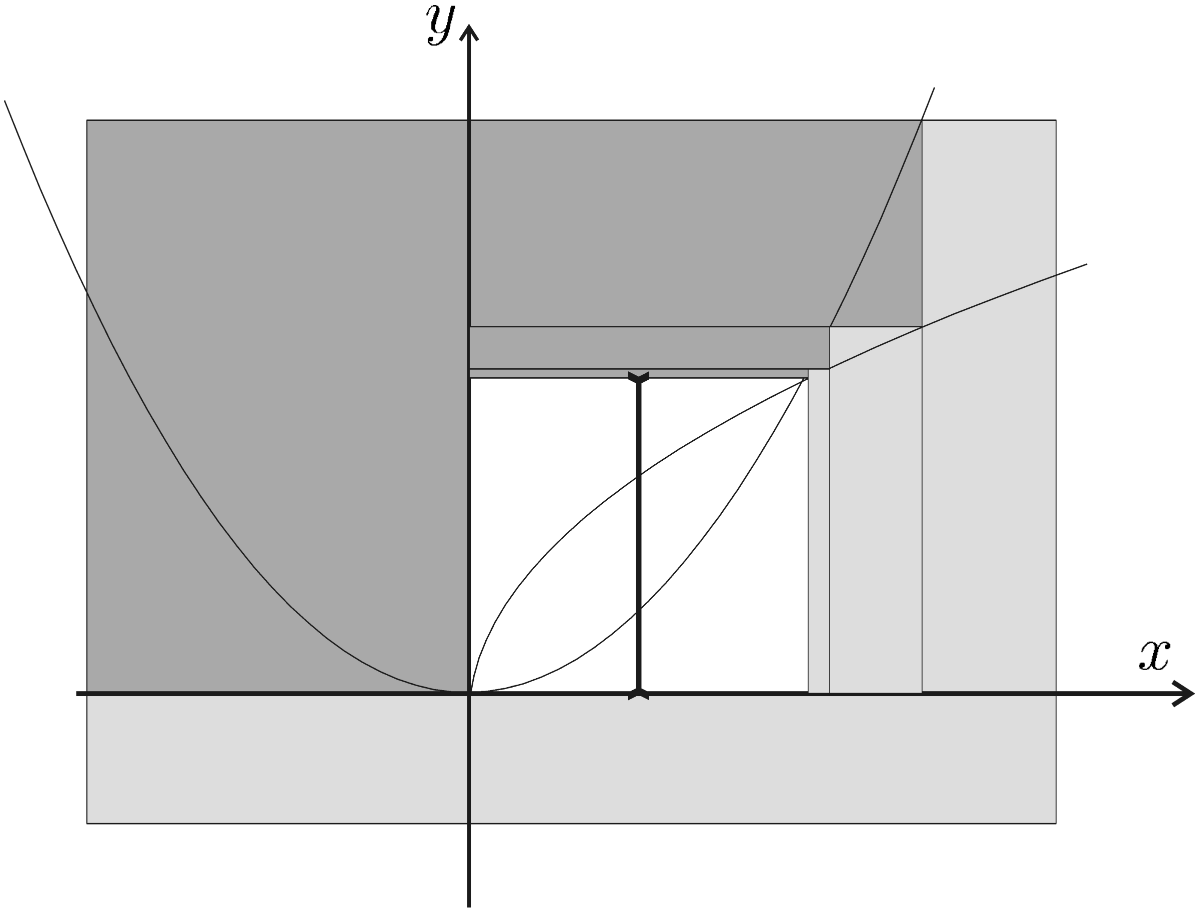
y
x


y
x

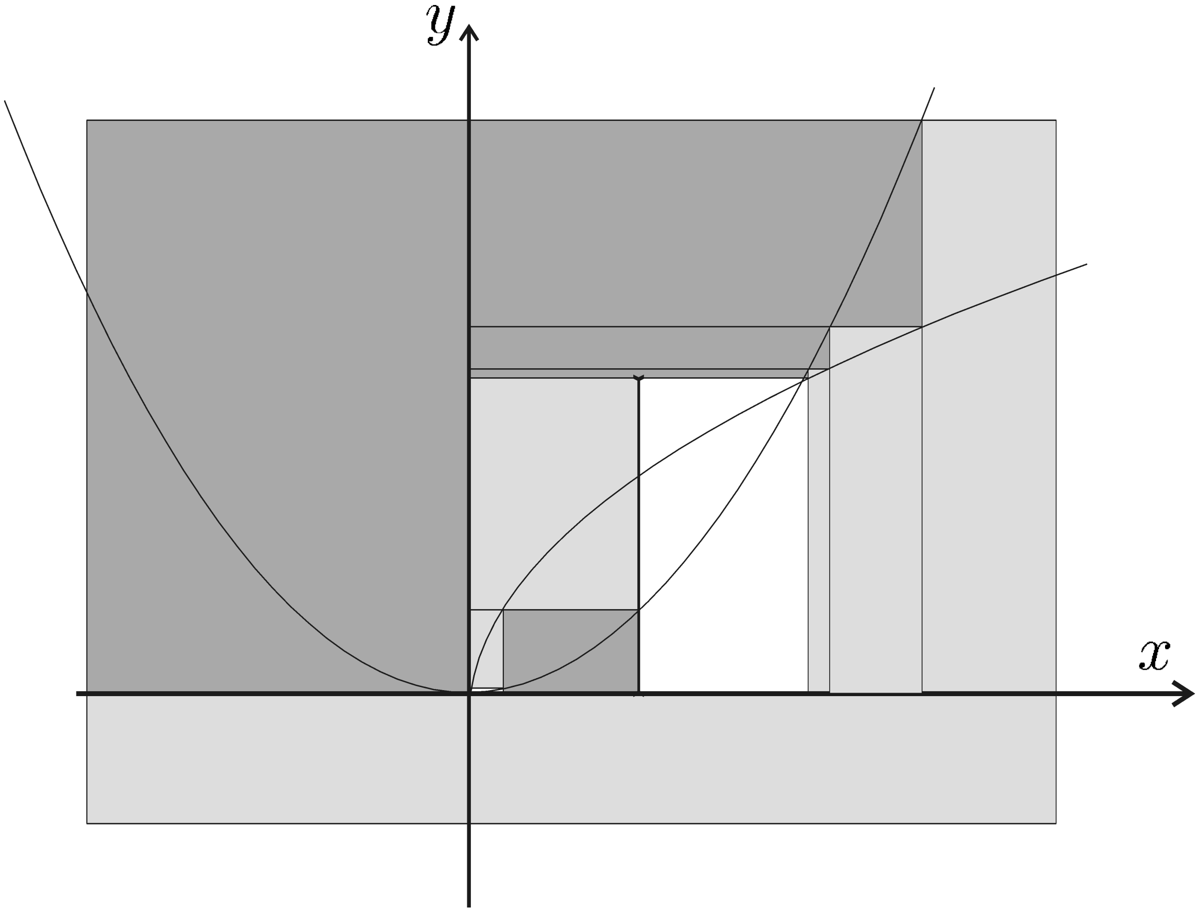


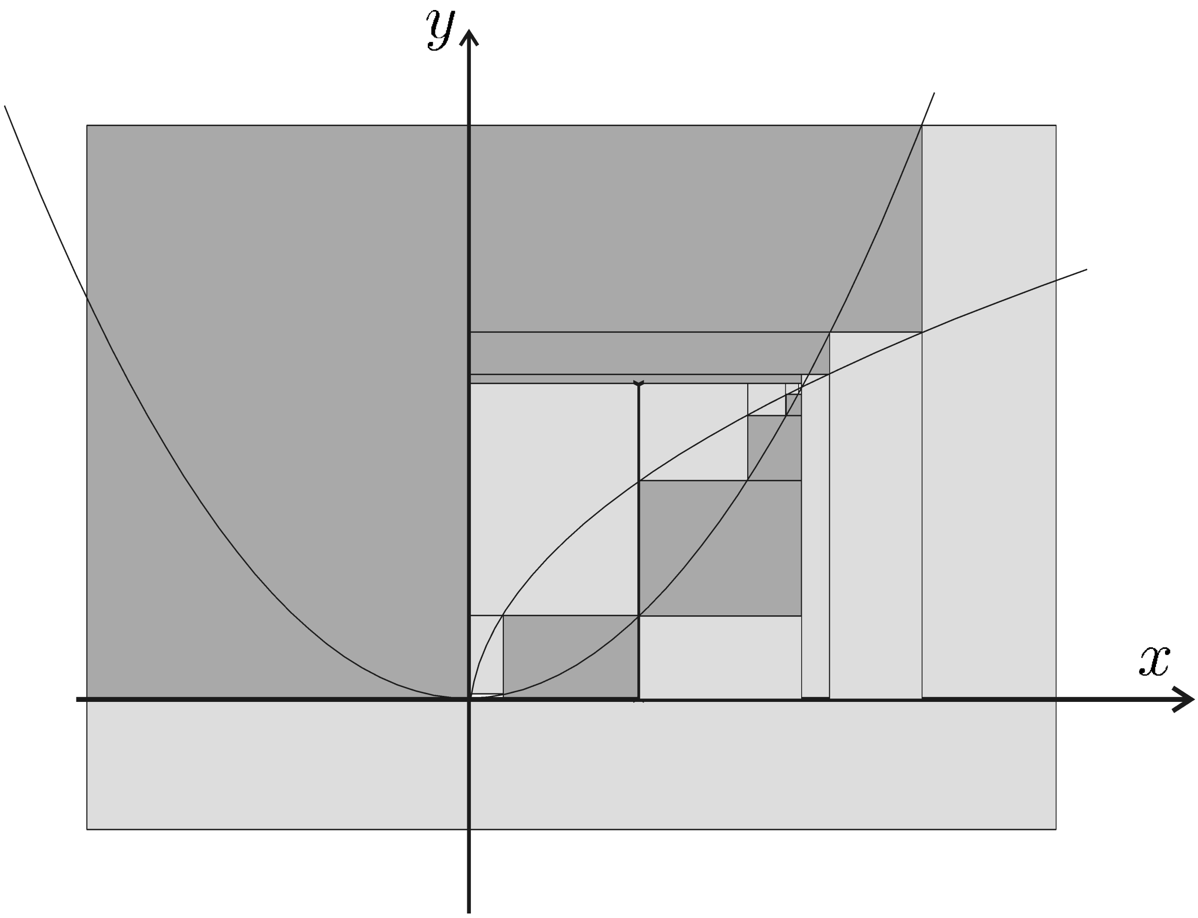
y
x


y
x



y
x


y
x



y
x

1.4 Decomposition into primitive constraints

$$
\begin{aligned}
&x + \mathsf{sin}(xy) \leq 0, \\
&x \in [-1, 1], y \in [-1, 1], z \in [-1, 1]
\end{aligned}
$$

can be decomposed into

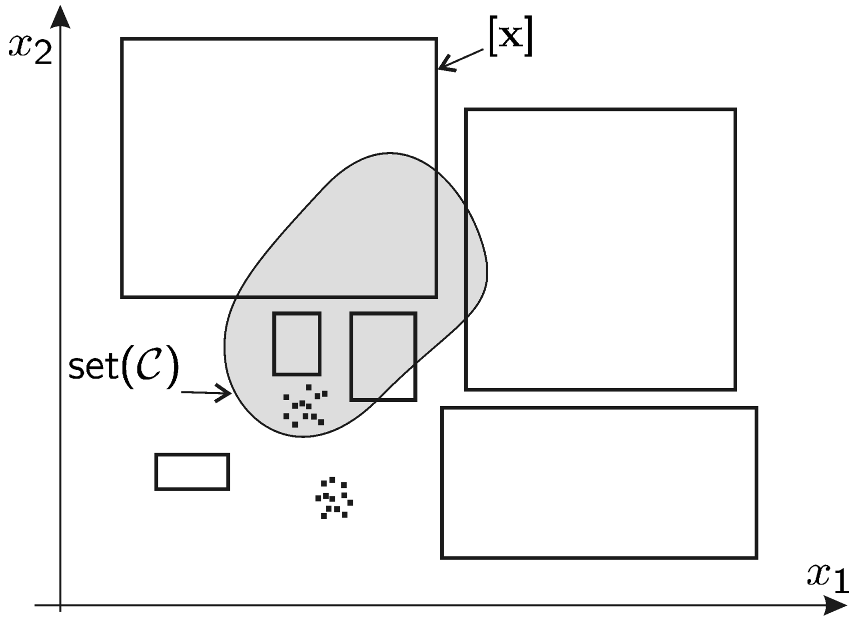
$$
\left\{
\begin{array}{l}
a = xy \\
b = \mathsf{sin}(a) \\
c = x + b
\end{array}
\right. ,
\qquad
\begin{array}{ll}
x \in [-1, 1] & a \in [-\infty, \infty] \\
y \in [-1, 1] & b \in [-\infty, \infty] \\
z \in [-1, 1] & c \in [-\infty, 0]
\end{array}
$$

1.5 Set and contractors

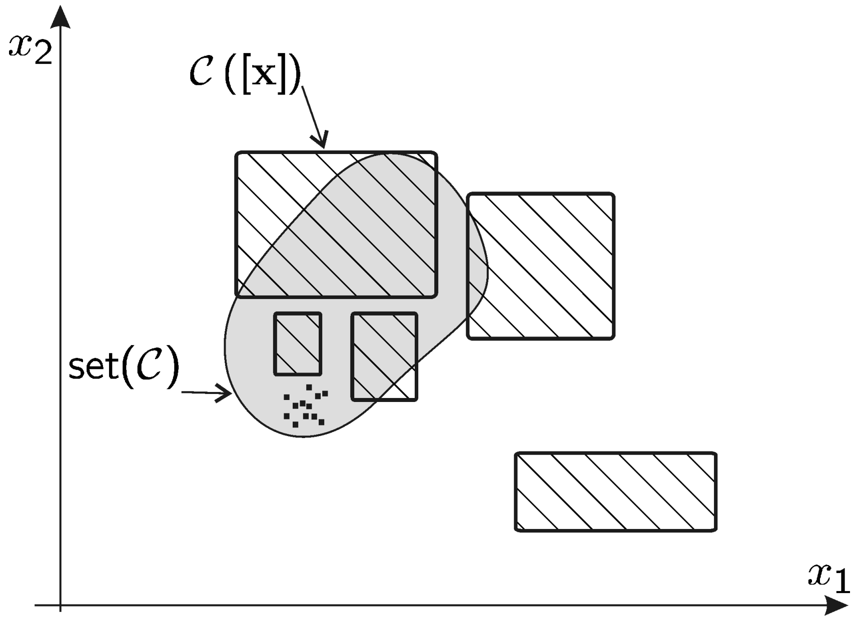
A contractor is one way to represent a set of $\mathbb{R}^n$.

The set associated with a contractor $\mathcal{C}$ is

$$\text{set}\,(\mathcal{C}) = \{\mathbf{x} \in \mathbb{R}^n, \mathcal{C}(\{\mathbf{x}\}) = \{\mathbf{x}\}\}\,.$$



x_2
$[\mathbf{x}]$
$\mathsf{set}(\mathcal{C})$
x_1


x_2
$\mathcal{C}([\mathbf{x}])$
set($\mathcal{C}$)
x_1

1.6 Operations on contractors

intersection	$(\mathcal{C}_1 \cap \mathcal{C}_2)([\mathbf{x}]) \stackrel{\text{def}}{=} \mathcal{C}_1([\mathbf{x}]) \cap \mathcal{C}_2([\mathbf{x}])$
union	$(\mathcal{C}_1 \cup \mathcal{C}_2)([\mathbf{x}]) \stackrel{\text{def}}{=} [\mathcal{C}_1([\mathbf{x}]) \cup \mathcal{C}_2([\mathbf{x}])]$
composition	$(\mathcal{C}_1 \circ \mathcal{C}_2)([\mathbf{x}]) \stackrel{\text{def}}{=} \mathcal{C}_1(\mathcal{C}_2([\mathbf{x}]))$
repetition	$\mathcal{C}^\infty \stackrel{\text{def}}{=} \mathcal{C} \circ \mathcal{C} \circ \mathcal{C} \circ \dots$
repeat intersection	$\mathcal{C}_1 \sqcap \mathcal{C}_2 = (\mathcal{C}_1 \cap \mathcal{C}_2)^\infty$
repeat union	$\mathcal{C}_1 \sqcup \mathcal{C}_2 = (\mathcal{C}_1 \cup \mathcal{C}_2)^\infty$

1.7 QUIMPER

Quimper is a high-level language for QUick Interval Modeling and Programming in a bounded-ERror context.

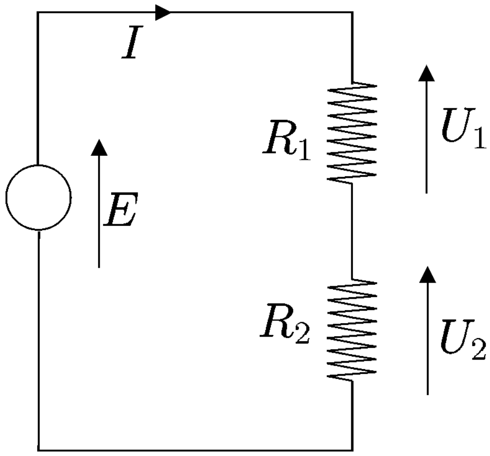
Quimper is an interpreted language for set computation.

Quimper is available at

http://ibex-lib.org/

2 Application of contractors

2.1 Estimation problem



Constraints

$$\begin{aligned} P &= EI;\ E = (R_1 + R_2)\, I; \\ U_1 &= R_1 I;\ U_2 = R_2 I;\ E = U_1 + U_2. \end{aligned}$$

Initial domains

$$\begin{array}{cc} R_1 \in [0, \infty]\Omega, & R_2 \in [0, \infty]\Omega, \\ E \in [23, 26]\mathrm{V}, & I \in [4, 8]\mathrm{A}, \\ U_1 \in [10, 11]\mathrm{V}, & U_2 \in [14, 17]\mathrm{V}, \\ P \in [124, 130]\mathrm{W}, & \end{array}$$

Constraints

$$
\begin{aligned}
P &= EI;\ E = (R_1 + R_2)\, I; \\
U_1 &= R_1 I;\ U_2 = R_2 I;\ E = U_1 + U_2.
\end{aligned}
$$

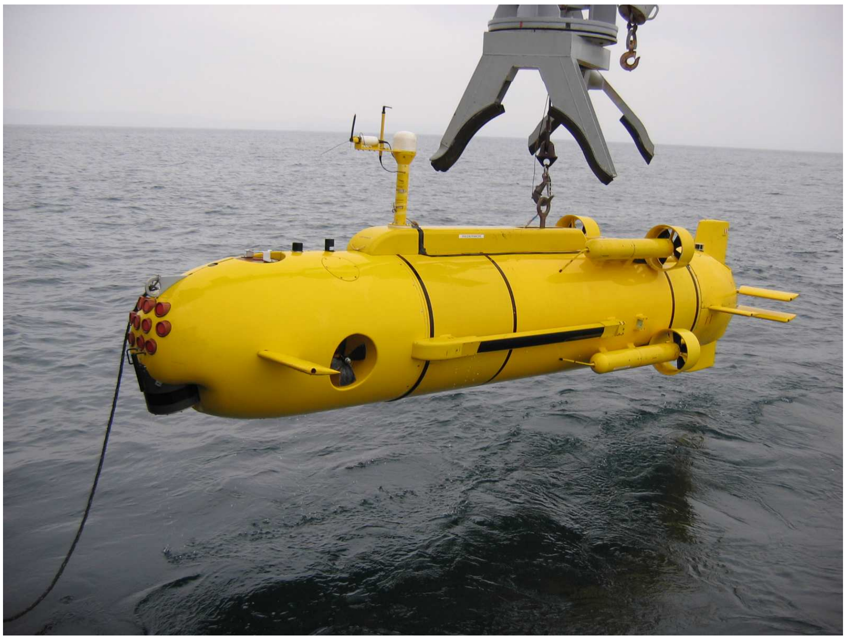
We get the contracted domains

$$\begin{array}{ll} R_1 \in [\mathbf{1.84}, \mathbf{2.31}]\,\Omega, & R_2 \in [\mathbf{2.58}, \mathbf{3.35}]\Omega, \\ E \in [\mathbf{24}, 26]\text{V}, & I \in [\mathbf{4.769}, \mathbf{5.417}]\,\text{A}, \\ U_1 \in [10, 11]\text{V}, & U_2 \in [14, \mathbf{16}]\text{V}, \\ P \in [124, 130]\text{W}, & \end{array}$$

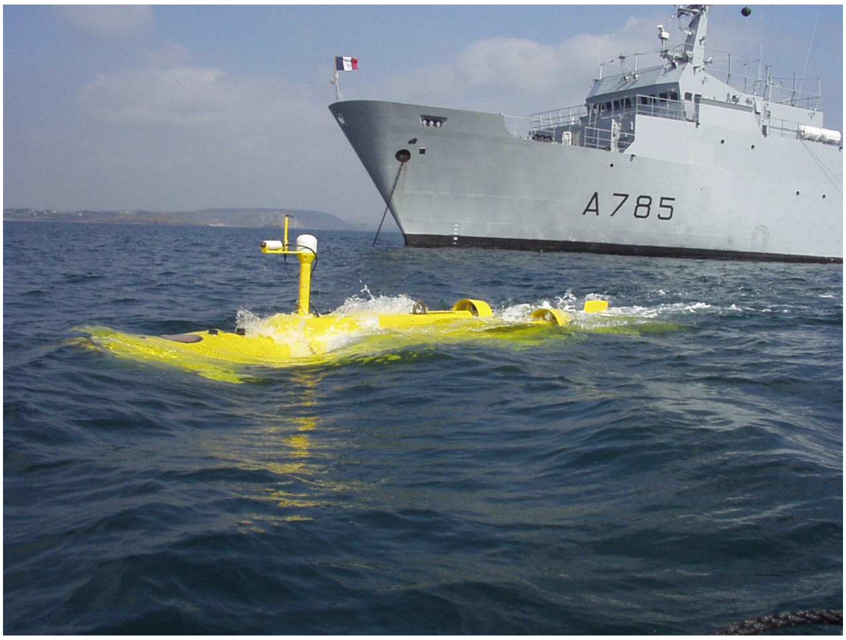
instead of the initial domains

$$\begin{array}{ll} R_1 \in [0, \infty]\Omega, & R_2 \in [0, \infty]\Omega, \\ E \in [23, 26]\text{V}, & I \in [4, 8]\text{A}, \\ U_1 \in [10, 11]\text{V}, & U_2 \in [14, 17]\text{V}, \\ P \in [124, 130]\text{W}, & \end{array}$$

2.2 SLAM



Redermor, GESMA
(Groupe d'Etude Sous-Marine de l'Atlantique)

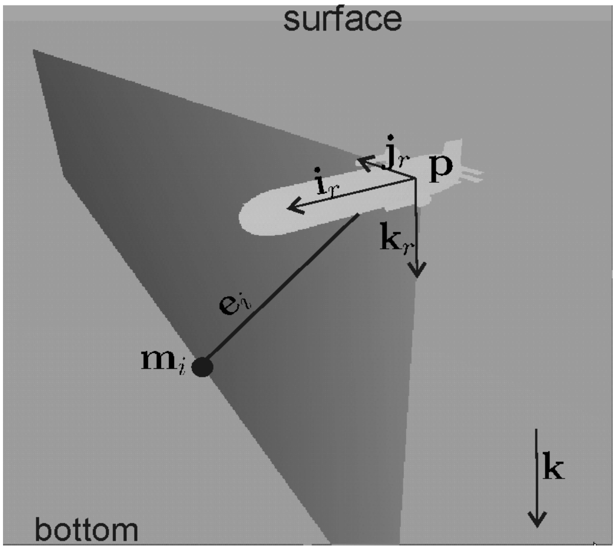

A785

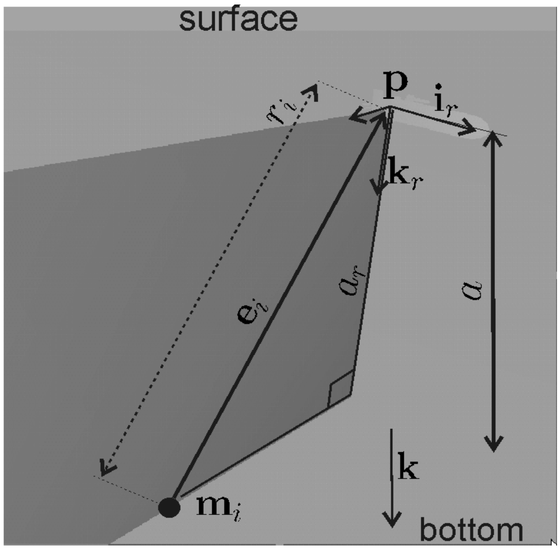
Montrer la simulation

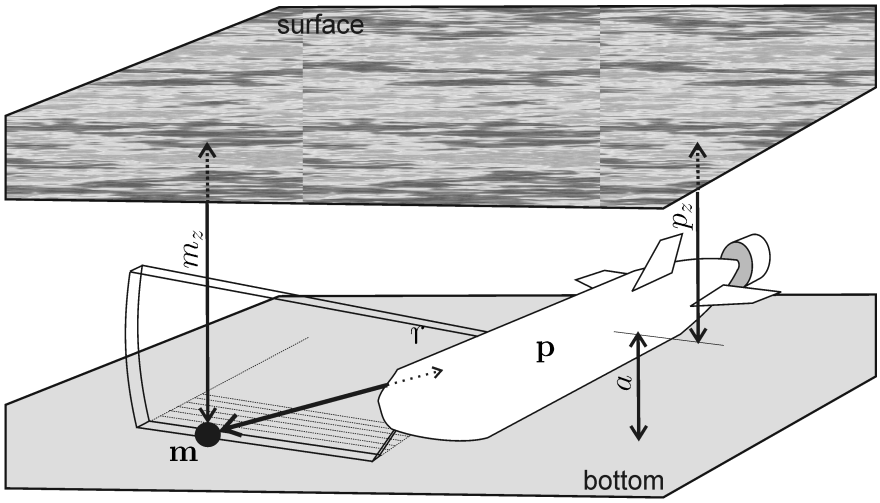
GPS (Global positioning system), only at the surface.

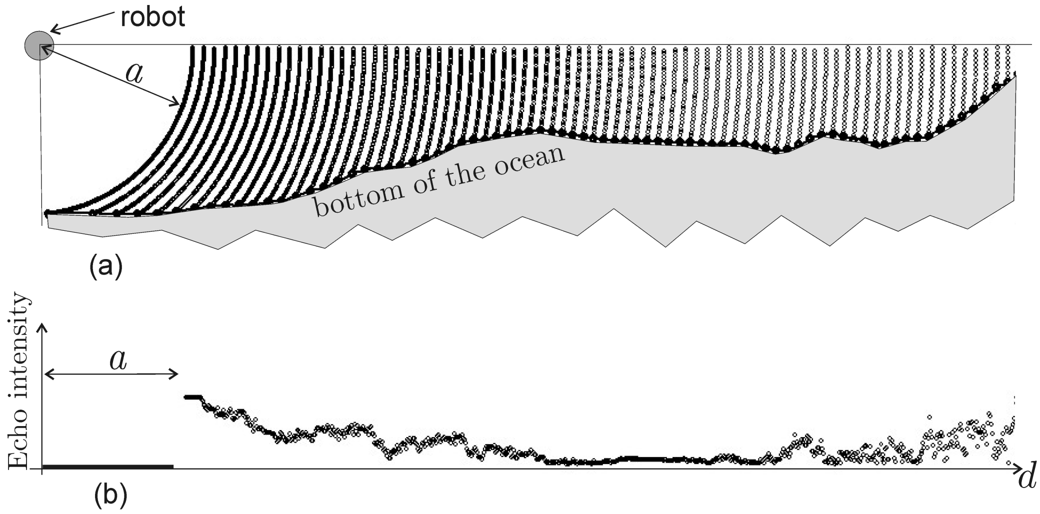
$$t_0 = 6000 \text{ s}, \quad \ell^0 = (-4.4582279^\circ, 48.2129206^\circ) \pm 2.5m$$
$$t_f = 12000 \text{ s}, \quad \ell^f = (-4.4546607^\circ, 48.2191297^\circ) \pm 2.5m$$

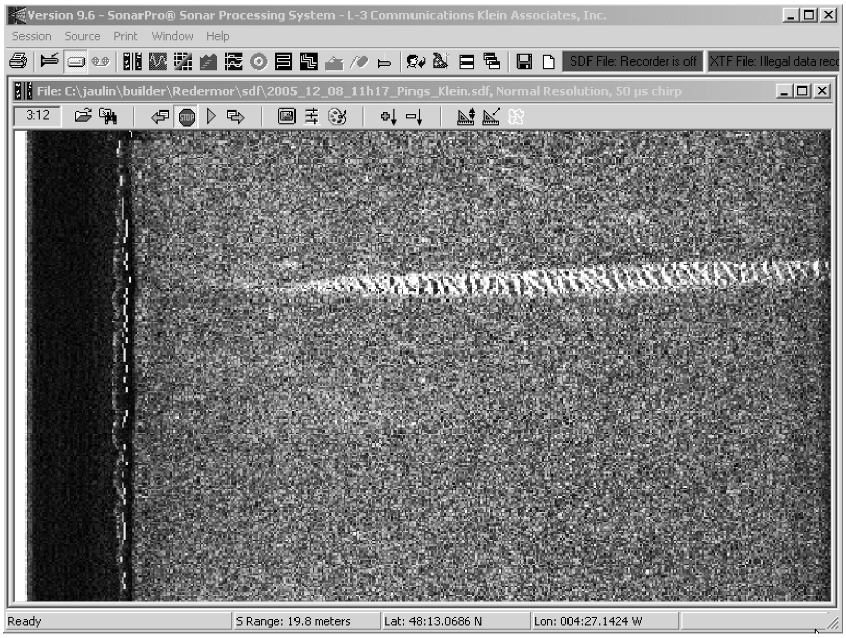
Sonar (KLEIN 5400 side scan sonar).



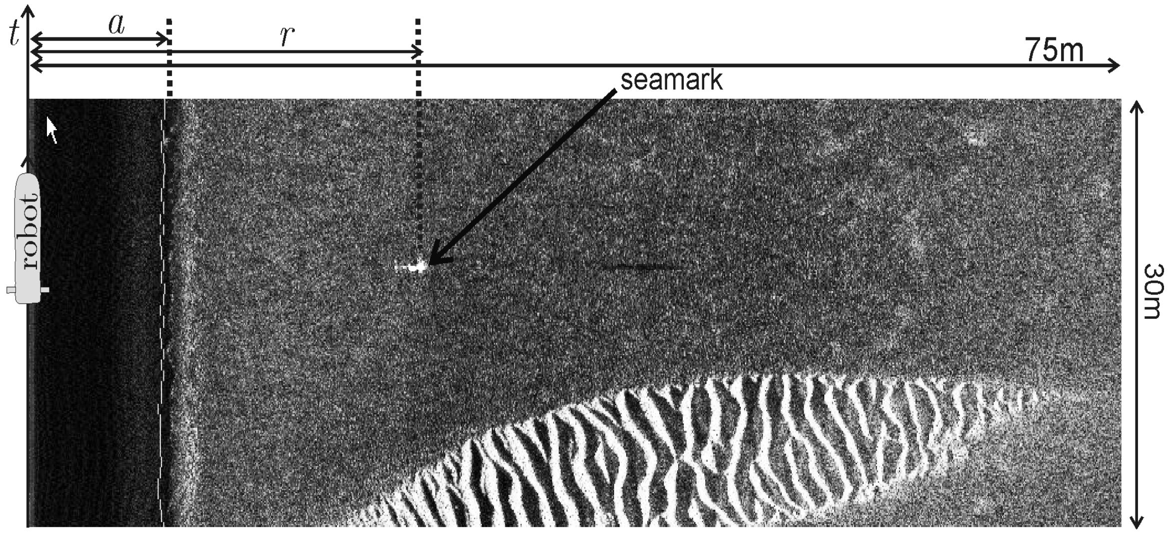



surface
m_z
p_z
r
$\mathbf{p}$
a
$\mathbf{m}$
bottom


robot
a
bottom of the ocean
(a)
Echo intensity
a
d
(b)



Screenshot of SonarPro



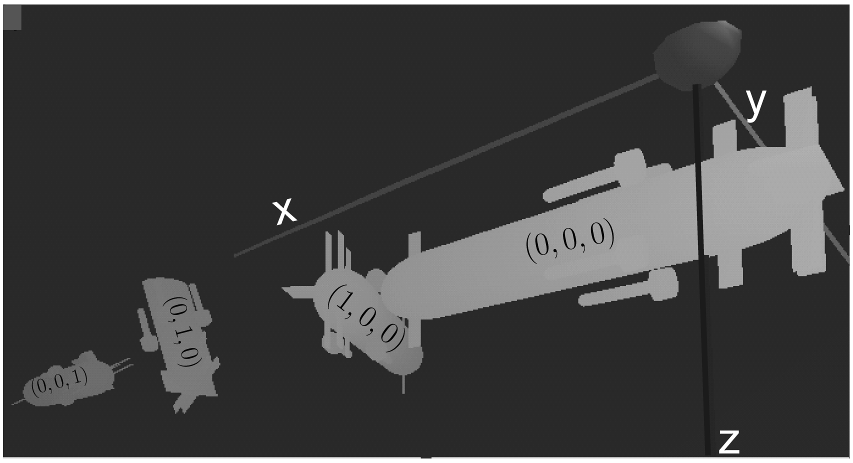
Mine detecttion with SonarPro

Loch-Doppler returns the speed robot $\mathbf{v}_r$.

$$\mathbf{v}_r \in \tilde{\mathbf{v}}_r + 0.004 * [-1, 1] \, .\tilde{\mathbf{v}}_r + 0.004 * [-1, 1]$$

Inertial central (Octans III from IXSEA).

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



Six mines have been detected.

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

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

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

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

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

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

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

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

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

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

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

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

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

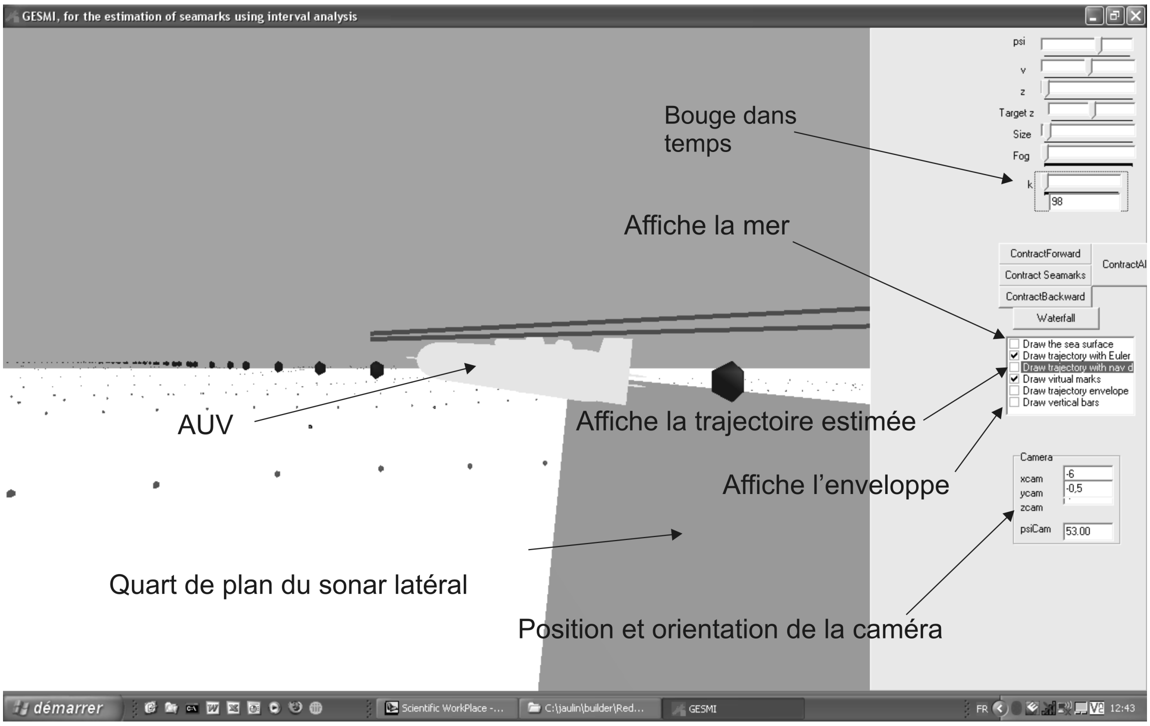
```
//------------------------------------------------------
Constants
N = 59996; // Number of time steps
Variables
  R[N-1][3][3], // rotation matrices
  p[N][3], // positions
  v[N-1][3], // speed vectors
  phi[N-1],theta[N-1],psi[N-1]; // Euler angles
  px[N],py[N]; // for display only
//------------------------------------------------------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

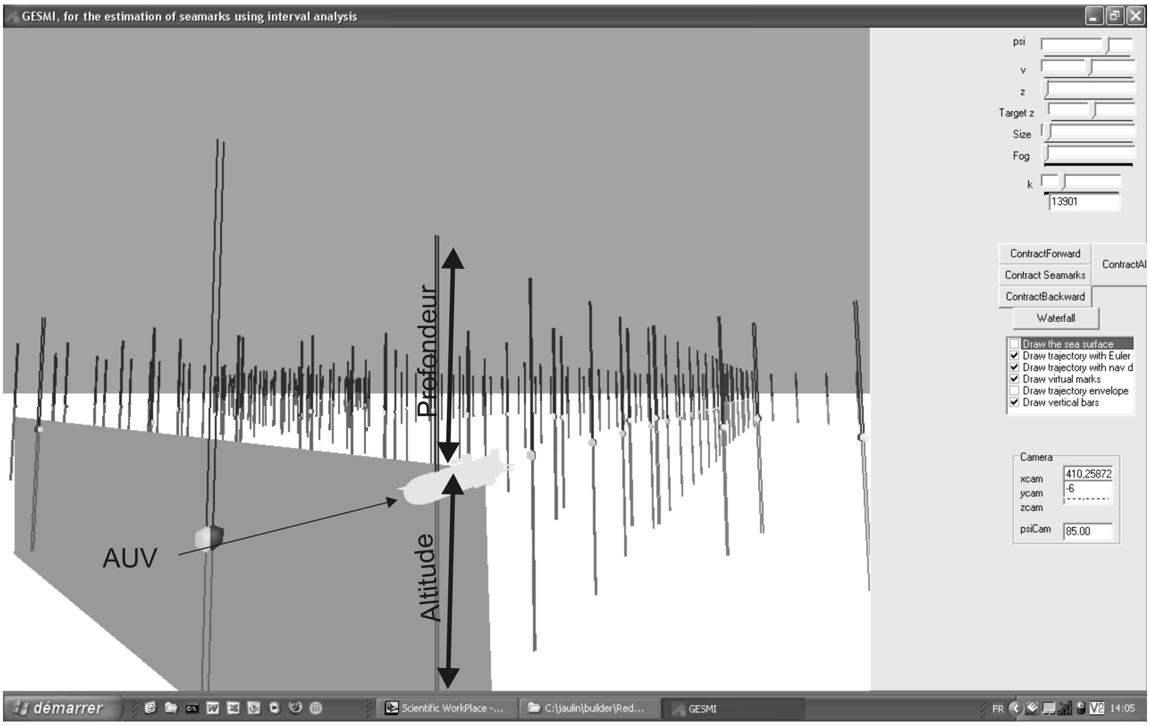
```
function R[3][3]=euler(phi,theta,psi)
  cphi = cos(phi);
  sphi = sin(phi);
  ctheta = cos(theta);
  stheta = sin(theta);
  cpsi = cos(psi);
  spsi = sin(psi);
  R[1][1]=ctheta*cpsi;
  R[1][2]=-cphi*spsi+stheta*cpsi*sphi;
  R[1][3]=spsi*sphi+stheta*cpsi*cphi;
  R[2][1]=ctheta*spsi;
  R[2][2]=cpsi*cphi+stheta*spsi*sphi;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

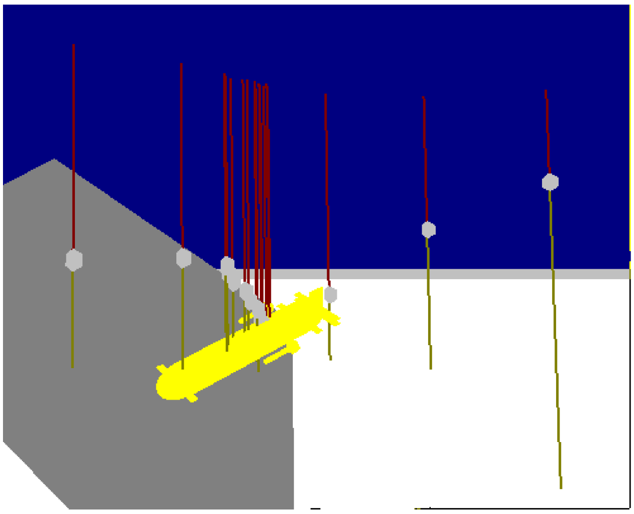
```
contractor-list rotation
  for k=1:N-1;
    R[k]=euler(phi[k],theta[k],psi[k]);
  end
end
//---------------------------------------------------------
contractor-list statequ
  for k=1:N-1;
    p[k+1]=p[k]+0.1*R[k]*v[k];
  end
end
//---------------------------------------------------------
contractor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

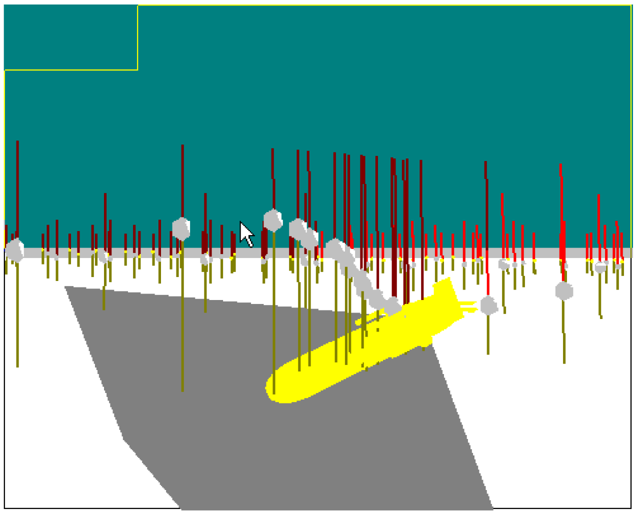
```
contractor fwd
  inter k=1:N-1;
    statequ(k)
  end
end
//-----------------------------------------------------------
contractor bwd
  inter k=1:N-1;
    statequ(N-k)
  end
end
```

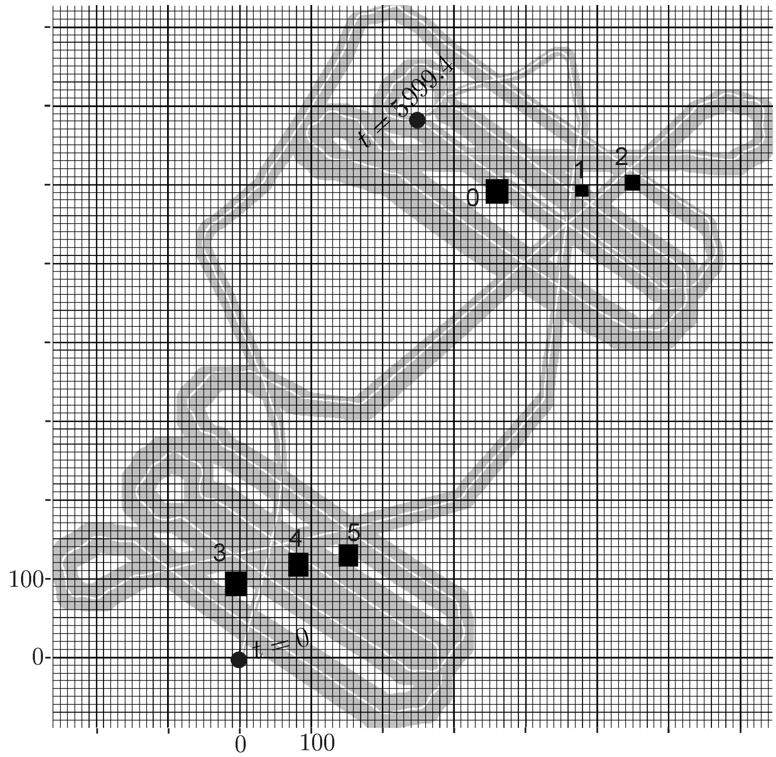
```
main
  p[1] :=read("gps_init.dat");
  v :=read("Quimper_v.dat");
  phi :=read("Quimper_phi.dat");
  theta :=read("Quimper_theta.dat");
  psi :=read("Quimper_psi.dat");
  init;
  fwd;
  bwd;
  column(p,px,1);
  column(p,py,2);
  print("--- Robot positions: ---");
  newplot("gesmi.dat");
  plot(px,py,color(rgb(1,1,1),rgb(0,0,0)));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


GESMI, for the estimation of seamarks using interval analysis
Bouge dans temps
Affiche la mer
AUV
Affiche la trajectoire estimée
Affiche l'enveloppe
Quart de plan du sonar latéral
Position et orientation de la caméra
psi
v
z
Target z
Size
Fog
k
98
ContractForward
Contract Seamarks
ContractBackward
Waterfall
Draw the sea surface
Draw trajectory with Euler
Draw virtual marks
Draw trajectory envelope
Draw vertical bars
Camera
xcam
ycam
zcam
psiCam
-6
-0,5
53.00
démarrer
GESMI
12:43


GESMI, for the estimation of seamarks using interval analysis
Profondeur
AUV
Altitude
psi
v
z
Target z
Size
Fog
k
13901
ContractForward
Contract Seamarks
ContractBackward
ContractAl
Waterfall
Draw the sea surface
Draw trajectory with Euler
Draw trajectory with nav d
Draw virtual marks
Draw trajectory envelope
Draw vertical bars
Camera
xcam
ycam
zcam
410,25872
-6
psiCam
85.00
démarrer
Scientific WorkPlace -...
C:\jaulin\builder\Red...
GESMI
FR
14:05







$t = 5999.4$
0
1
2
3
4
5
100
0
$t = 0$
0
100

3 Applications with bissections

3.1 Sailboat

State equations

$$
\left\{
\begin{array}{rcl}
\dot{x} & = & v\cos\theta \\
\dot{y} & = & v\sin\theta - 1 \\
\dot{\theta} & = & \omega \\
\dot{\delta}_s & = & u_1 \\
\dot{\delta}_r & = & u_2 \\
\dot{v} & = & f_s\sin\delta_s - f_r\sin\delta_r - v \\
\dot{\omega} & = & (1-\cos\delta_s)\, f_s - \cos\delta_r.f_r - \omega \\
f_s & = & \cos(\theta+\delta_s) - v\sin\delta_s \\
f_r & = & v\sin\delta_r.
\end{array}
\right.
$$

In a cruising phase

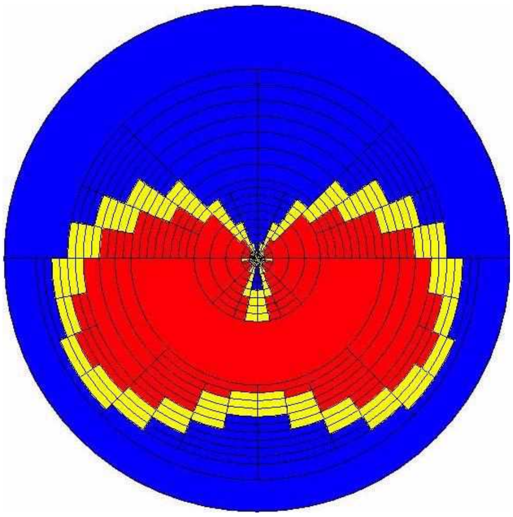
$$\dot{\theta} = 0, \dot{\delta}_s = 0, \dot{\delta}_r = 0, \dot{v} = 0, \dot{\omega} = 0.$$

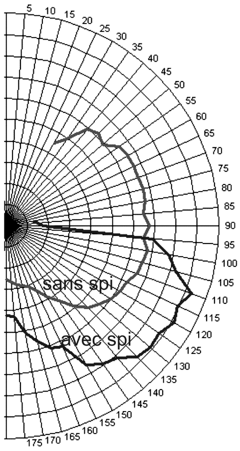
i.e.,

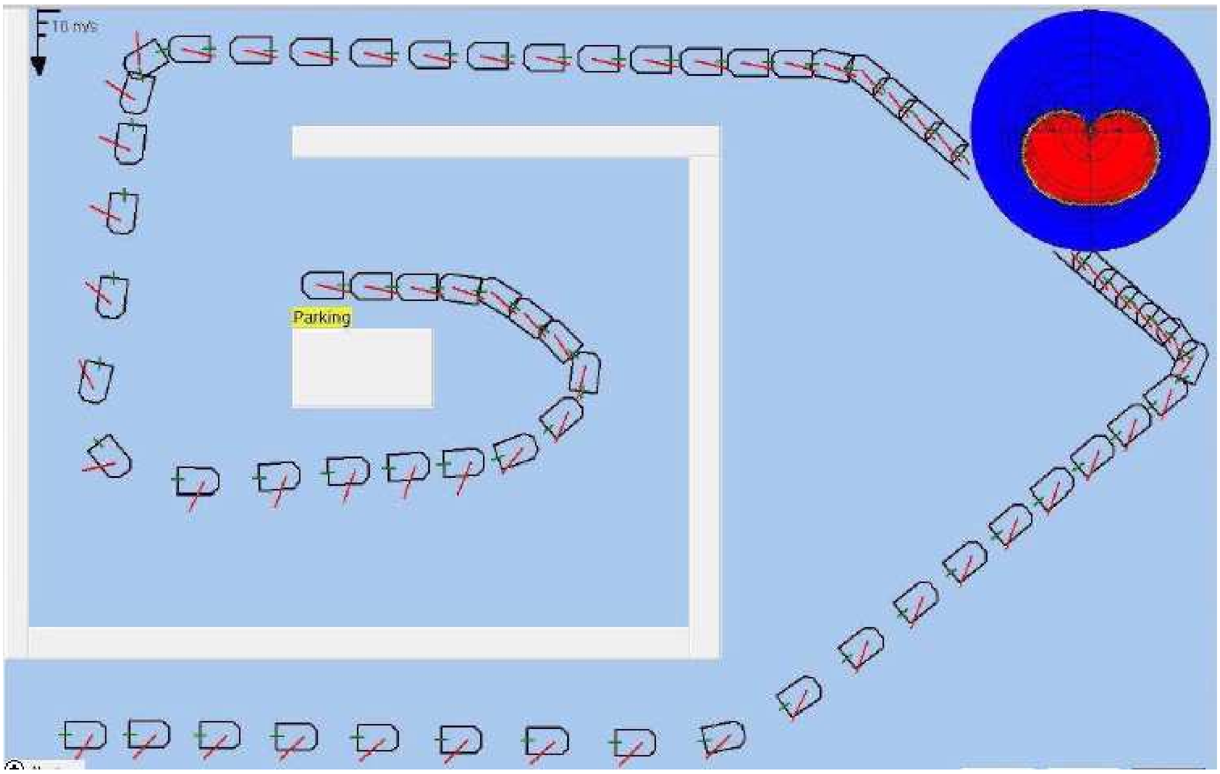
$$\begin{cases} 0 &= \omega \\ 0 &= u_1 \\ 0 &= u_2 \\ 0 &= f_s \sin \delta_s - f_r \sin \delta_r - v \\ 0 &= (1 - \cos \delta_s) f_s - \cos \delta_r . f_r - \omega \\ f_s &= \cos(\theta + \delta_s) - v \sin \delta_s \\ f_r &= v \sin \delta_r . \end{cases}$$

The polar diagram is

$$
\begin{aligned}
\mathbb{S}_y = \ & \{(\theta, v) \qquad \mid \exists f_s, \delta_s, f_r, \delta_r, \\
& f_s \sin\delta_s - f_r \sin\delta_r - v = 0 \\
& (1 - \cos\delta_s)\, f_s - \cos\delta_r f_r = 0 \\
& f_s = \cos(\theta + \delta_s) - v \sin\delta_s \\
& f_r = v \sin\delta_r \ \}
\end{aligned}
$$

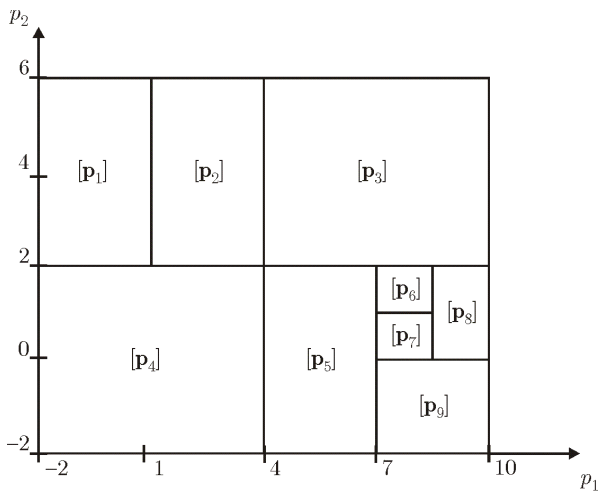



5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135 140 145 150 155 160 165 170 175
sans spi
avec spi

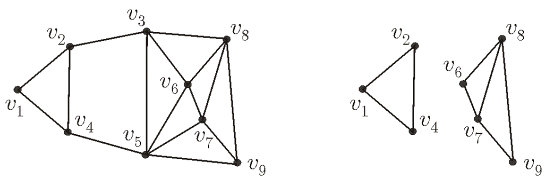


10 m/s
Parking

3.2 Path planning

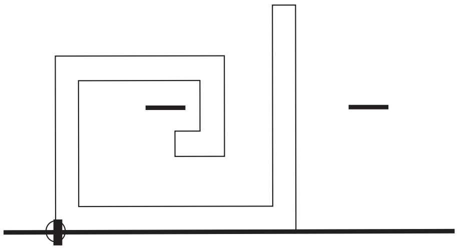


p_2
6
4
2
0
−2
−2
1
4
7
10
p_1
$[\mathbf{p}_1]$
$[\mathbf{p}_2]$
$[\mathbf{p}_3]$
$[\mathbf{p}_4]$
$[\mathbf{p}_5]$
$[\mathbf{p}_6]$
$[\mathbf{p}_7]$
$[\mathbf{p}_8]$
$[\mathbf{p}_9]$

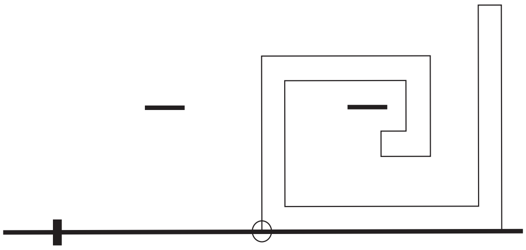


v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9

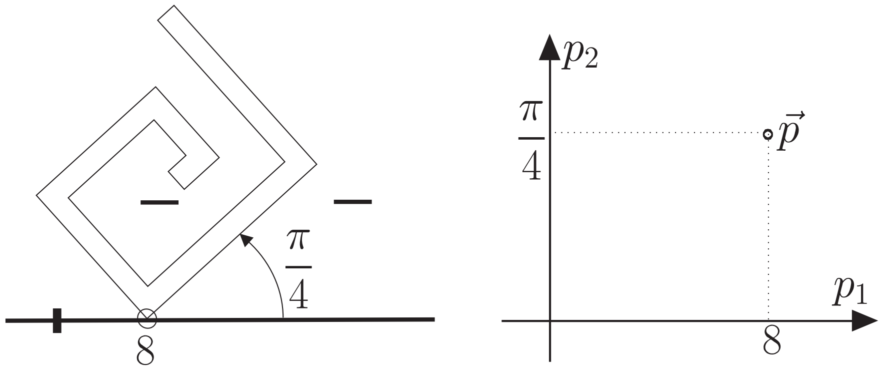
(a) (b)



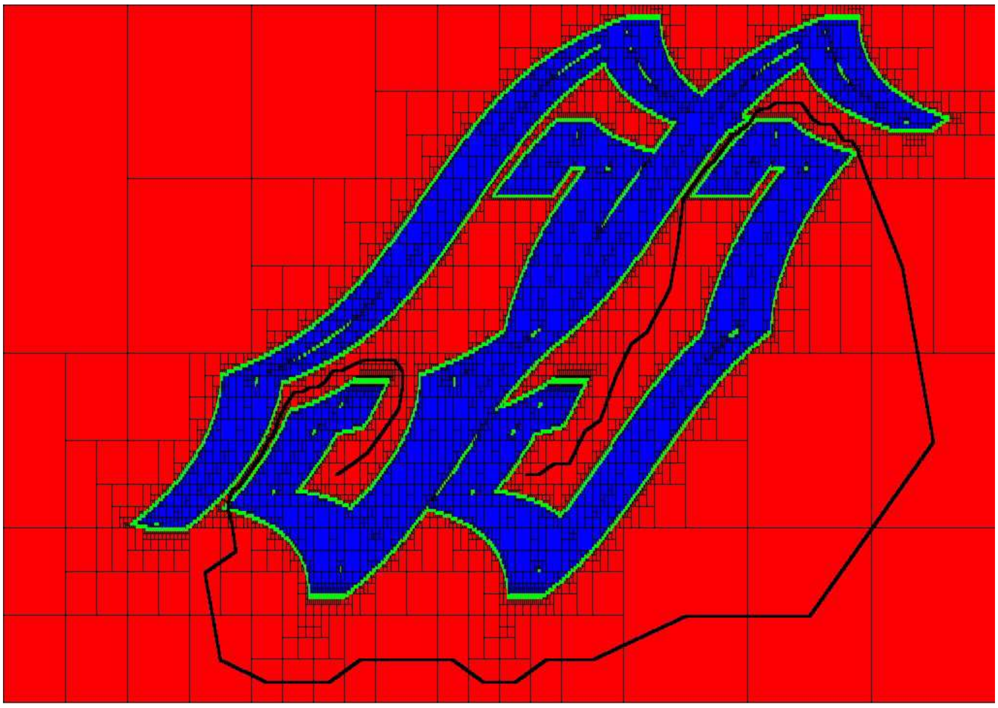
Initial configuration: $\vec{p} = (0\ 0)^{\mathrm{T}}$

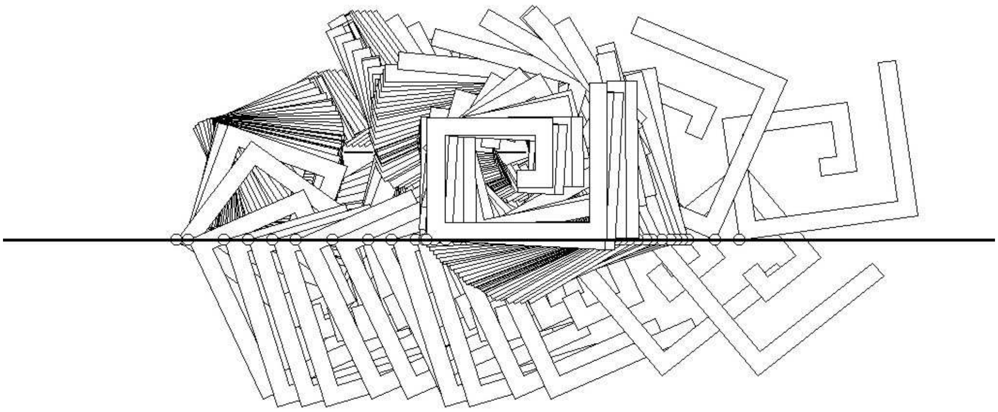


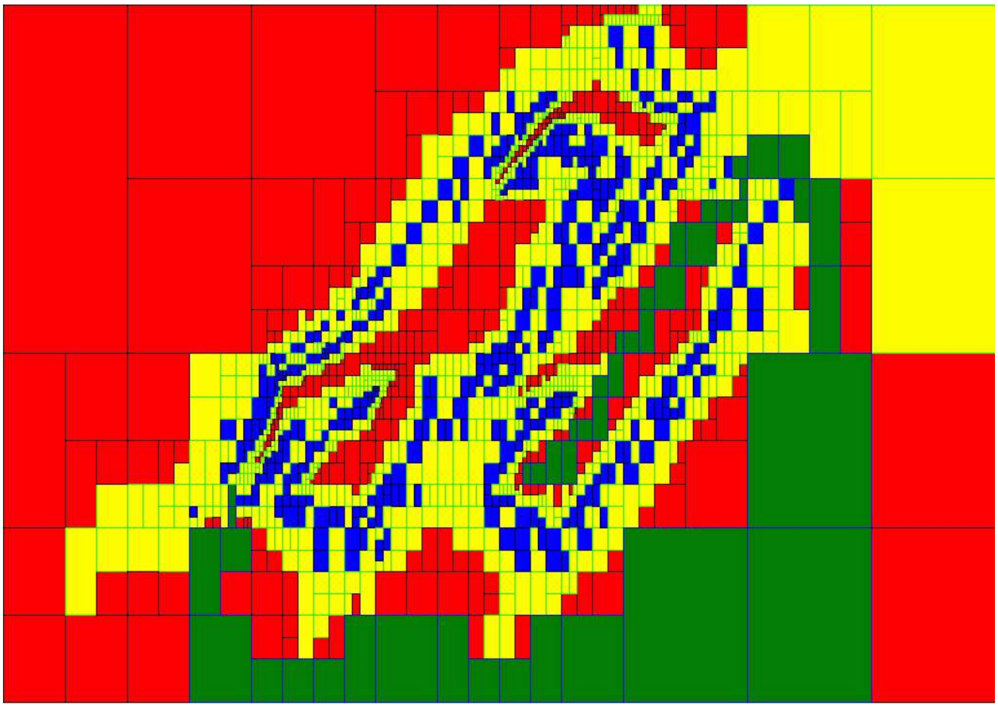
Goal configuration: $\vec{p} = (17\ 0)^{\mathrm{T}}$

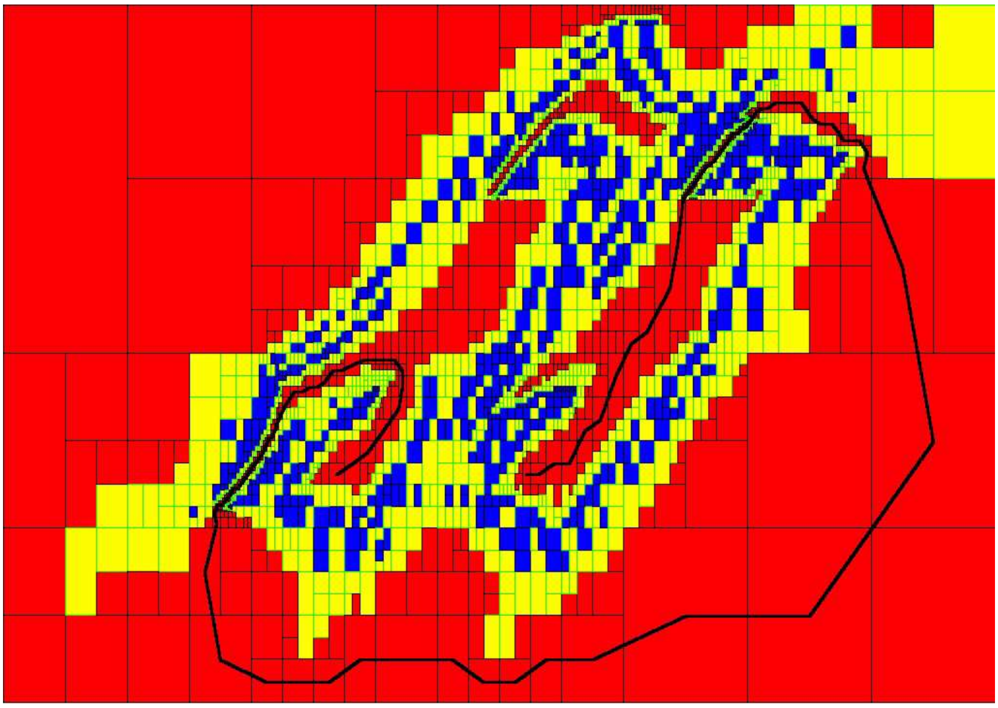


Room Configuration space









3.3 Counting connected components

(Collaboration with N. Delanoue and B. Cottenceau)

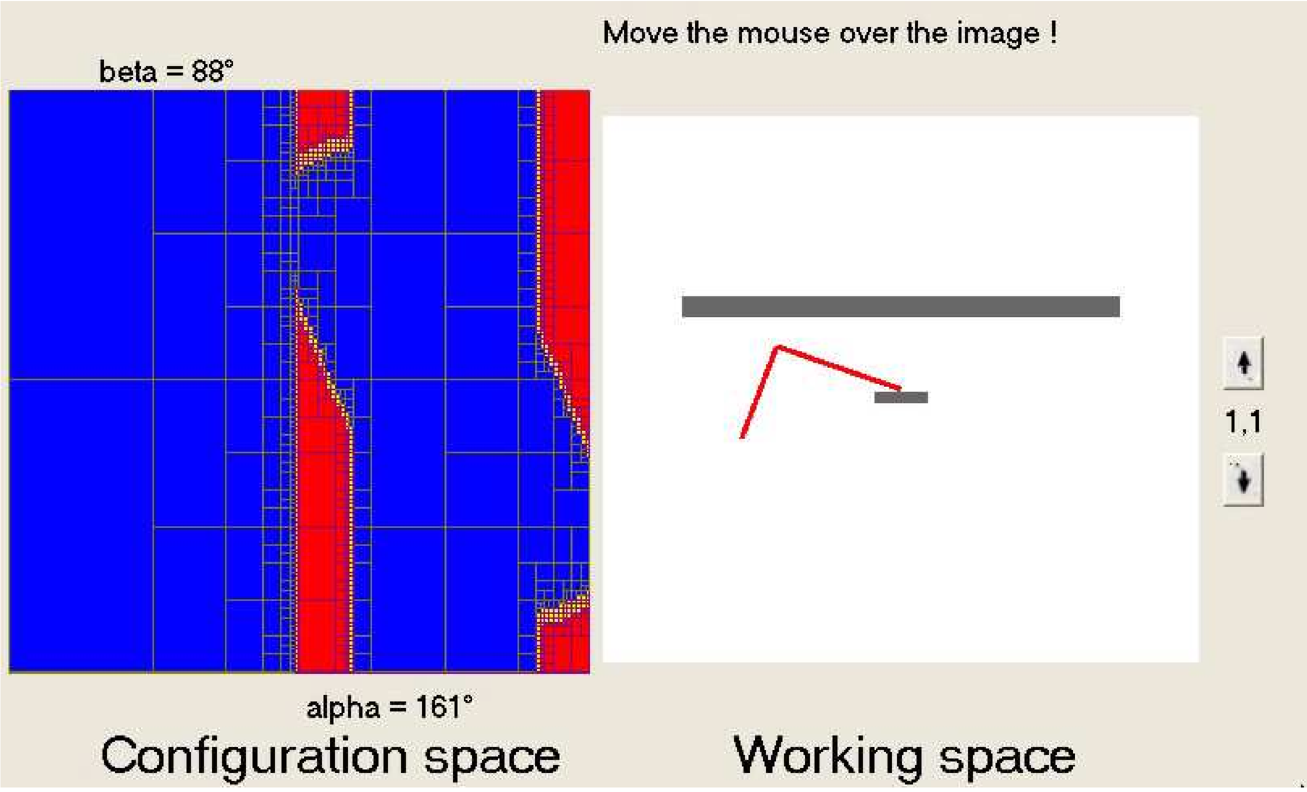
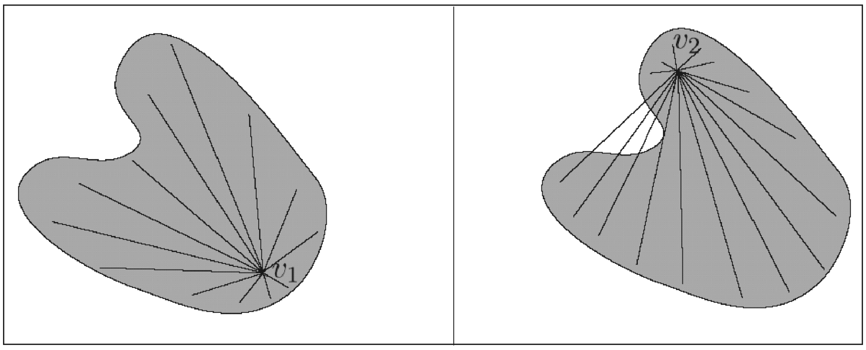


Figure 1:

The point $\mathbf{v}$ is a *star* for $\mathbb{S} \subset \mathbb{R}^n$ if $\forall \mathbf{x} \in \mathbb{S}$, $\forall \alpha \in [0, 1]$, $\alpha \mathbf{v} + (1 - \alpha)\mathbf{x} \in \mathbb{S}$. .



$\mathbf{v}_1$ is a star for $\mathbb{S}$ whereas $\mathbf{v}_2$ is not

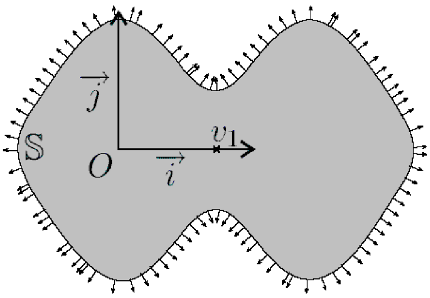
The set $\mathbb{S} \subset \mathbb{R}^n$ is *star-shaped* is there exists $\mathbf{v}$ such that $\mathbf{v}$ is a star for $\mathbb{S}$.

Theorem: Define the set

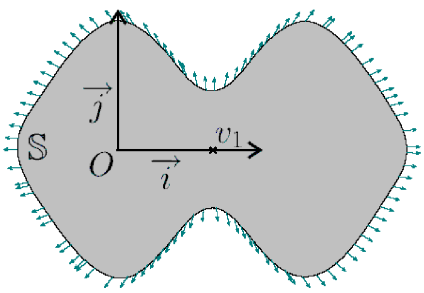
$$\mathbb{S} \overset{\text{def}}{=} \{\mathbf{x} \in [\mathbf{x}] | f(\mathbf{x}) \leq 0\}$$

where f is differentiable. We have

$$\left\{\mathbf{x} \in [\mathbf{x}] \mid f(\mathbf{x}) = 0, \frac{df}{d\mathbf{x}}(\mathbf{x}).(\mathbf{x} - \mathbf{v}) \leq 0\right\} = \emptyset \Rightarrow \mathbf{v} \text{ is a star for}$$



$Df(x)$



$x - v_1$

Consider a subpaving $\mathcal{P} = \{[\mathbf{p}_1], [\mathbf{p}_2], \ldots\}$ covering $\mathbb{S}$. The relation $\mathcal{R}$ defined by

$$[\mathbf{p}]\mathcal{R}[\mathbf{q}] \Leftrightarrow \mathbb{S} \cap [\mathbf{p}] \cap [\mathbf{q}] \neq \emptyset$$

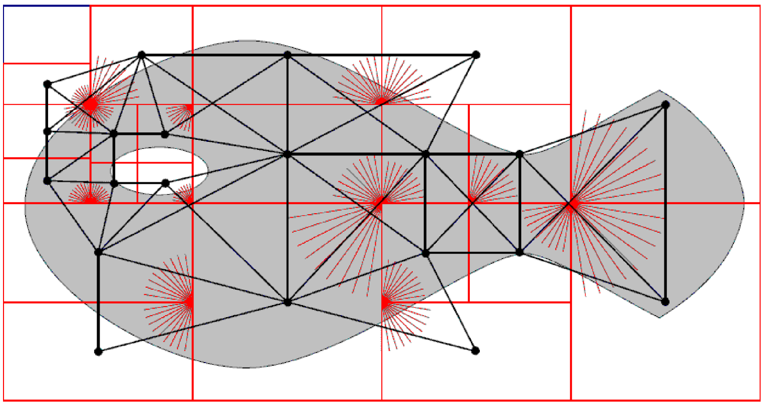
is *star-spangled graph* of the set $\mathbb{S}$ if

$$\forall [\mathbf{p}] \in \mathcal{P}, \mathbb{S} \cap [\mathbf{p}] \text{ is star-shaped.}$$

A star-spangled graph for the set

$$\mathbb{S} \stackrel{\text{def}}{=} \left\{ (x, y) \in \mathbb{R}^2 \mid \begin{pmatrix} x^2 + 4y^2 - 16 \\ 2\sin x - \cos y + y^2 - \frac{3}{2} \\ -(x + \frac{5}{2})^2 - 4(y - \frac{2}{5})^2 + \frac{3}{10} \end{pmatrix} \leq 0 \right\},$$

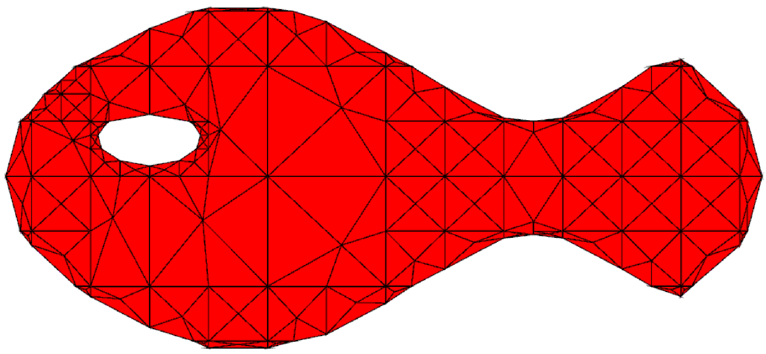
is



For each $[\mathbf{p}]$ of the paving $\mathcal{P}$, a common star located at the corner of $[\mathbf{p}]$ (represented in red) has been found.

Theorem: The number of connected components of the star-spangled graph of $\mathbb{S}$ is equal to that of $\mathbb{S}$.

An extension of this approach has also been developed to compute a triangulation homeomorphic to $\mathbb{S}$.



3.4 Capture basin

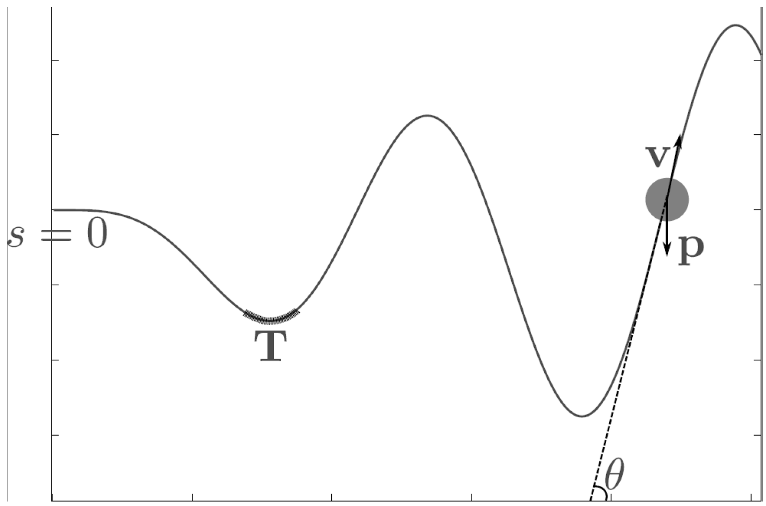
(With M. Lhommeau and L. Hardouin)

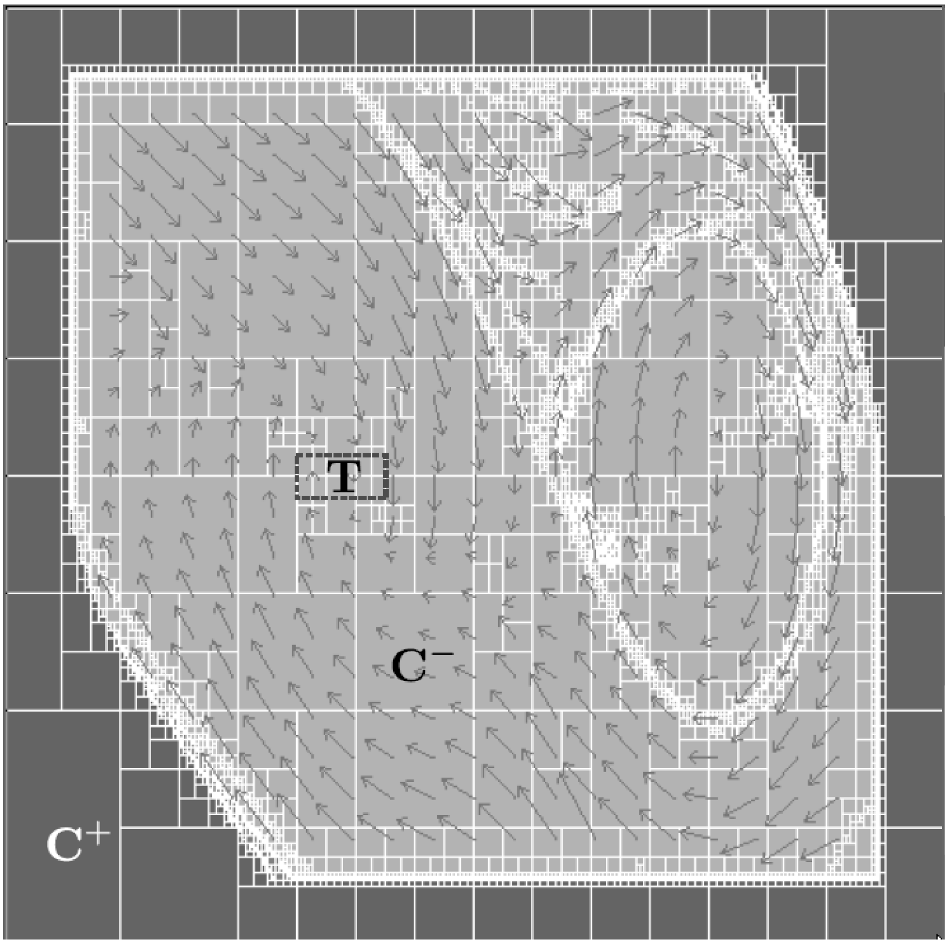
Consider a rolling ball described by

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\sin\left(\theta(x_1)\right) - x_2 + u \end{cases}$$

where

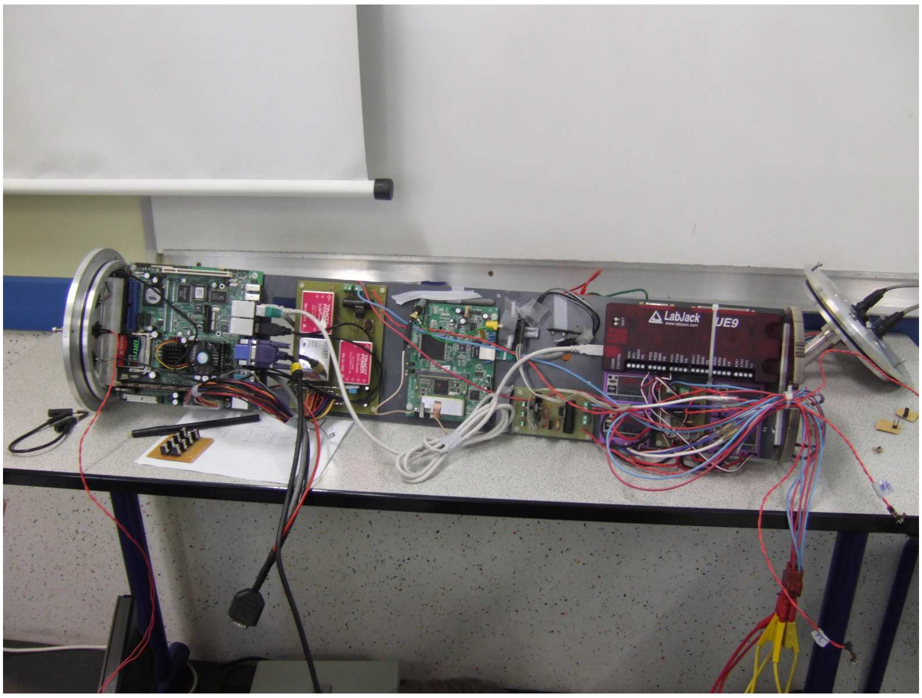
$$\theta(x) = \sin(1.1.x) - \frac{1}{2}\sin(x)$$

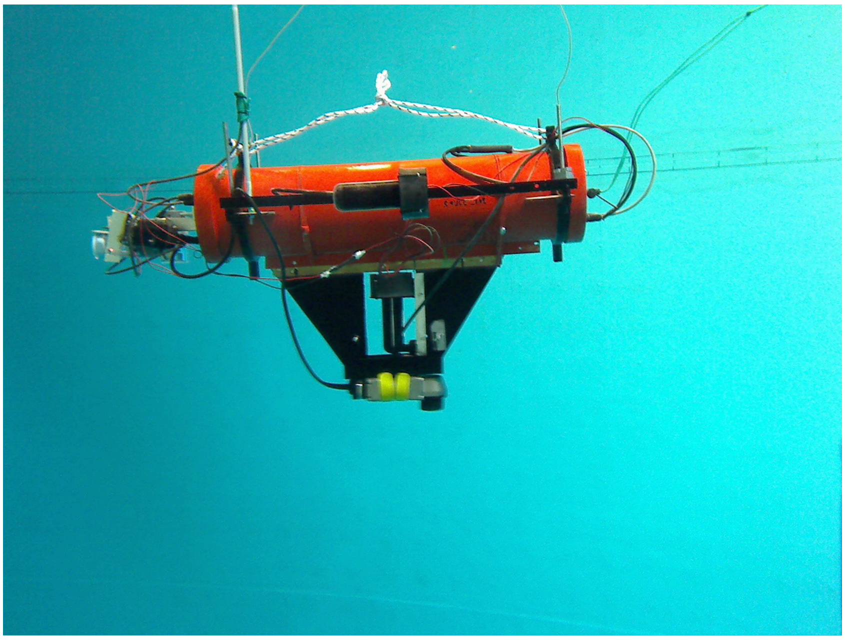



T
C−
C+

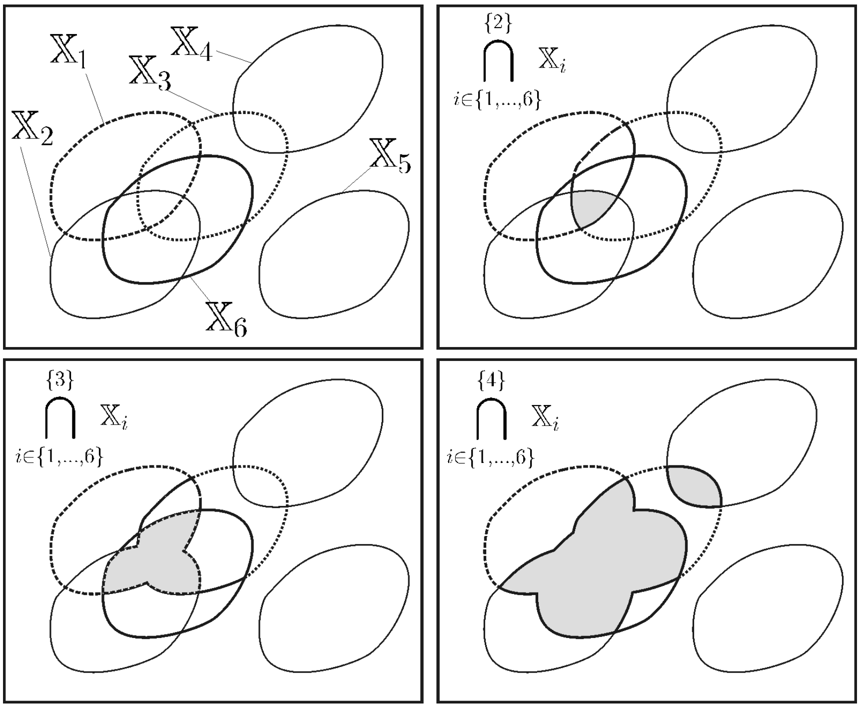
3.5 Robust state estimation

Portsmouth, July 12-15, 2007.


LabJack
UE9



Sauc'isse robot



q-relaxed intersection the 6 sets