Intervals for control

Luc Jaulin, ENSIETA, Brest

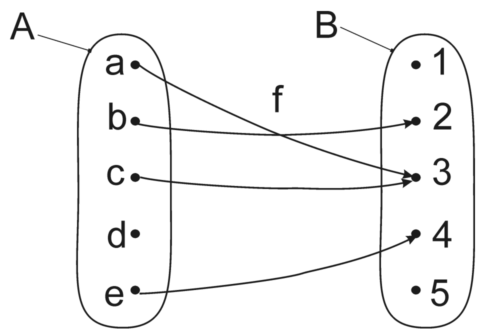
www.ensieta.fr/jaulin/

Mexico, March 17-19, 2010

1 Interval approach

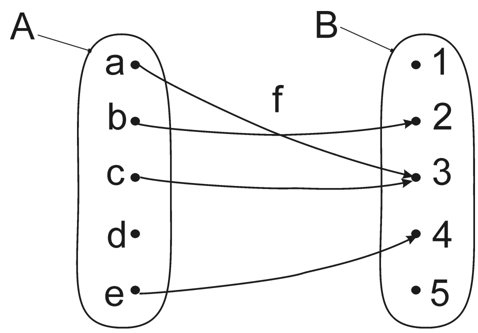
1.1 Basic notions on set theory

Exercise: If f is defined as follows



$$\begin{aligned}
f(A) &= ?. \\
f^{-1}(B) &= ?. \\
f^{-1}(f(A)) &= ? \\
f^{-1}(f(\{b,c\})) &= ?.
\end{aligned}$$

Exercise: If f is defined as follows



$$\begin{aligned} f(A) &= \{2,3,4\} = \mathsf{Im}(f). \\ f^{-1}(B) &= \{a,b,c,e\} = \mathsf{dom}(f). \\ f^{-1}(f(A)) &= \{a,b,c,e\} \subset A \\ f^{-1}(f(\{b,c\})) &= \{a,b,c\}. \end{aligned}$$

Exercise: If $f(x) = x^2$, then

$$\begin{aligned} f([2,3]) &= \text{?} \\ f^{-1}([4,9]) &= \text{?}. \end{aligned}$$

Exercise: If $f(x) = x^2$, then

$$\begin{aligned} f([2,3]) &= [4,9] \\ f^{-1}([4,9]) &= [-3,-2] \cup [2,3]. \end{aligned}$$

This is consistent with the property

$$f\left(f^{-1}(\mathbb{Y})\right) \subset \mathbb{Y}.$$

1.2 Interval arithmetic

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

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

For instance,

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

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

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

For instance,

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

$$
\begin{aligned}
[x^-, x^+] + [y^-, y^+] = \ & [x^- + y^-, x^+ + y^+], \\
[x^-, x^+].[y^-, y^+] = \ & [x^-y^- \wedge x^+y^- \wedge x^-y^+ \wedge x^+y^+, \\
& x^-y^- \vee x^+y^- \vee x^-y^+ \vee x^+y^+], \\
[x^-, x^+] \vee [y^-, y^+] = \ & [\vee(x^-, y^-), \vee(x^+, y^+)].
\end{aligned}
$$

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

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

For instance,

$$\begin{array}{rcl}
\sin([0,\pi]) & = & ?, \\
\mathrm{sqr}([-1,3]) & = & [-1,3]^2 = ?, \\
\mathrm{abs}([-7,1]) & = & ?, \\
\mathrm{sqrt}([-10,4]) & = & \sqrt{[-10,4]} = ?, \\
\log([-2,-1]) & = & ?.
\end{array}$$

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

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

For instance,

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

1.3 Boxes

A *box*, or *interval vector* $[\mathbf{x}]$ of $\mathbb{R}^n$ is

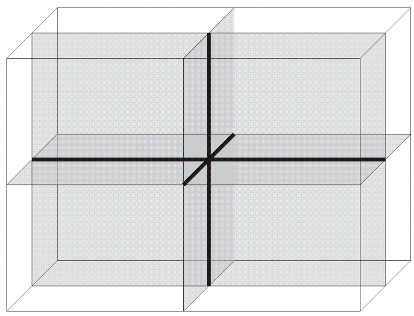
$$[\mathbf{x}] = [x_1^-, x_1^+] \times \cdots \times [x_n^-, x_n^+] = [x_1] \times \cdots \times [x_n].$$

The set of all boxes of $\mathbb{R}^n$ will be denoted by $\mathbb{IR}^n$.

The *width* $w\left([\mathbf{x}]\right)$ of a box $[\mathbf{x}]$ is the length of its largest side. For instance

$$w([1,2]\times[-1,3]) = 4$$

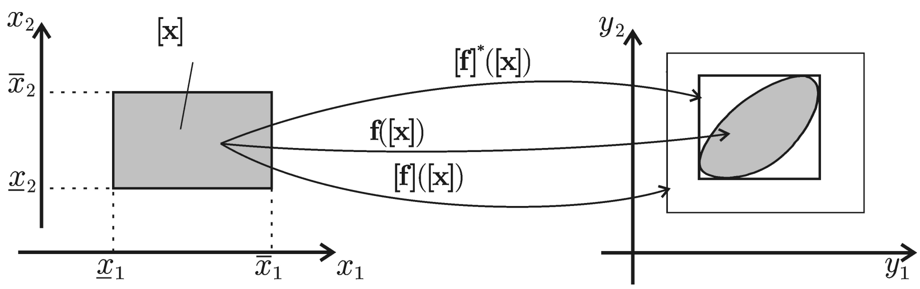
The *principal plane* of $[\mathbf{x}]$ is the symmetric plane $[\mathbf{x}]$ perpendicular to its largest side.



1.4 Inclusion function

The interval function $[\mathbf{f}]$ from $\mathbb{IR}^n$ to $\mathbb{IR}^m$, is an *inclusion function* of $\mathbf{f}$ if

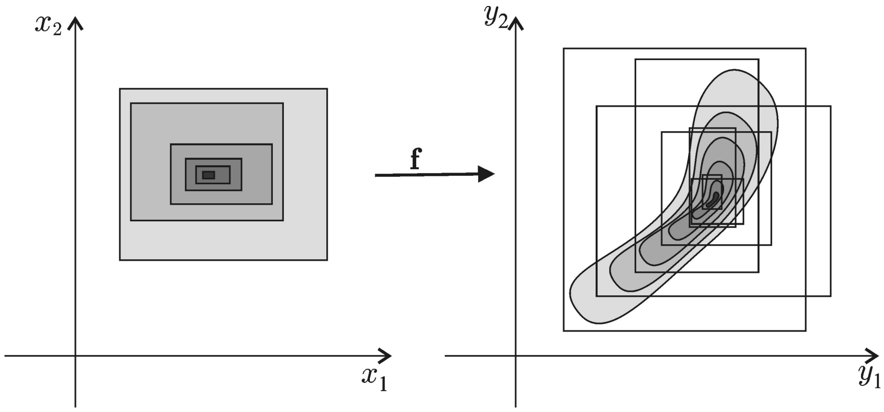
$$\forall\, [\mathbf{x}] \in \mathbb{IR}^n, \quad \mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}]\,([\mathbf{x}]).$$



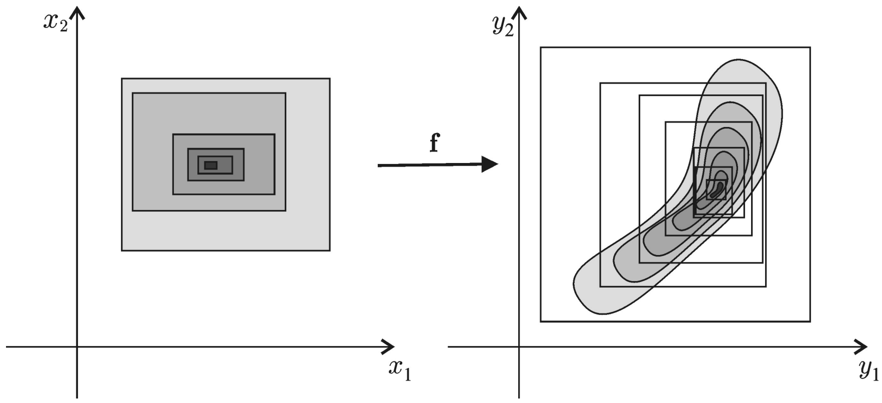
Inclusion functions $[\mathbf{f}]$ and $[\mathbf{f}]^*$; here, $[\mathbf{f}]^*$ is minimal.

The inclusion function $[\mathbf{f}]$ is

monotonic	if	$([\mathbf{x}] \subset [\mathbf{y}]) \Rightarrow ([\mathbf{f}]([\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{y}]))$
minimal	if	$\forall [\mathbf{x}] \in \mathbb{IR}^n,\ [\mathbf{f}]([\mathbf{x}]) = [\mathbf{f}([\mathbf{x}])]$
thin	if	$w([\mathbf{x}]) = 0 \Rightarrow w([\mathbf{f}]([\mathbf{x}]) = 0$
convergent	if	$w([\mathbf{x}]) \to 0 \Rightarrow w([\mathbf{f}]([\mathbf{x}]) \to 0$.



Convergent but non-monotonic inclusion function



Convergent and monotonic inclusion function

The natural inclusion function for $f(x) = x^2 + 2x + 4$ is

$$[f]([x]) = [x]^2 + 2[x] + 4.$$

If $[x] = [-3, 4]$, we have

$$\begin{aligned} [f]([-3,4]) &= [-3,4]^2 + 2[-3,4] + 4 \\ &= [0,16] + [-6,8] + 4 \\ &= [-2,28]. \end{aligned}$$

Note that $f([-3,4]) = [3,28] \subset [f]([-3,4]) = [-2,28]$.

A minimal inclusion function for

$$\mathbf{f}: \begin{array}{ccc} \mathbb{R}^2 & \longrightarrow & \mathbb{R}^3 \\ (x_1, x_2) & \mapsto & \left(x_1 x_2, x_1^2, x_1 - x_2\right). \end{array}$$

is

$$[\mathbf{f}]: \begin{array}{ccc} \mathbb{IR}^2 & \rightarrow & \mathbb{IR}^3 \\ ([x_1], [x_2]) & \rightarrow & \left([x_1] * [x_2], [x_1]^2, [x_1] - [x_2]\right). \end{array}$$

If $\mathbf{f}$ is given by the algorithm

Algorithm $\mathbf{f}$(in: $\mathbf{x} = (x_1, x_2, x_3)$, out: $\mathbf{y} = (y_1, y_2)$)

$z := x_1$;
for $k := 0$ to 100
    $z := x_2(z + kx_3)$;
next;
$y_1 := z$;
$y_2 := \sin(zx_1)$;

Its natural inclusion function is

Algorithm $[\mathbf{f}]$(in: $[\mathbf{x}]$, out: $[\mathbf{y}]$)
$[z] := [x_1]$;
for $k := 0$ to 100
$[z] := [x_2] * ([z] + k * [x_3])$;
next;
$[y_1] := [z]$;
$[y_2] :=$ sin$([z] * [x_1])$;

Here, $[\mathbf{f}]$ is a convergent, thin and monotonic inclusion function for $\mathbf{f}$.

1.5 Subpavings

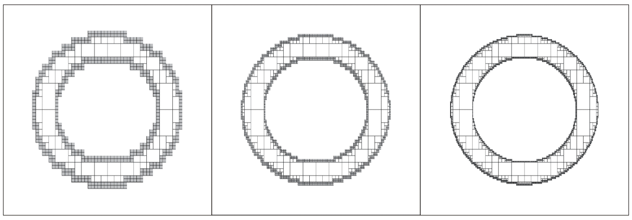
A subpaving of $\mathbb{R}^n$ is a set of non-overlapping boxes of $\mathbb{R}^n$.

Compact sets $\mathbb{X}$ can be bracketed between inner and outer subpavings:

$$\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+.$$

Example.

$$\mathbb{X} = \{(x_1, x_2) \mid x_1^2 + x_2^2 \in [1, 2]\}.$$



Set operations such as $\mathbb{Z} := \mathbb{X} + \mathbb{Y}$, $\mathbb{X} := \mathbf{f}^{-1}(\mathbb{Y})$, $\mathbb{Z} := \mathbb{X} \cap \mathbb{Y} \ldots$ can be approximated by subpaving operations.

1.6 Set inversion

Let $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and let $\mathbb{Y}$ be a subset of $\mathbb{R}^m$. Set inversion is the characterization of

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{f}(\mathbf{x}) \in \mathbb{Y}\} = \mathbf{f}^{-1}(\mathbb{Y}).$$

We shall use the following tests.

$$\begin{array}{lll}\text{(i)} & [\mathbf{f}]([\mathbf{x}]) \subset \mathbb{Y} & \Rightarrow \; [\mathbf{x}] \subset \mathbb{X} \\ \text{(ii)} & [\mathbf{f}]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset & \Rightarrow \; [\mathbf{x}] \cap \mathbb{X} = \emptyset.\end{array}$$

Boxes for which these tests failed, will be bisected, except if they are too small.

Algorithm Sivia(in: $[\mathbf{x}](0), \mathbf{f}, \mathbb{Y}$)	
1	$\mathcal{L} := \{[\mathbf{x}](0)\}$;
2	pull $[\mathbf{x}]$ from $\mathcal{L}$;
3	if $[\mathbf{f}]([\mathbf{x}]) \subset \mathbb{Y}$, draw($[\mathbf{x}]$, 'red');
4	elseif $[\mathbf{f}]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset$, draw($[\mathbf{x}]$, 'blue');
5	elseif $w([\mathbf{x}]) < \varepsilon$, {draw ($[\mathbf{x}]$, 'yellow')};
6	else bisect $[\mathbf{x}]$ and push into $\mathcal{L}$;
7	if $\mathcal{L} \neq \emptyset$, go to 2

If $\mathbf{\Delta}\mathbb{X}$ denotes the union of yellow boxes and if $\mathbb{X}^-$ is the union of red boxes then :

$$\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^- \cup \mathbf{\Delta}\mathbb{X}.$$

1.7 Image evaluation

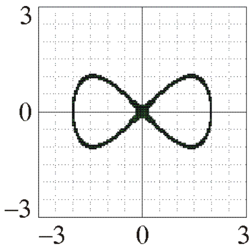
Define

$$\mathbf{f}(x_1, x_2) = \begin{pmatrix} (x_1 - 1)^2 - 1 + x_2 \\ -x_1^2 + (x_2 - 1)^2 \end{pmatrix},$$

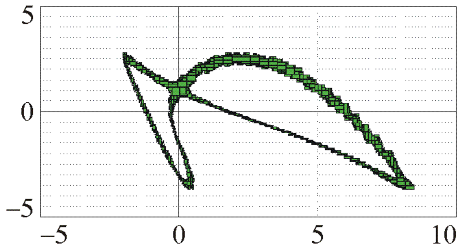
and

$$\mathbb{X}_1 = \left\{ (x_1, x_2) \in \mathbb{R}^2 \,\middle|\, x_1^4 - x_1^2 + 4x_2^2 \in [-0.1, 0.1] \right\}.$$

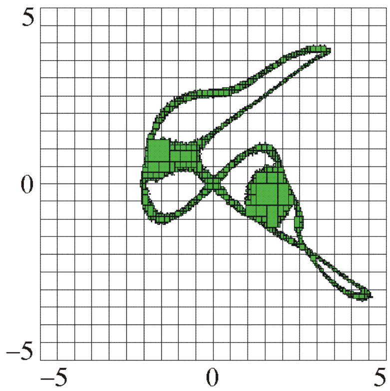
We shall compute $\mathbb{X}_1$, $\mathbf{f}(\mathbb{X}_1)$ and $\mathbf{f}^{-1} \circ \mathbf{f}(\mathbb{X}_1)$.



(a): $\mathbb{X}_1$



(b): $\mathbf{f}(\mathbb{X}_1)$



(c): $\mathbf{f}^{-1}(\mathbf{f}(\mathbb{X}_1))$

2 Applications of set computation

2.1 Bounded-error estimation

Model : $\phi(\mathbf{p}, t) = p_1 e^{-p_2 t}$.

Prior feasible box for the parameters : $[\mathbf{p}] \subset \mathbb{R}^2$

Measurement times : $t_1, t_2, \ldots, t_m$

Data bars : $[y_1^-, y_1^+], [y_2^-, y_2^+], \ldots, [y_m^-, y_m^+]$

$\mathbb{S} = \{\mathbf{p} \in [\mathbf{p}], \phi(\mathbf{p}, t_1) \in [y_1^-, y_1^+], \ldots, \phi(\mathbf{p}, t_m) \in [y_m^-, y_m^+]\}$.

If

$$\boldsymbol{\phi}(\mathbf{p}) = \begin{pmatrix} \phi(\mathbf{p}, t_1) \\ \\ \phi(\mathbf{p}, t_m) \end{pmatrix}$$

and

$$[\mathbf{y}] = [y_1^-, y_1^+] \times \cdots \times [y_m^-, y_m^+]$$

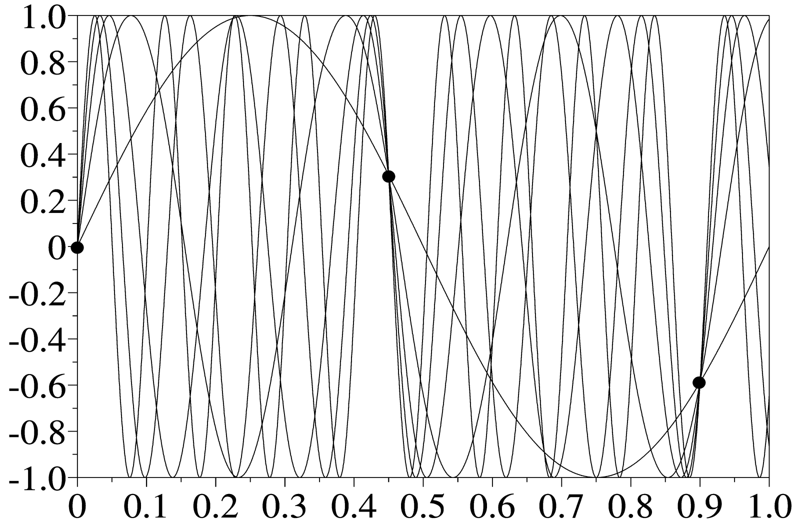
then

$$\mathbb{S} = [\mathbf{p}] \cap \boldsymbol{\phi}^{-1}([\mathbf{y}]) .$$

Show Setdemo (Guillaume Baffet), available at

`www.ensieta.fr/jaulin/demo.html`

If now $\phi(\mathbf{p}, t) = p_1 \sin(2\pi p_2 t)$ and $t_k = k\delta, \ldots$. $\mathbb{S}$ contains an infinite number of connected components.



2.2 Robust stability

The *stability domain* $\mathbb{S}_{\mathrm{p}}$ of the polynomial

$$P(s, \mathbf{p}) = s^n + a_{n-1}(\mathbf{p})s^{n-1} + \ldots + a_1(\mathbf{p})s + a_0(\mathbf{p})$$

is the set of all $\mathbf{p}$ such that $P(s, \mathbf{p})$ is stable.

If $P(s, \mathbf{p})$ is given by

$$s^3+(p_1+p_2+2)s^2+(p_1+p_2+2)s+2p_1p_2+6p_1+6p_2+2.25,$$

Its Routh table is given by

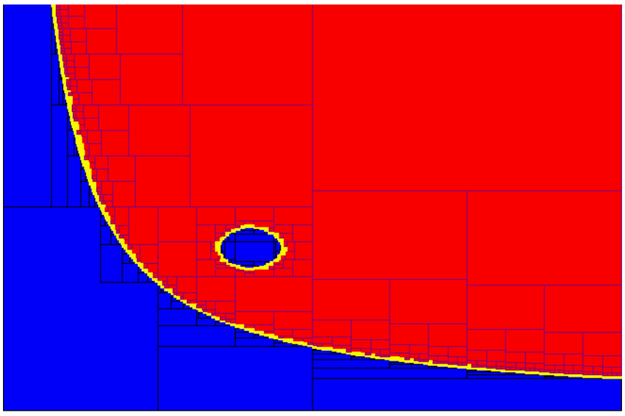
1	p_1+p_2+2
p_1+p_2+2	$2p_1p_2+6p_1+6p_2+2.25$
$\frac{(p_1-1)^2+(p_2-1)^2-0.25}{p_1+p_2+2}$	0
$2(p_1+3)(p_2+3)-15.75$	0

Its stability domain is thus defined by

$$\mathbb{S}_{\mathsf{p}} \triangleq \{\mathbf{p} \in \mathbb{R}^n \mid \mathbf{r}(\mathbf{p}) > \mathbf{0}\} = \mathbf{r}^{-1}\left(]0, +\infty[^{\times n}\right).$$

where

$$\mathbf{r}(\mathbf{p}) = \begin{pmatrix} p_1 + p_2 + 2 \\ (p_1 - 1)^2 + (p_2 - 1)^2 - 0.25 \\ 2(p_1 + 3)(p_2 + 3) - 15.75 \end{pmatrix}.$$



Stability domain $\mathbb{S}_p$ generated by Proj2d

2.3 Sailboat

State equations

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

In a cruising phase

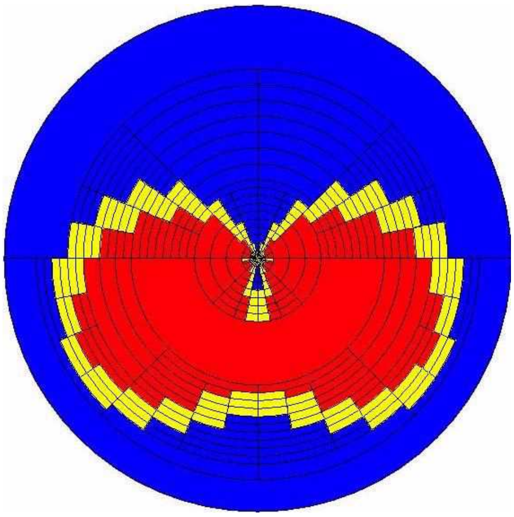
$$\dot{\theta} = 0, \dot{\delta}_s = 0, \dot{\delta}_r = 0, \dot{v} = 0, \dot{\omega} = 0.$$

i.e.,

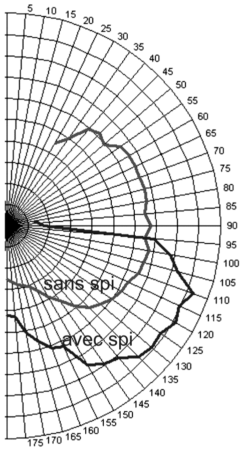
$$\begin{cases} 0 &= \omega \\ 0 &= u_1 \\ 0 &= u_2 \\ 0 &= f_s \sin \delta_s - f_r \sin \delta_r - v \\ 0 &= (1 - \cos \delta_s) f_s - \cos \delta_r . f_r - \omega \\ f_s &= \cos(\theta + \delta_s) - v \sin \delta_s \\ f_r &= v \sin \delta_r . \end{cases}$$

The polar diagram is

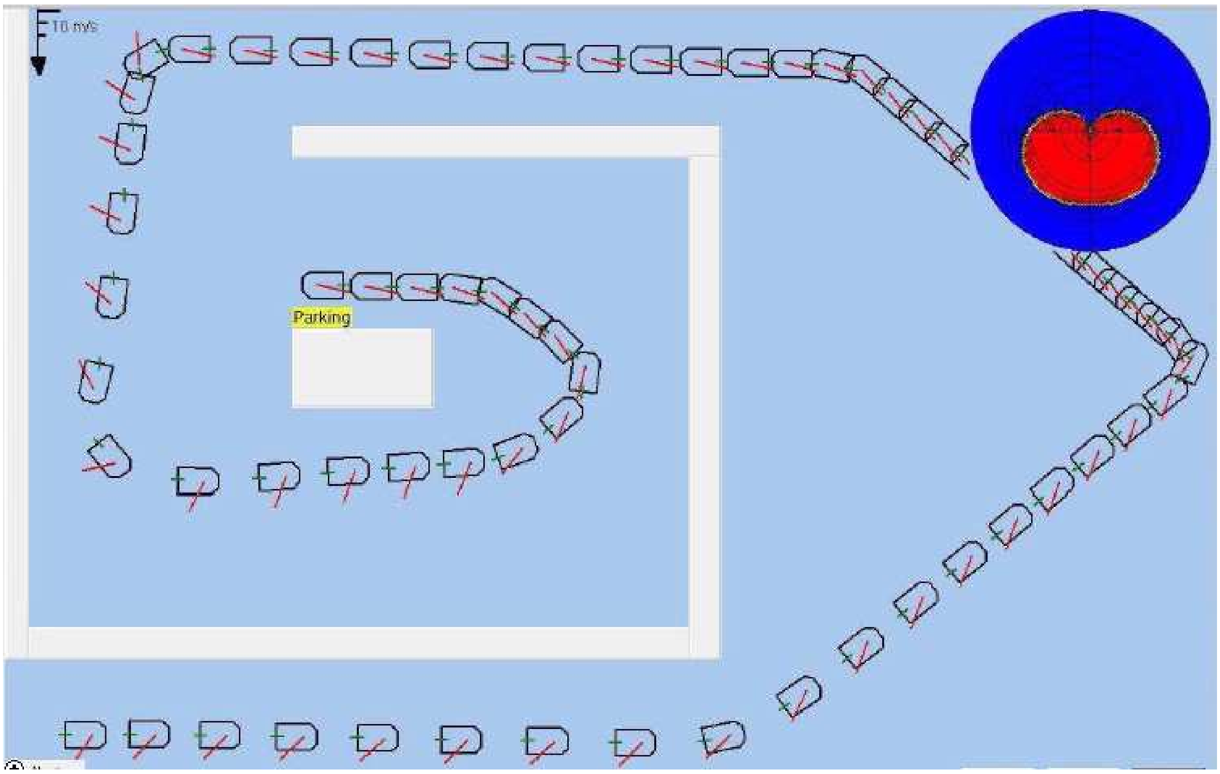
$$\begin{aligned}\mathbb{S}_y = \{(\theta, v) \quad & \mid \exists f_s, \delta_s, f_r, \delta_r, \\ & f_s \sin\delta_s - f_r \sin\delta_r - v = 0 \\ & (1 - \cos\delta_s)\, f_s - \cos\delta_r f_r = 0 \\ & f_s = \cos(\theta + \delta_s) - v \sin\delta_s \\ & f_r = v \sin\delta_r \ \}\end{aligned}$$



UBS

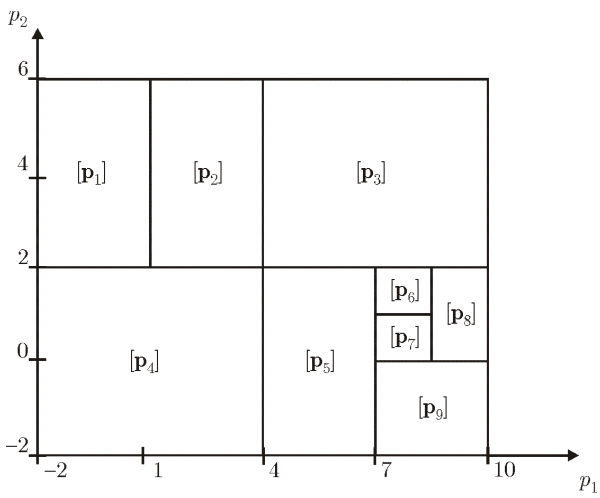


5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
sans spi
avec spi

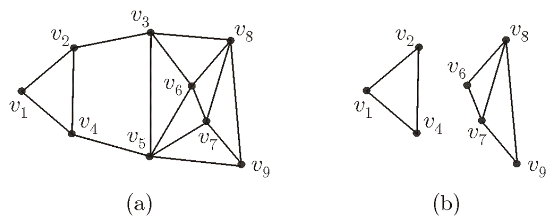

10 m/s
Parking

3 Interval and graphs

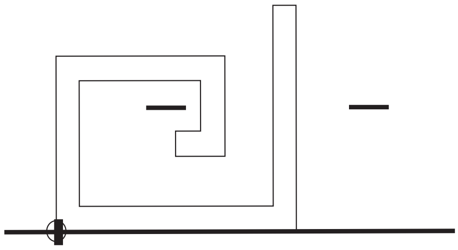
3.1 Path planning



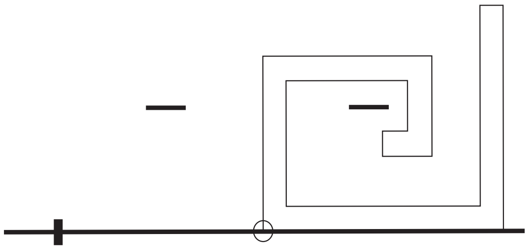
p_2
6
4
2
0
−2
$[\mathbf{p}_1]$
$[\mathbf{p}_2]$
$[\mathbf{p}_3]$
$[\mathbf{p}_4]$
$[\mathbf{p}_5]$
$[\mathbf{p}_6]$
$[\mathbf{p}_7]$
$[\mathbf{p}_8]$
$[\mathbf{p}_9]$
−2
1
4
7
10
p_1



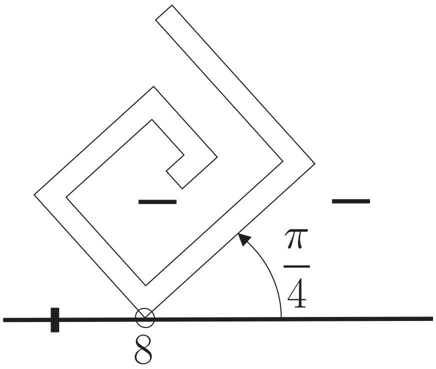
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
(a)
v_1
v_2
v_4
v_6
v_7
v_8
v_9
(b)



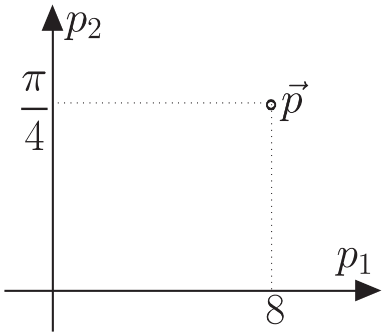
Initial configuration: $\vec{p} = (0\ 0)^{\mathrm{T}}$



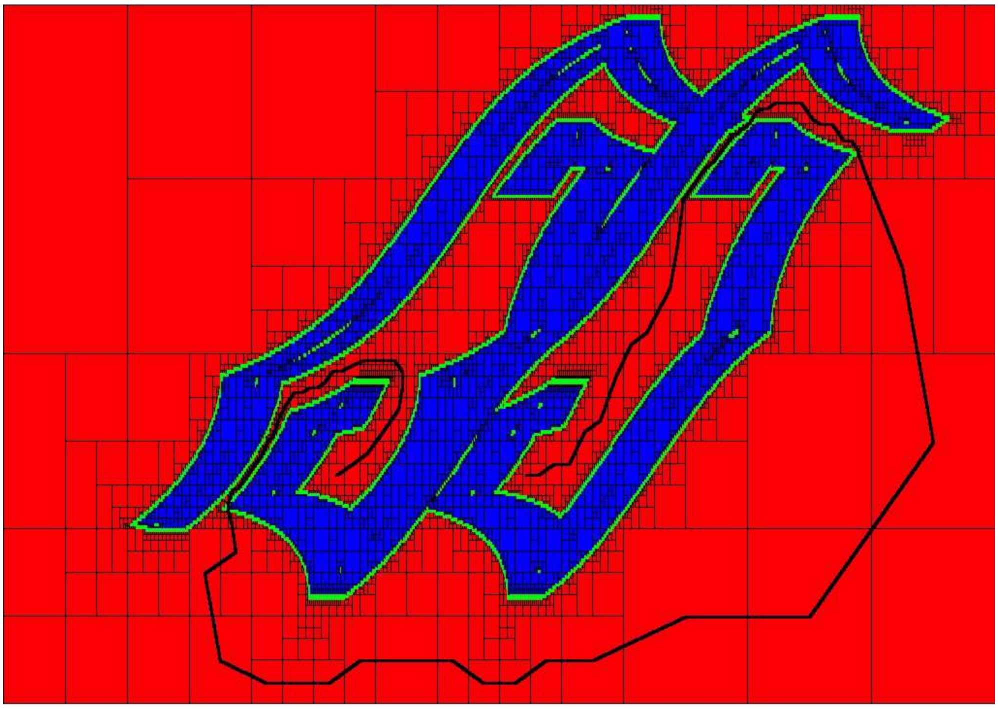
Goal configuration: $\vec{p} = (17\ 0)^{\mathrm{T}}$

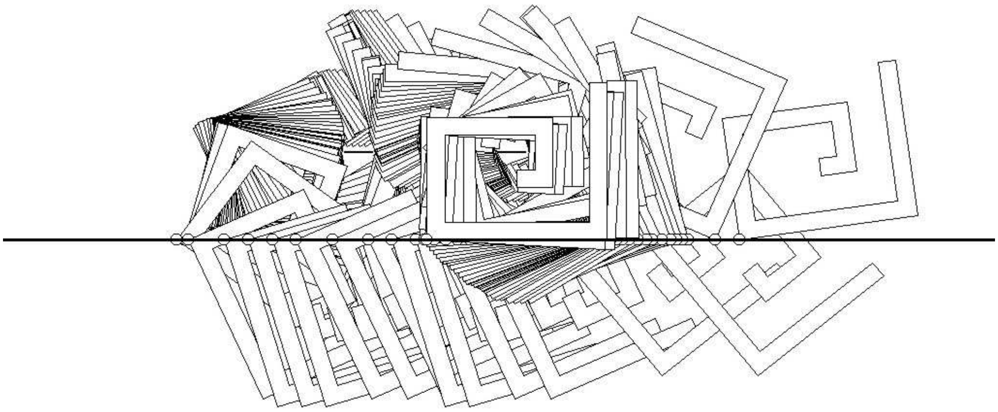


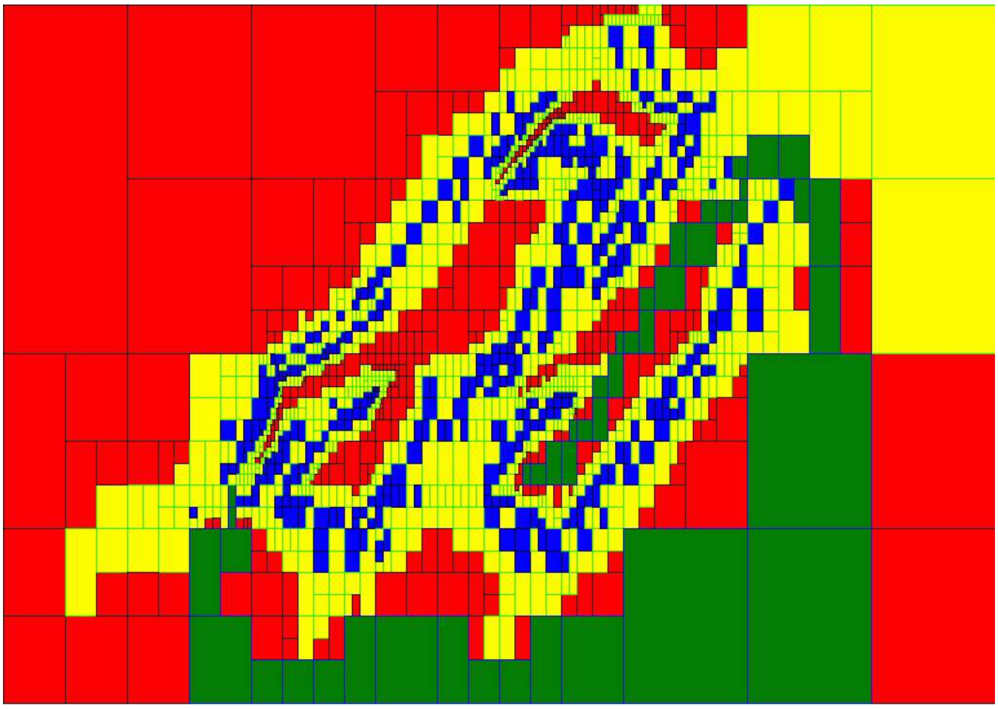
Room

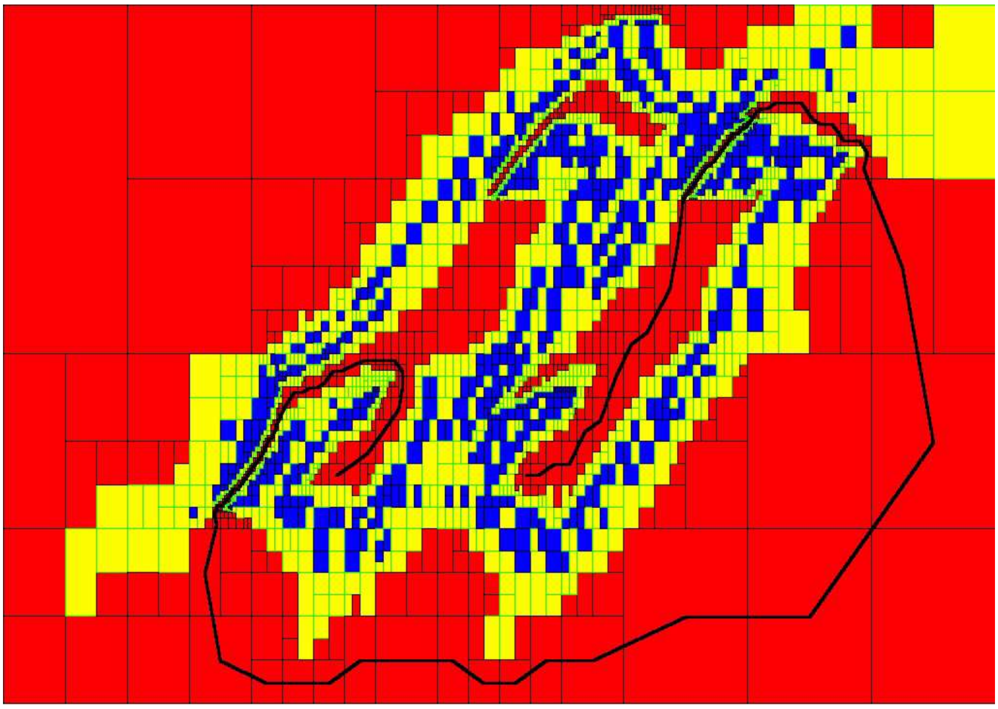


Configuration space









3.2 Counting connected components

(Collaboration with N. Delanoue and B. Cottenceau)

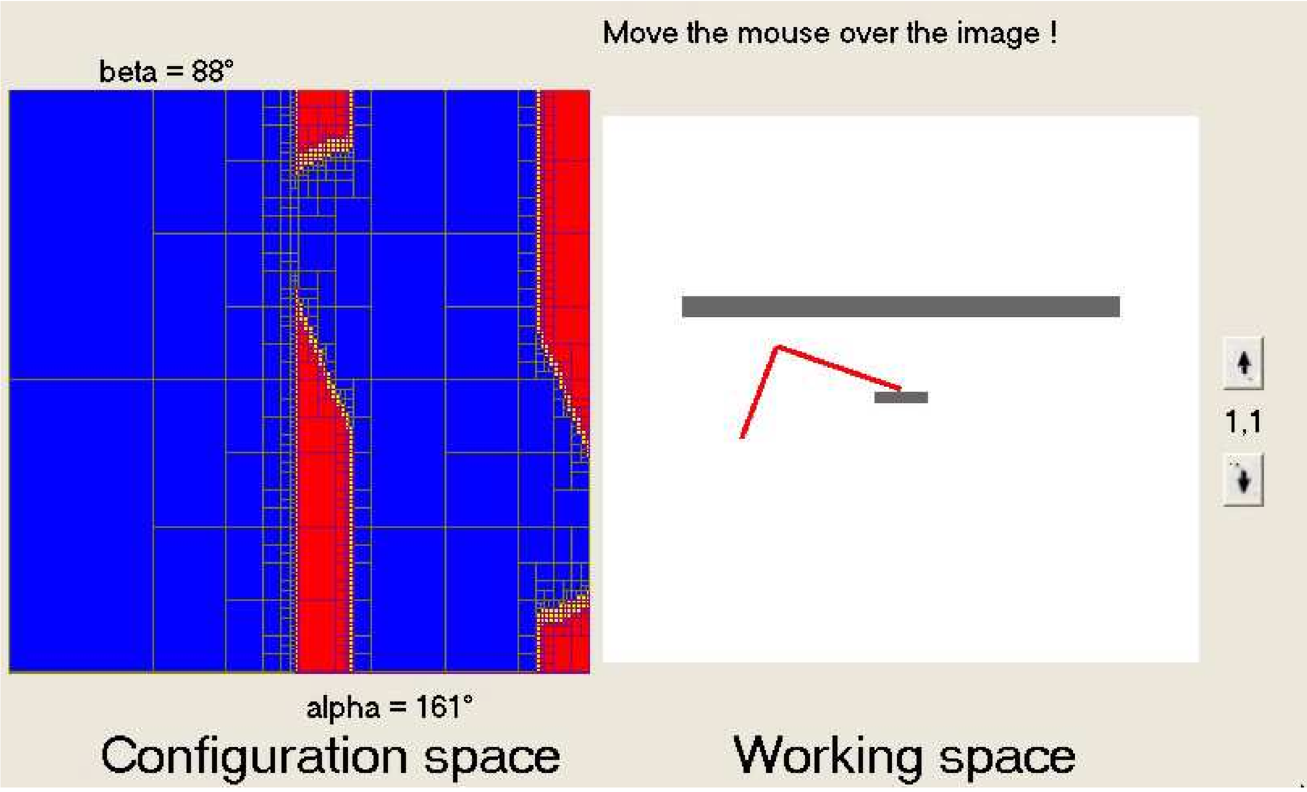


Figure 1:

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

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

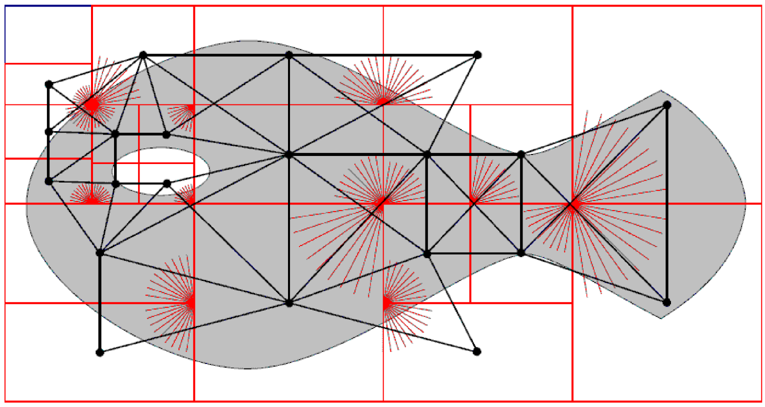
is *star-spangled graph* of the set $\mathbb{S}$ if

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

For instance, a star-spangled graph for the set

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

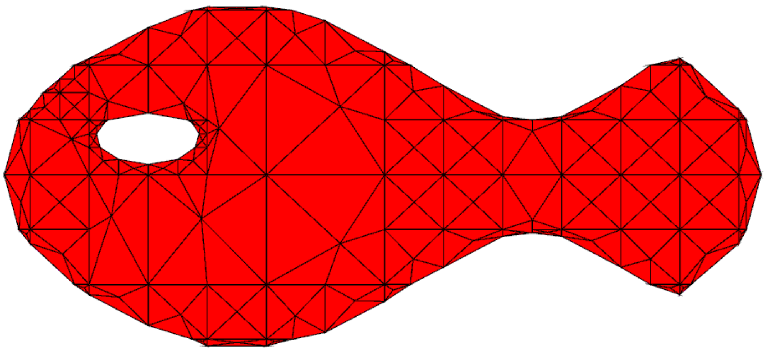
is



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 for all three constraints.

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 with N. Delanoue to compute a triangulation homeomorphic to $\mathbb{S}$.



3.3 Capture basin

(With M. Lhommeau and L. Hardouin)

$$\begin{cases} \dot{\mathbf{x}}(t) & = & \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{x}(0) & = & \mathbf{x}_0 \end{cases}$$

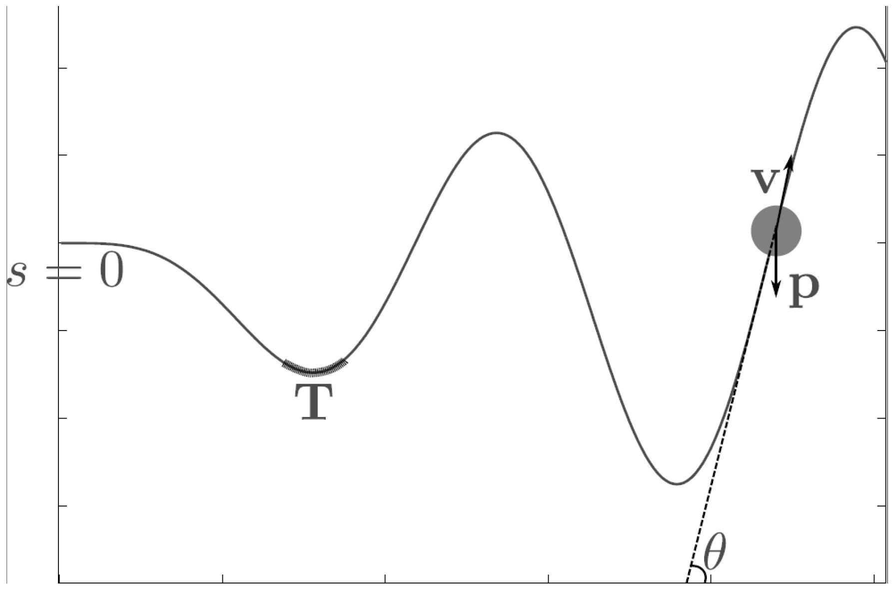
$\mathbf{u}(t) \in [\mathbf{u}] \in \mathbb{R}^m$ is the control, $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector.

Consider a rolling ball described by

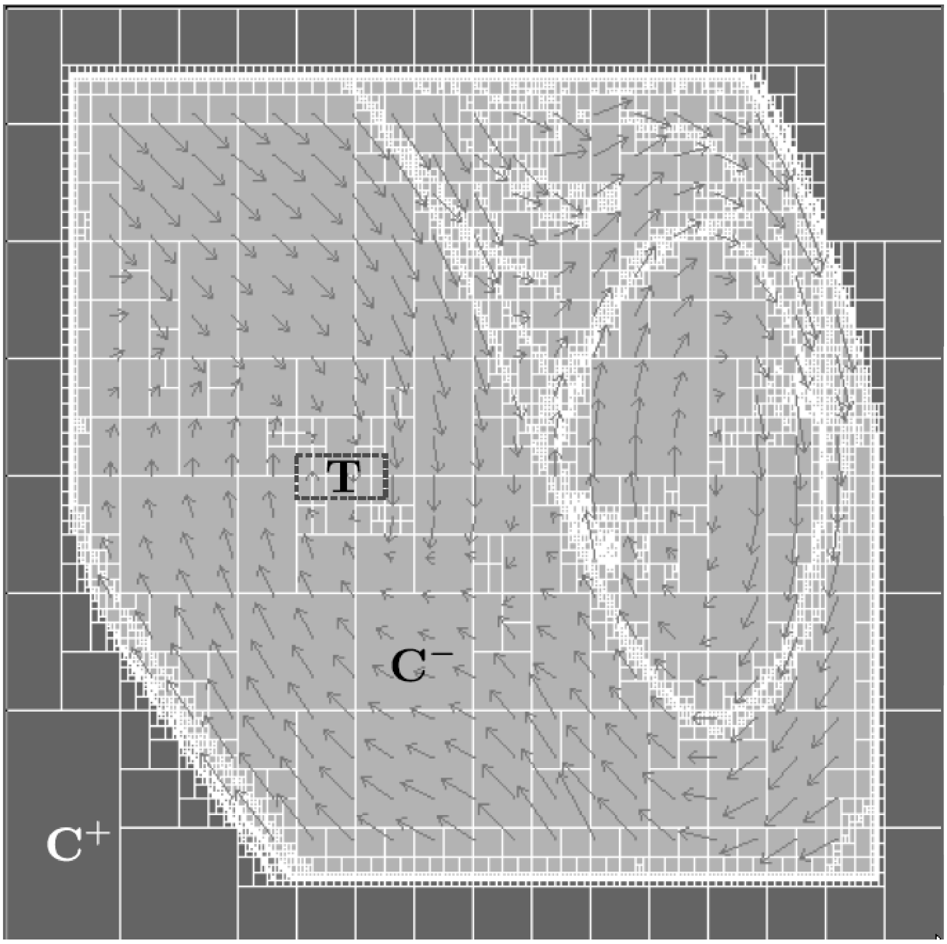
$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\sin\left(\theta(x_1)\right) - x_2 + u \end{cases} \tag{1}$$

where

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



$s = 0$
$\mathbf{T}$
$\mathbf{v}$
$\mathbf{p}$
θ



T
C^-
C^+

4 Contractors

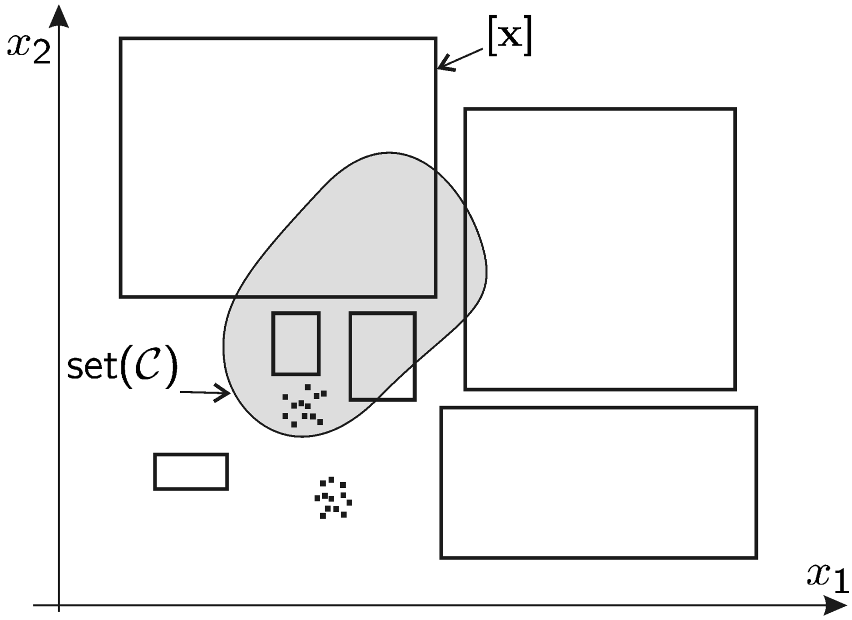
To characterize $\mathbb{X} \subset \mathbb{R}^n$, bisection algorithms bisect all boxes in all directions and become inefficient. Interval methods can still be useful if

- 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.

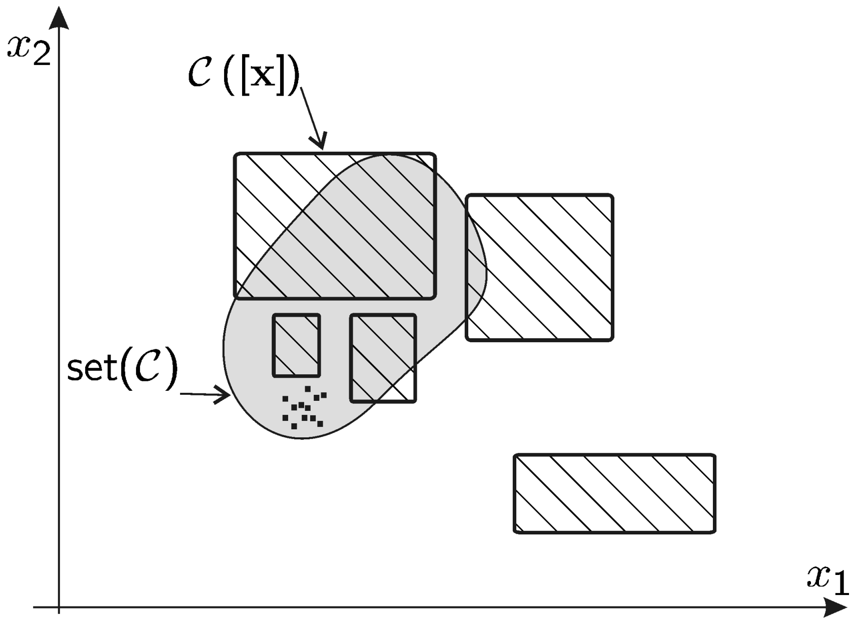
4.1 Definition

The operator $\mathcal{C}_{\mathbb{X}} : \mathbb{IR}^n \rightarrow \mathbb{IR}^n$ is a *contractor* for $\mathbb{X} \subset \mathbb{R}^n$ if

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



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	$\forall [\mathbf{x}] \in \mathbb{IR}^n,\ \mathcal{C}_{\mathbb{X}}([\mathbf{x}]) = [[\mathbf{x}] \cap \mathbb{X}]$
$\mathcal{C}_{\mathbb{X}}$ is *thin* if	$\forall \mathbf{x} \in \mathbb{R}^n,\ \mathcal{C}_{\mathbb{X}}(\{\mathbf{x}\}) = \{\mathbf{x}\} \cap \mathbb{X}$
$\mathcal{C}_{\mathbb{X}}$ is *idempotent* if	$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathcal{C}_{\mathbb{X}}\left(\mathcal{C}_{\mathbb{X}}([\mathbf{x}])\right) = \mathcal{C}_{\mathbb{X}}([\mathbf{x}])$.

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

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

4.2 Projection of constraints

Let x, y, z be 3 variables 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.

To *project* a constraint (here, $z = x + y$), is to compute the smallest intervals which contains all consistent values.

For our example, this amounts to project onto x, y and z the set

$$\mathbb{S} = \{(x, y, z) \in [-\infty, 5] \times [-\infty, 4] \times [6, \infty] \mid z = x + y\}\,.$$

4.3 Numerical method for projection

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

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

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 $[y] := [y] \cap ([z] * 1/[x])$.

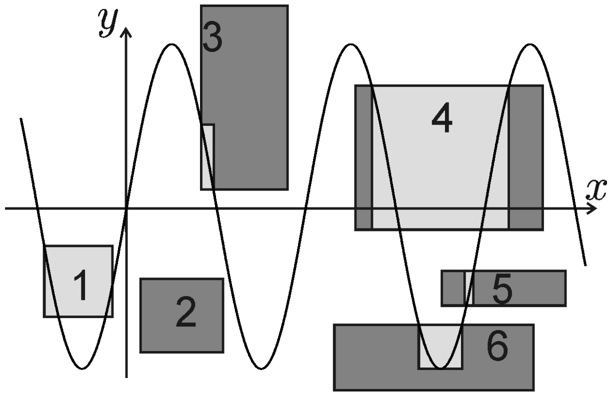
Consider the binary constraint

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

The associated contractor is

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

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



Projection of the sine constraint

4.4 Solvers

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:

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

until a steady box is reached.

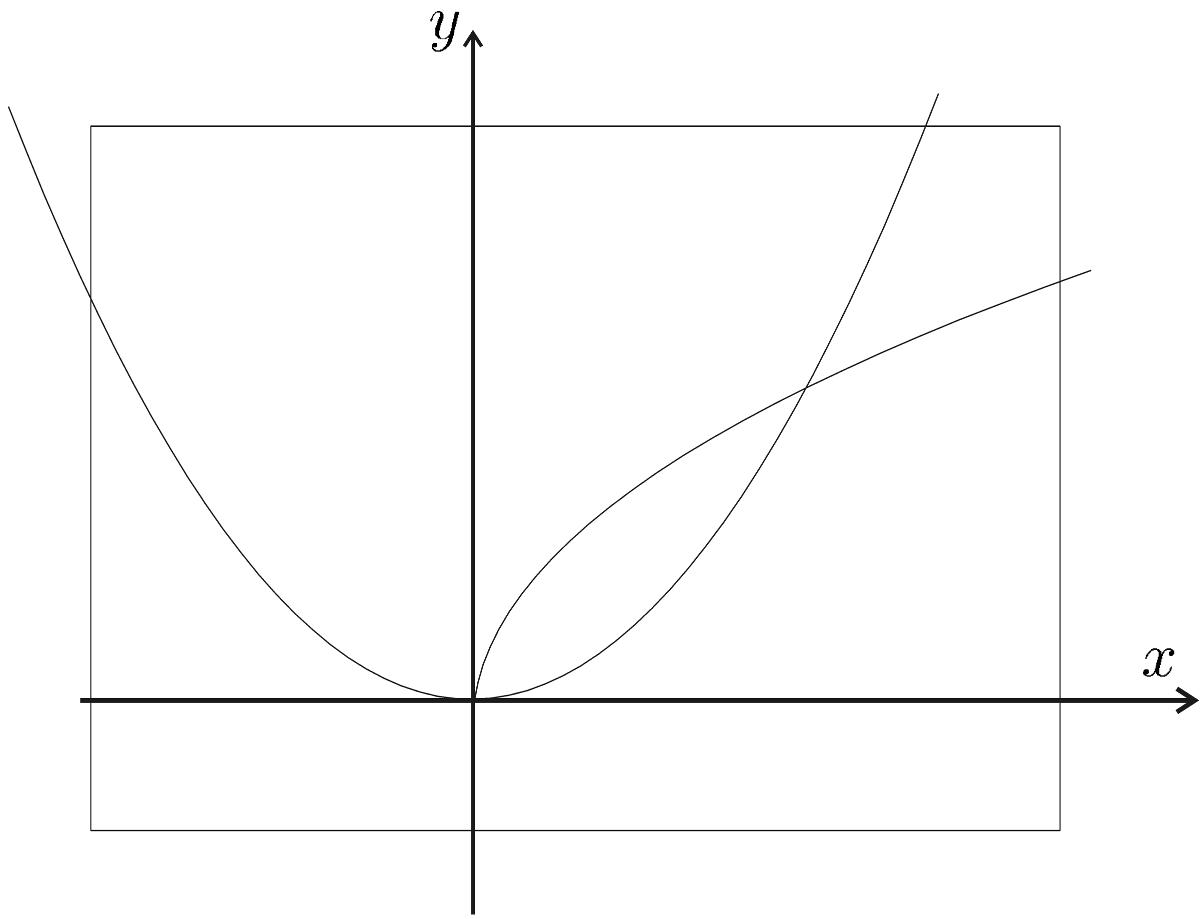
Example. Consider the system.

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

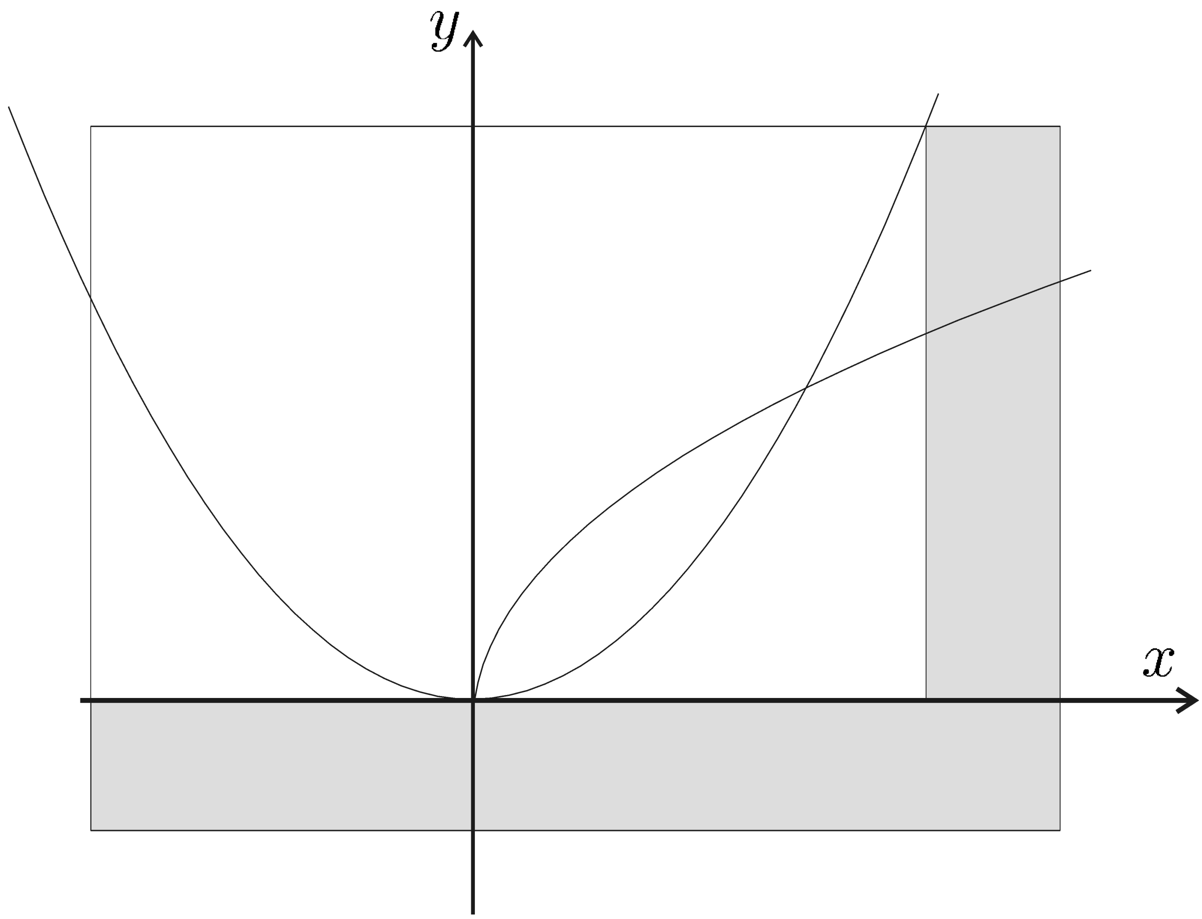
We build two contractors

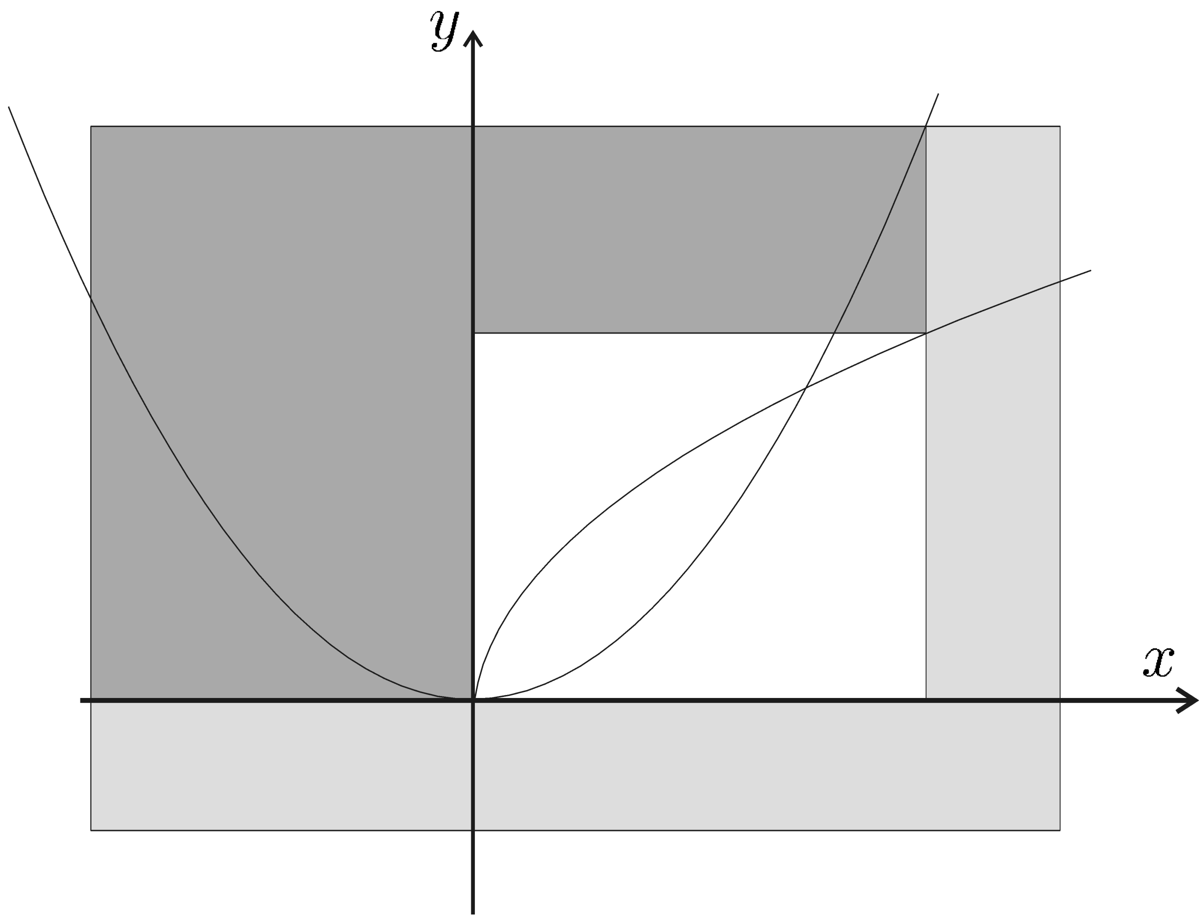
$$\mathcal{C}_1 : \left\{ \begin{array}{l} [y] = [y] \cap [x]^2 \\ [x] = [x] \cap \sqrt{[y]} \end{array} \right. \quad \text{associated to } y = x^2$$

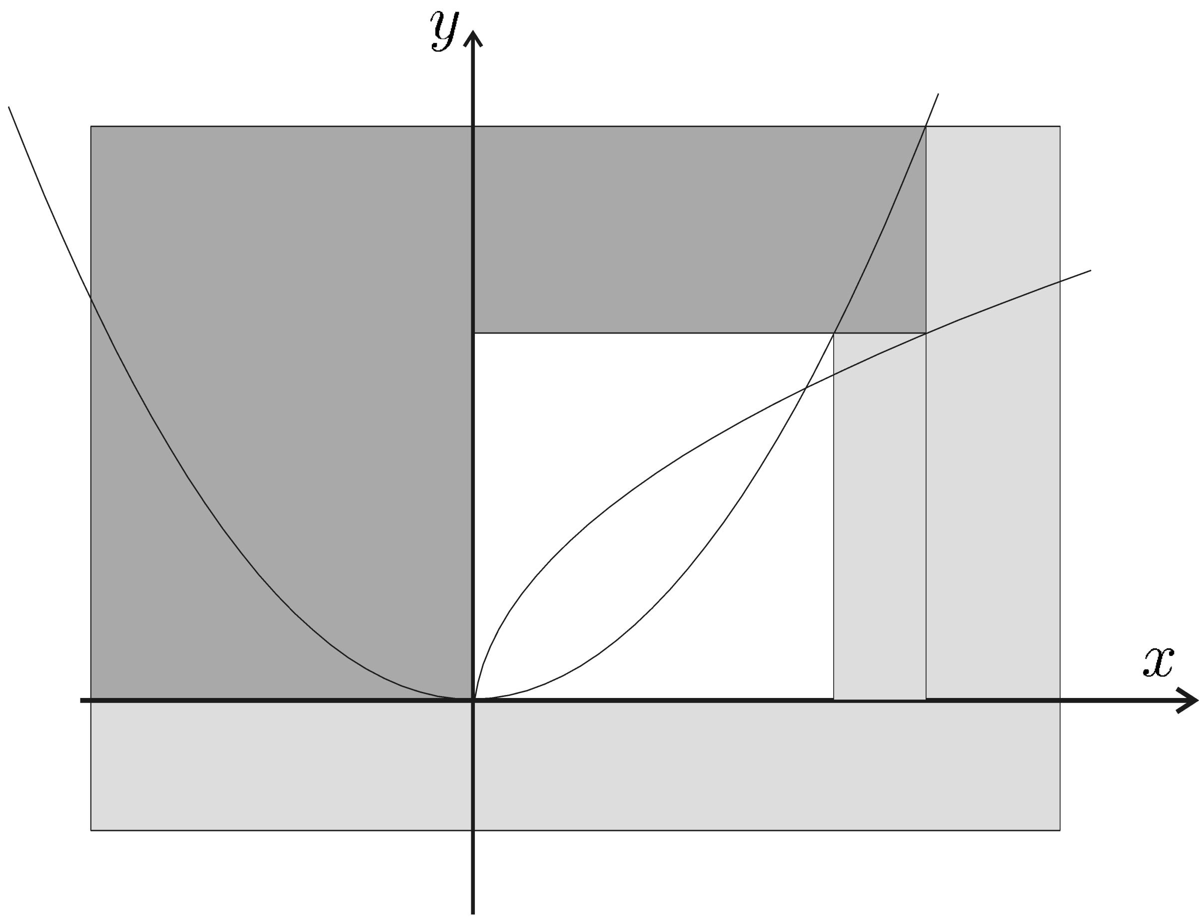
$$\mathcal{C}_2 : \left\{ \begin{array}{l} [y] = [y] \cap \sqrt{[x]} \\ [x] = [x] \cap [y]^2 \end{array} \right. \quad \text{associated to } y = \sqrt{x}$$

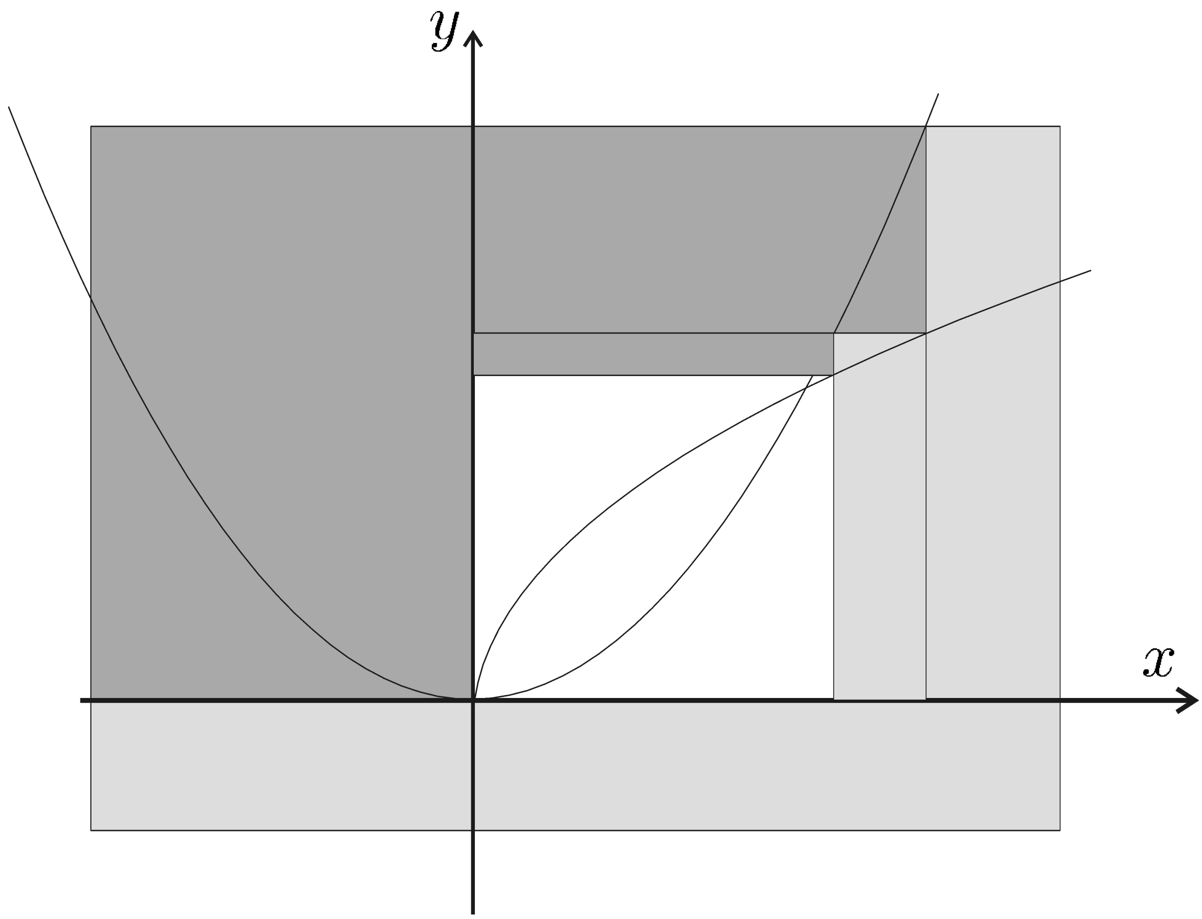


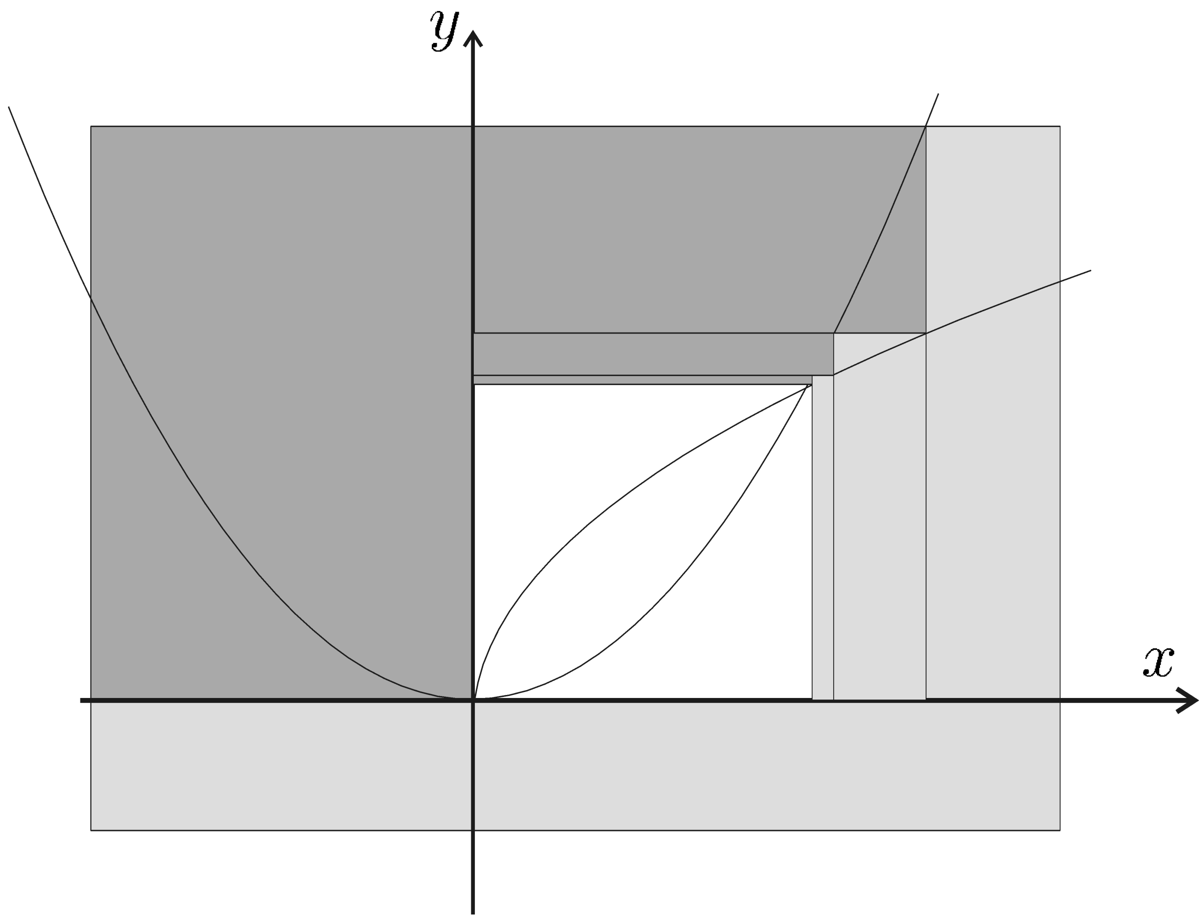
y
x

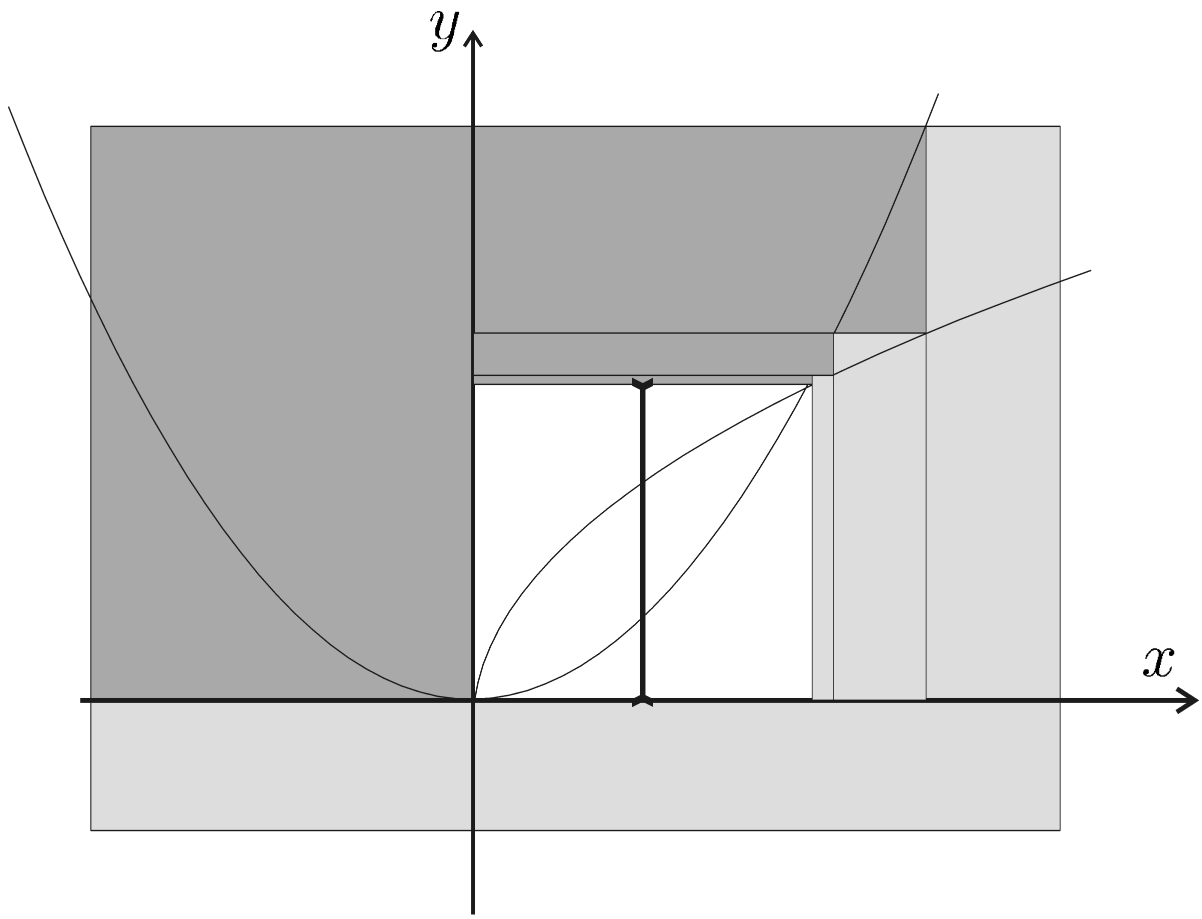

y
x

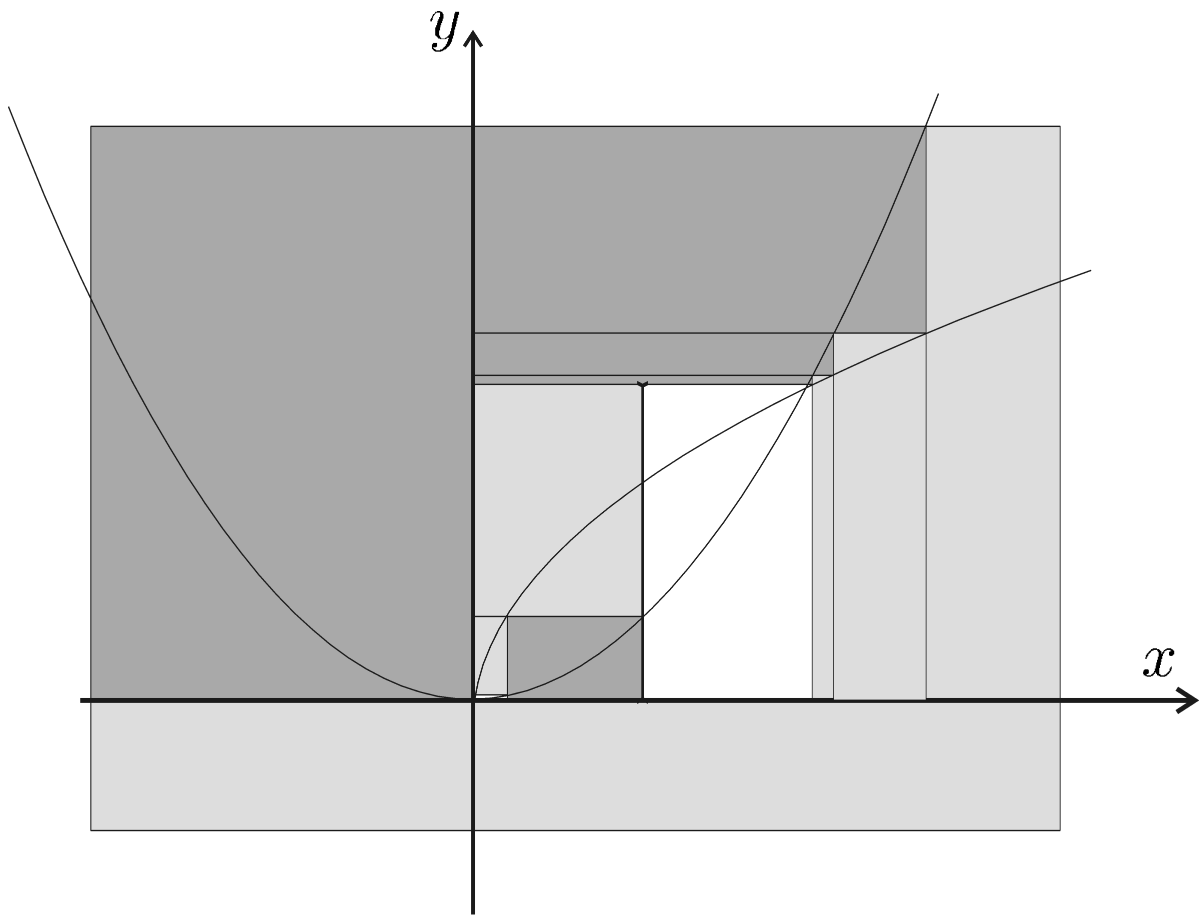

y
x


y
x

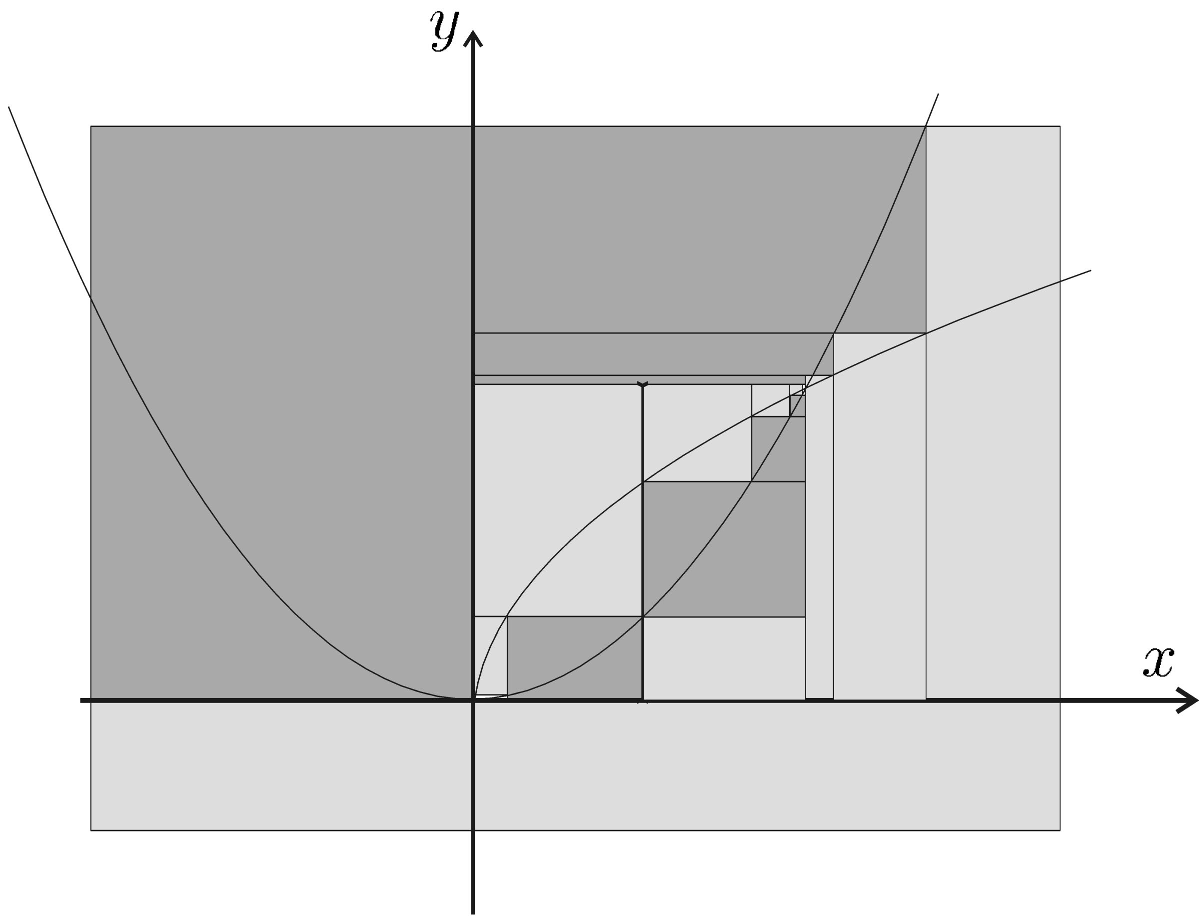

y
x


y
x


y
x

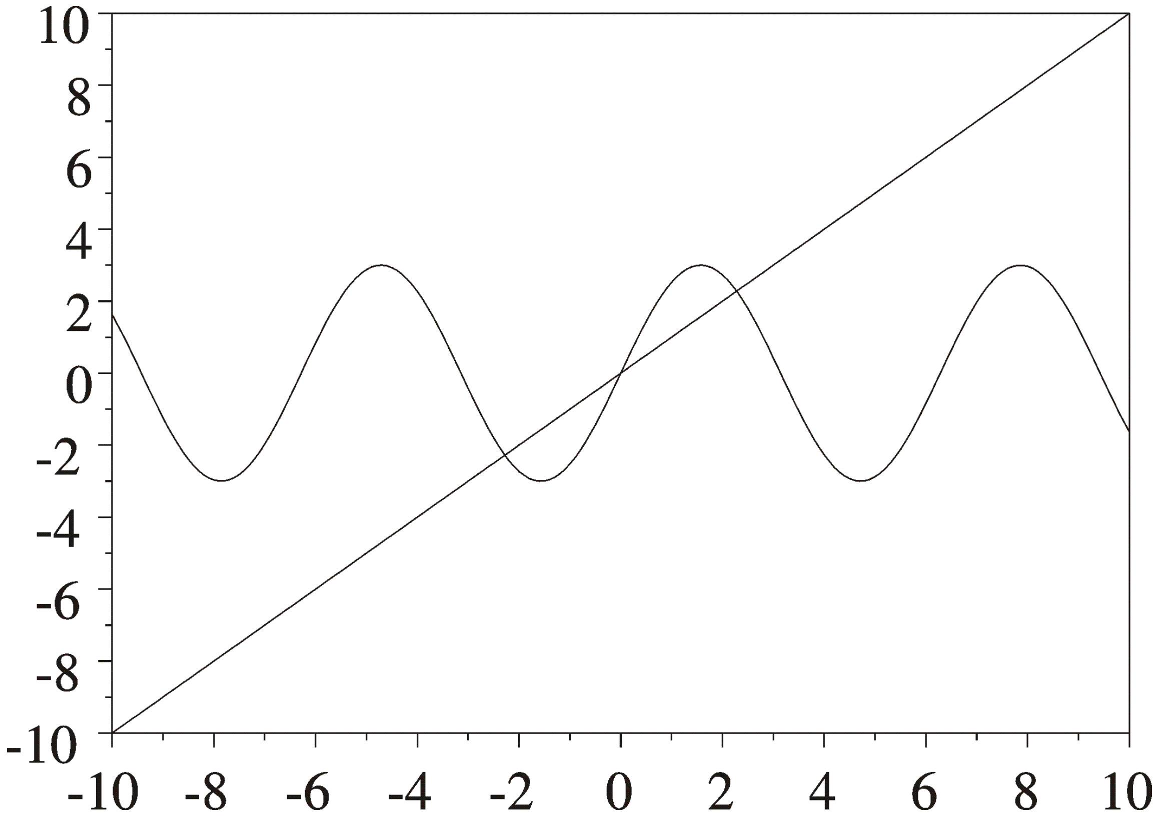


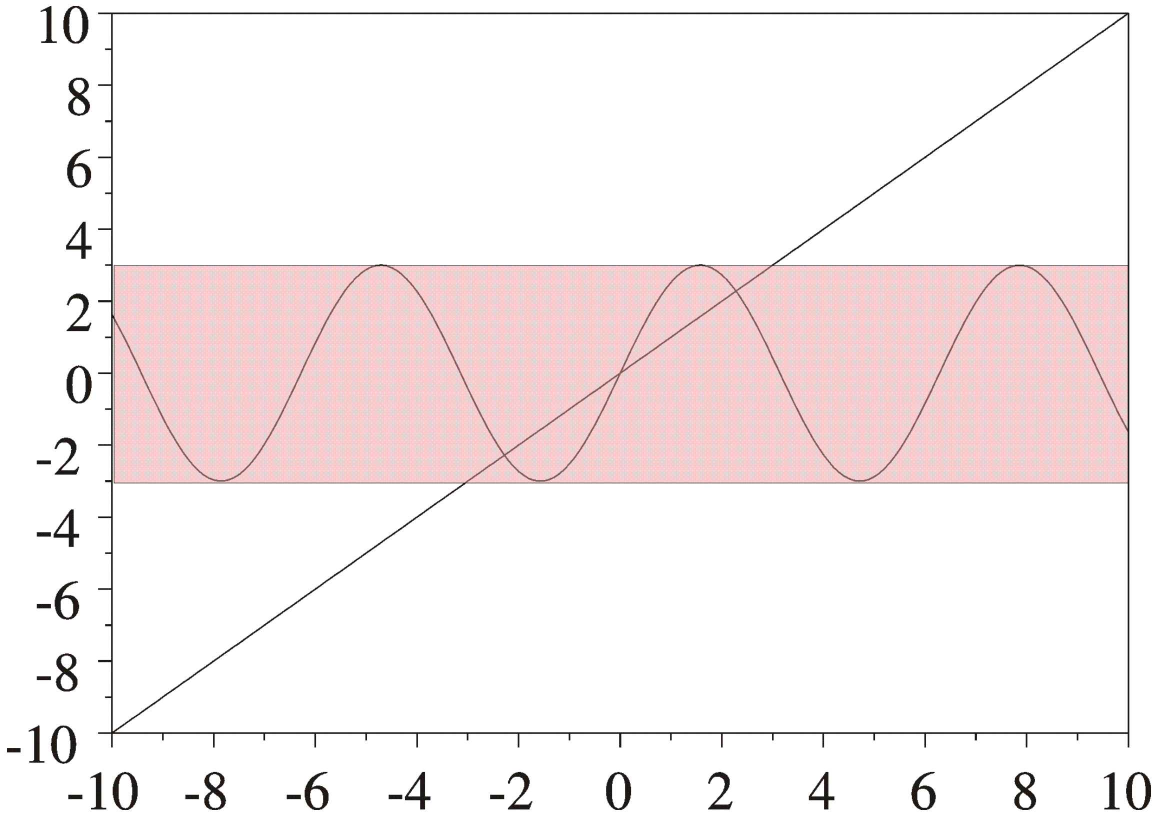
y
x

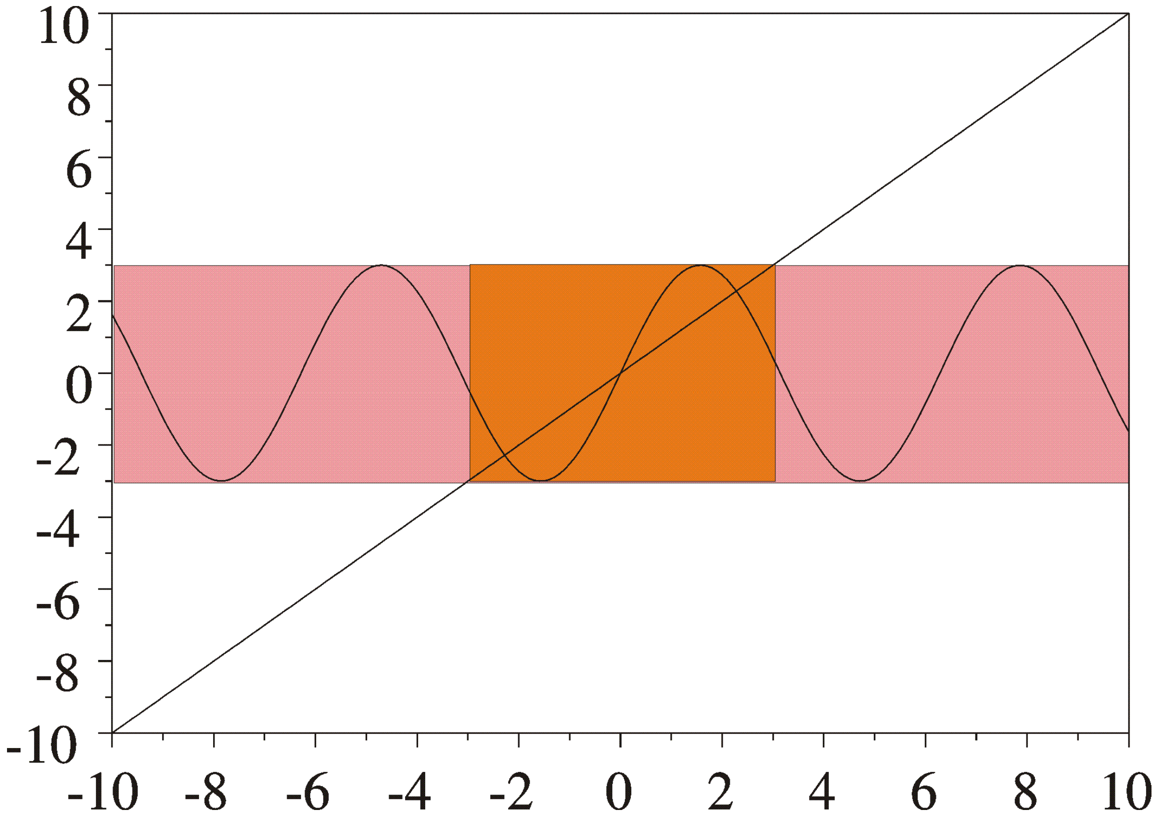

y
x

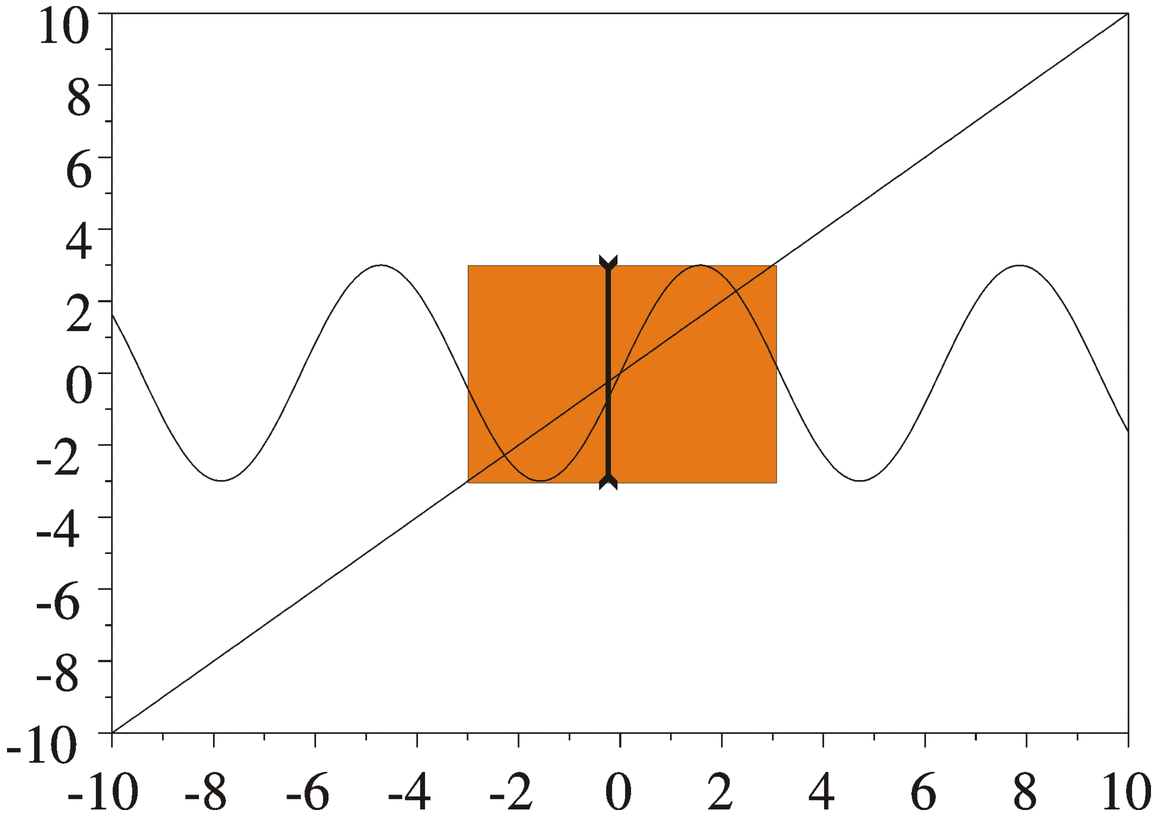
Exemple. Consider the system

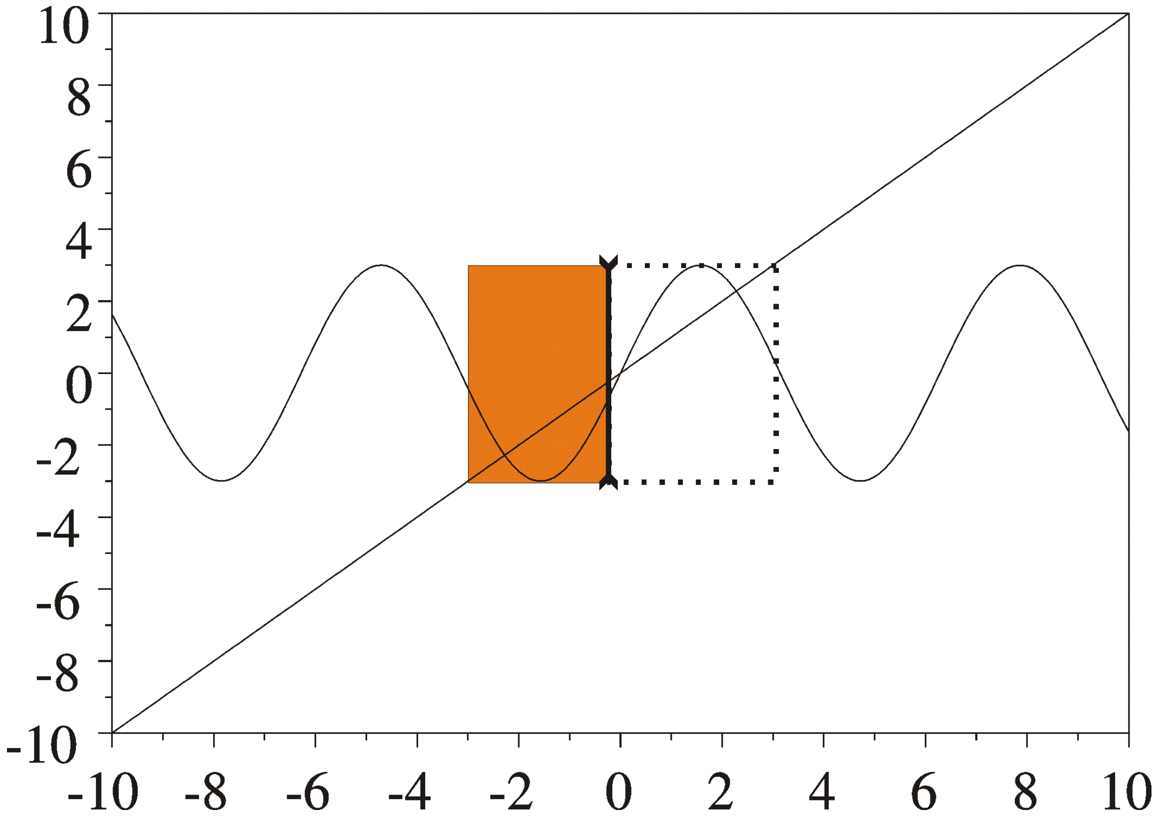
$$\left\{\begin{array}{lcl} y & = & 3\sin(x) \\ y & = & x \end{array}\right. \qquad x \in \mathbb{R},\ y \in \mathbb{R}.$$


10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

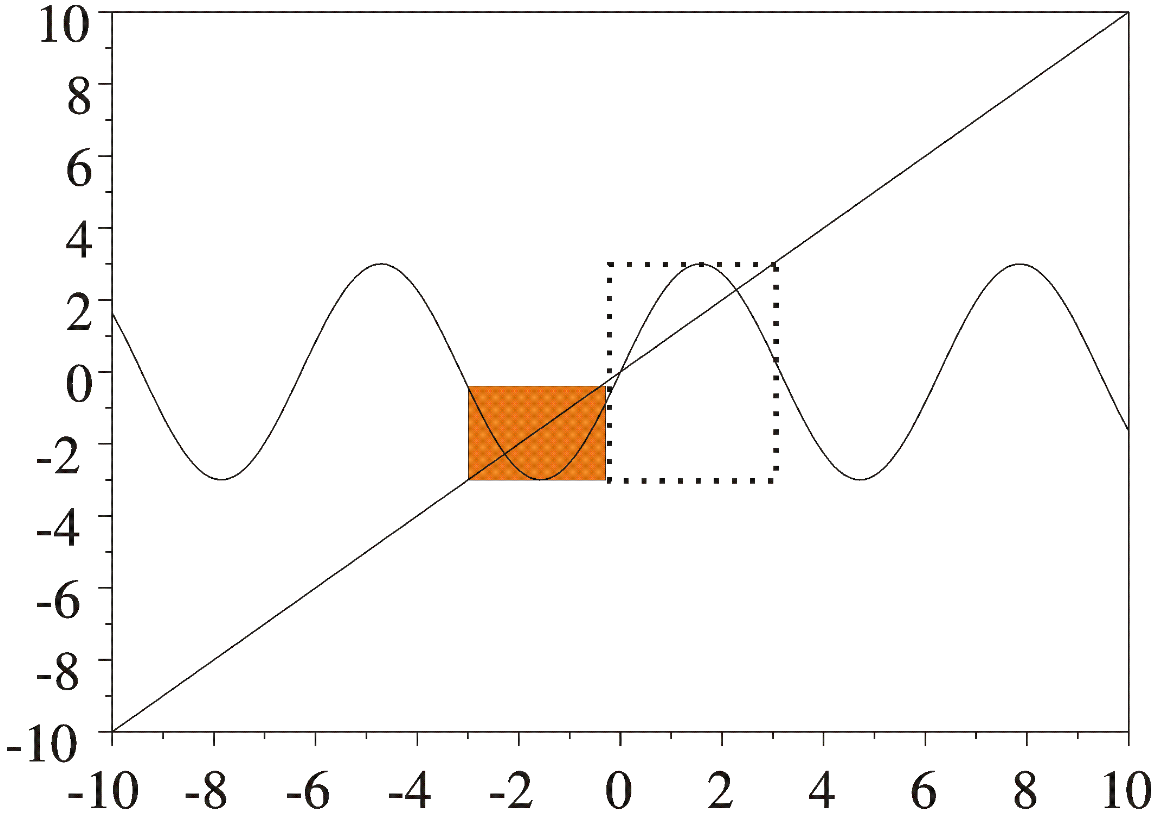

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

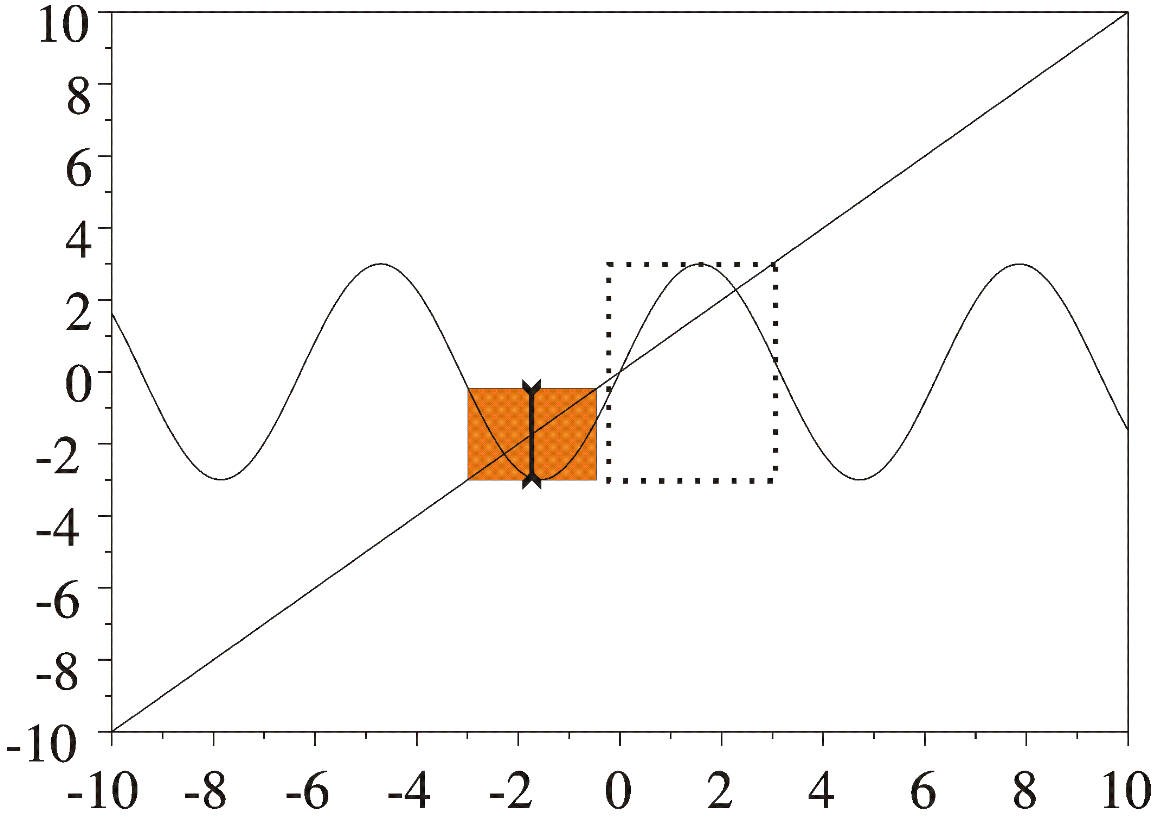

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

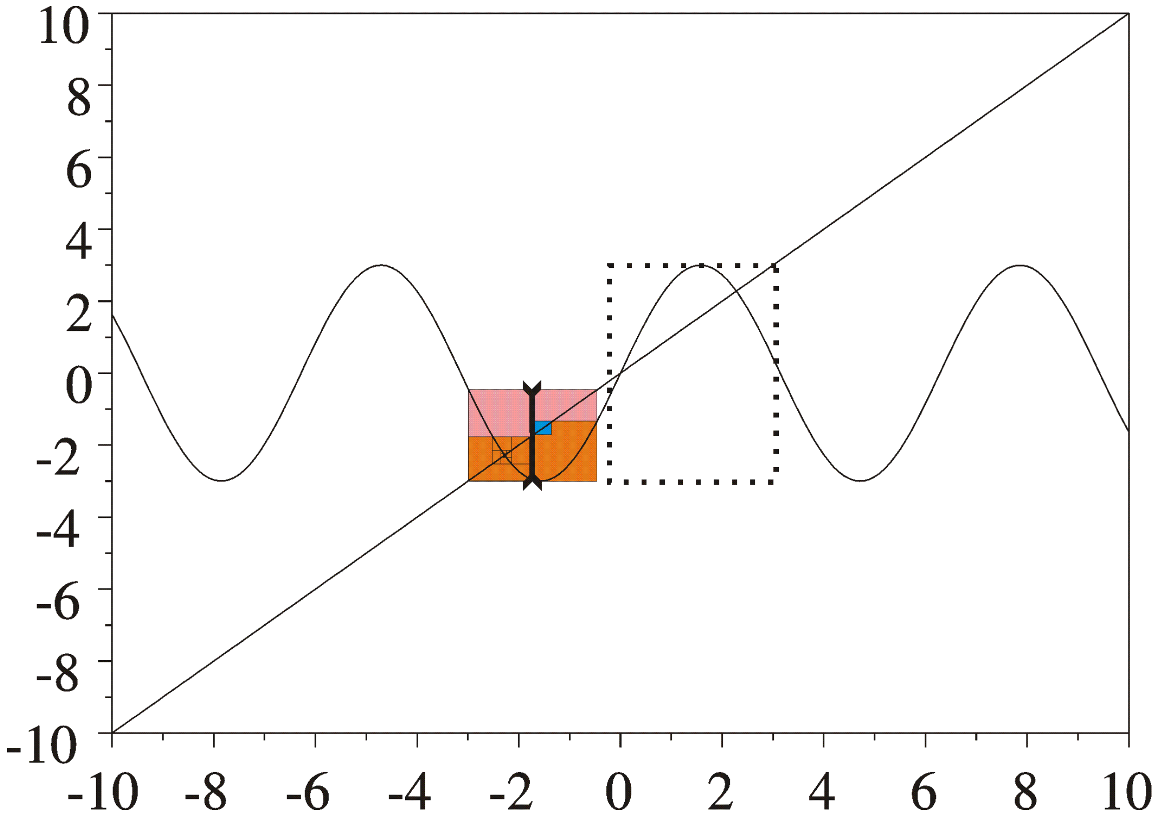

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10



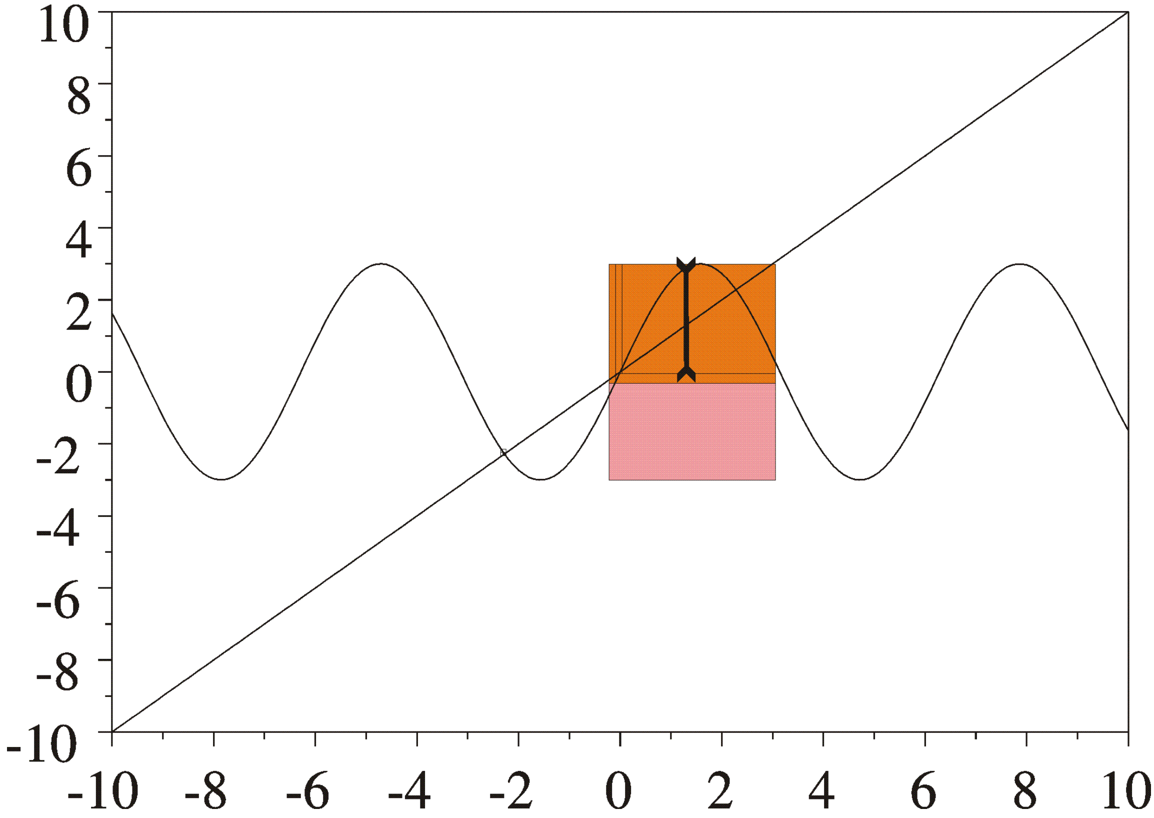
10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10


10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

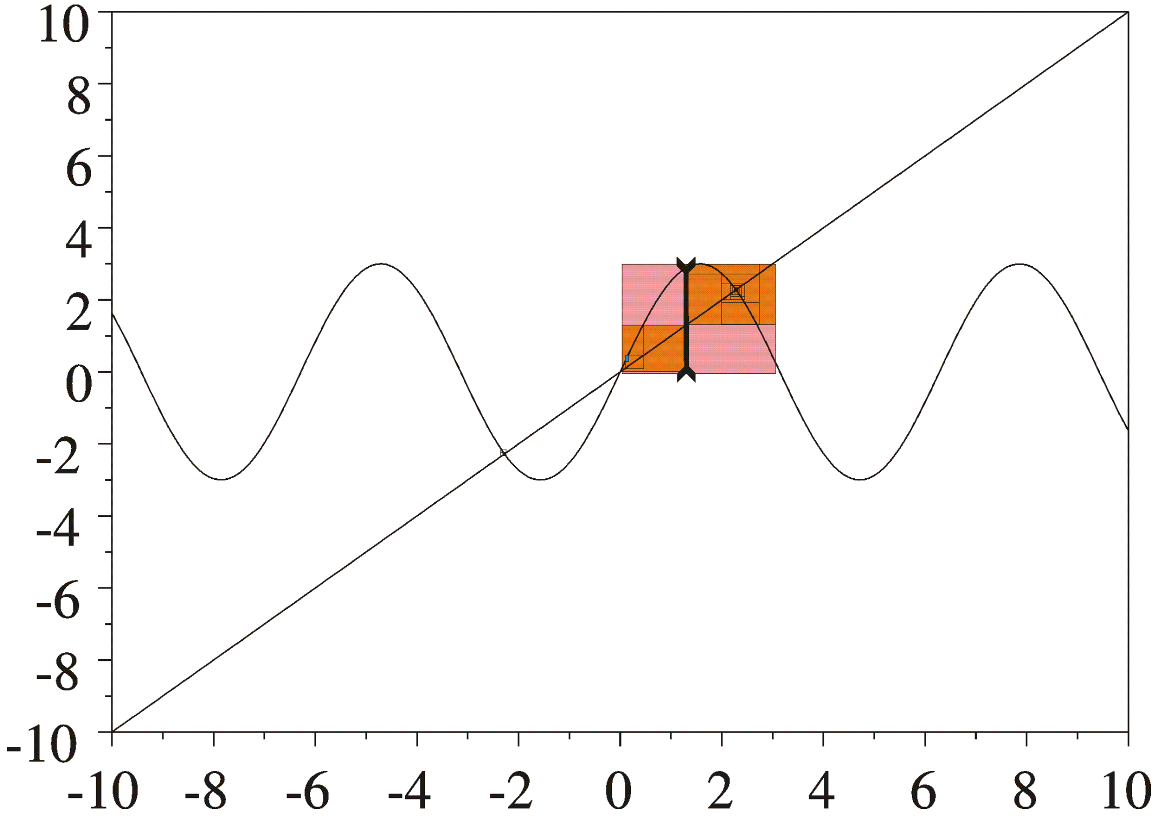

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10



10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10



10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10


10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

4.5 Decomposition into primitive constraints

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

can be decomposed into

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

4.6 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.

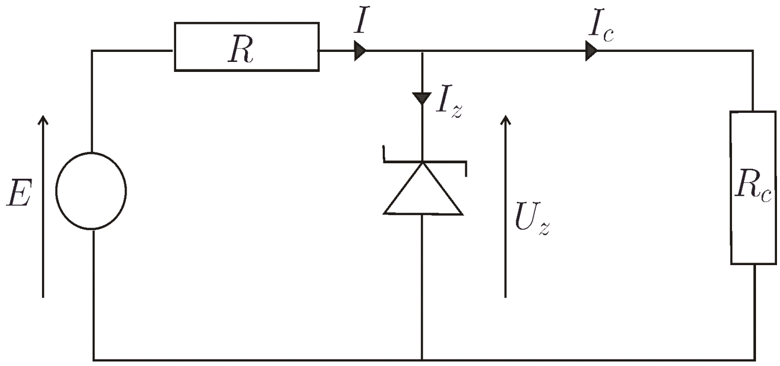
A Quimper program is a set of complementary contractors.

Quimper returns m subpavings, where m is the number of contractors

It is available at

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

4.7 Bounded-error estimation



R
I
I_c
I_z
E
U_z
R_c

It is known that

$$U_z \in [6,7]V,\ r \in [7,8]\Omega,\ U_0 \in [6,6.2]V$$
$$R \in [100,110]\Omega,\ E \in [18,20]V,\ I_z \in [0,\infty]A$$
$$I \in]-\infty,\infty[A,\ I_c \in]-\infty,\infty[A, R_c \in [50,60]\Omega.$$

The constraints are

Zener diode	$I_z = \max(0, \frac{U_z - U_0}{r})$,
Ohm rule	$U_z = R_c I_c$,
Current rule	$I = I_c + I_z$,
Voltage rule	$E = RI + U_z$.

IntervalPeeler contracts the domains into:

$$U_z \in [6,007;6,518], r \in [7,8]\Omega,$$
$$U_0 \in [6,6.2]V, R \in [100,110]\Omega,$$
$$E \in [18,20]V, I_z \in [0.,0.398]A$$
$$I \in [0.11;0.14]A,\ \ I_c \in [0.1;0,13]A,$$
$$R_c \in [50,60]\Omega$$

4.8 Robust stability

A CSP is *infallible* if any arbitrary instantiation of the variables is a solution.

Consider the CSP

$$
\begin{aligned}
\mathcal{V} &= \{x, y\} \\
\mathcal{D} &= \{[x],[y]\} \\
\mathcal{C} &= \{\ f(x,y) \leq 0,\ g(x,y) \leq 0\}\,.
\end{aligned}
$$

The CSP is infallible if

$$
\begin{aligned}
& \forall x \in [x], \forall y \in [y],\ f(x,y) \leq 0 \text{ and } g(x,y) \leq 0, \\
\Leftrightarrow\ & \{(x,y) \in [x] \times [y] \mid f(x,y) > 0 \text{ or } g(x,y) > 0\} = \emptyset \\
\Leftrightarrow\ & \{(x,y) \in [x] \times [y] \mid \max\left(f(x,y), g(x,y)\right) > 0\} = \emptyset.
\end{aligned}
$$

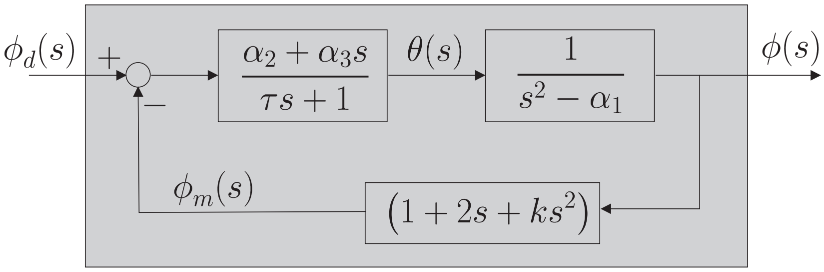
Consider a motorbike with a speed of 1m/s.

Angle of the handlebars: θ.

Rolling angle: ϕ

Wanted rolling angle: ϕ_d

Measured rolling angle: ϕ_m.



The input-output relation of the closed-loop system is :

$$\phi(s) = \frac{\alpha_2 + \alpha_3 s}{\left(s^2 - \alpha_1\right)(\tau s + 1) + (\alpha_2 + \alpha_3 s)\left(1 + 2s + ks^2\right)}\phi_d(s).$$

Its characteristic polynomial is thus

$$\begin{aligned} P(s) &= \left(s^2 - \alpha_1\right)(\tau s + 1) + (\alpha_2 + \alpha_3 s)\left(1 + 2s + ks^2\right) \\ &= a_3 s^3 + a_2 s^2 + a_1 s + a_0, \end{aligned}$$

with

$$\begin{array}{cc} a_3 = \tau + \alpha_3 k & a_2 = \alpha_2 k + 2\alpha_3 + 1 \\ a_1 = \alpha_3 - \alpha_1 \tau + 2\alpha_2 & a_0 = -\alpha_1 + \alpha_2. \end{array}$$

The Routh table is :

a_3	a_1
a_2	a_0
$\frac{a_2a_1-a_3a_0}{a_2}$	0
a_0	0

The closed-loop system is stable if $a_3, a_2, \frac{a_2a_1-a_3a_0}{a_2}$ and a_0 have the same sign.

Assume that it is known that

$$\begin{array}{ll} \alpha_1 \in [8.8; 9.2] & \alpha_2 \in [2.8; 3.2] \\ \alpha_3 \in [0.8; 1.2] & \tau \in [1.8; 2.2] \\ k \in [-3.2; -2.8]. & \end{array}$$

The system is robustly stable if,

$\forall \alpha_1 \in [\alpha_1], \forall \alpha_2 \in [\alpha_2], \forall \alpha_3 \in [\alpha_3], \forall \tau \in [\tau], \forall k \in [k],$
$a_3,\ a_2,\ \frac{a_2 a_1 - a_3 a_0}{a_2}$ and a_0 have the same sign.

Now, we have the equivalence

$$\begin{array}{rl} & b_1,\ b_2,\ b_3 \text{ and } b_4 \text{ have the same sign} \\ \Leftrightarrow & \max\left(\min\left(b_1, b_2, b_3, b_4\right), -\max\left(b_1, b_2, b_3, b_4\right)\right) > 0 \end{array}$$

The robust stability condition amounts to proving that

$$\begin{aligned} \exists\alpha_1 \in\ & [\alpha_1], \exists\alpha_2 \in [\alpha_2], \exists\alpha_3 \in [\alpha_3], \exists\tau \in [\tau], \exists k \in [k], \\ & \max\Big(\min\left(a_3, a_2, \tfrac{a_2 a_1 - a_3 a_0}{a_2}, a_0\right), \\ & \qquad -\max(a_3, a_2, \tfrac{a_2 a_1 - a_3 a_0}{a_2}, a_0) \Big) \le 0 \end{aligned}$$

is false,. . .

i.e., that the CSP

$$
\begin{array}{rcl}
\mathcal{V} & = & \{a_0, a_1, a_2, a_3, \alpha_1, \alpha_2, \alpha_3, \tau, k\}, \\
\mathcal{D} & = & \{[\alpha_0], [\alpha_1], [\alpha_2], [\alpha_3], [\alpha_2], [\alpha_3], [\tau], [k]\}, \\
\mathcal{C} & = & \left\{ \begin{array}{l}
a_3 = \tau + \alpha_3 k \ ; \ a_2 = \alpha_2 k + 2\alpha_3 + 1 \ ; \\
a_1 = \alpha_3 - \alpha_1 \tau + 2\alpha_2, \\
a_0 = -\alpha_1 + \alpha_2 \ ; \\
m_1 = \min\left(a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}, a_0\right) ; \\
m_2 = \max\left(a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}, a_0\right) \\
\max\left(m_1, -m_2\right) \leq 0.
\end{array} \right\}
\end{array}
$$

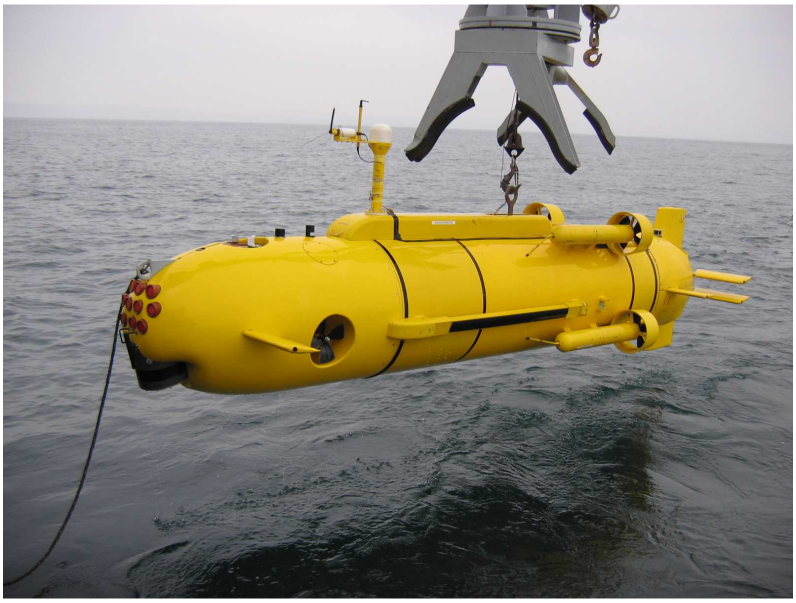
has no solution.

With QUIMPER.

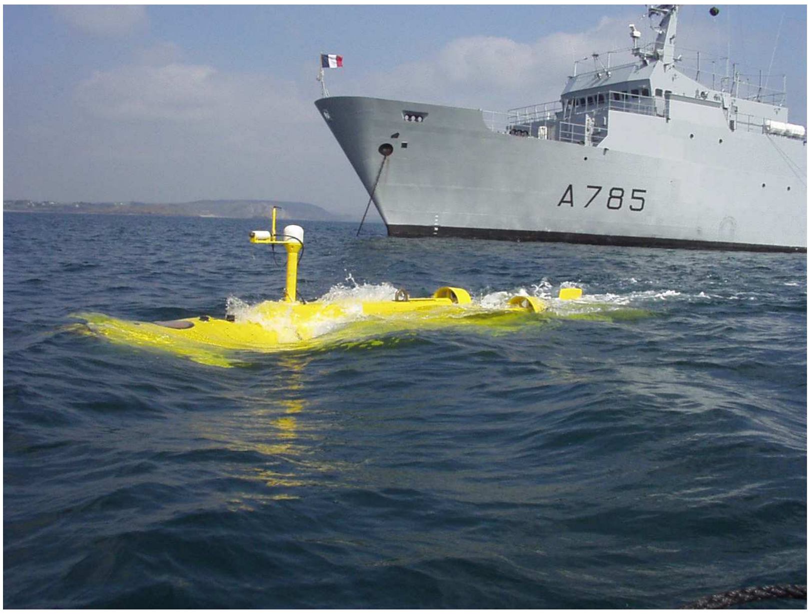
```
variables
 alpha1 in [8.8,9.2];
 alpha2 in [2.8,3.2];
 alpha3 in [0.8,1.2];
 tau in [1.8,2.2];
 k in [-3.2,-2.8];
 r in [-1e08,0];
 b1 in [-1e08,0];
 b2 in [0,-1e08];
 a3,a2,a1,a0,b;
```

```
contractor_list L
 a3=tau+alpha3*k;
 a2=alpha2*k+2*alpha3+1;
 a1=alpha3-alpha1*tau+2*alpha2;
 a0=alpha2-alpha1;
 b1=min(a3,a2,(a2*a1-a3*a0)/a2,a0);
 b2=max(a3,a2,(a2*a1-a3*a0)/a2,a0);
end
contractor C
   compose(L)
end
```

5 Redermor



The *Redermor*, GESMA



The *Redermor* at the surface

Show simulation

Why choosing an interval constraint approach for SLAM ?

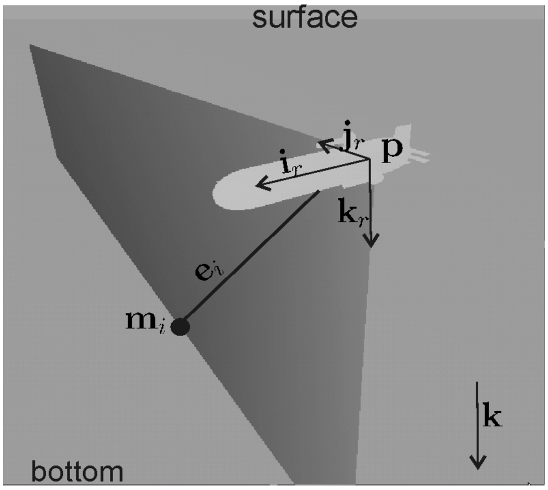
1) A reliable method is needed.
2) The model is nonlinear.
3) The pdf of the noises are unknown.
4) Reliable error bounds are provided by the sensors.
5) A huge number of redundant data are available.

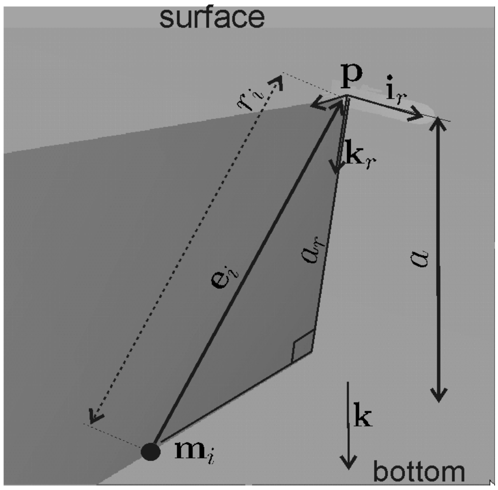
5.1 Sensors

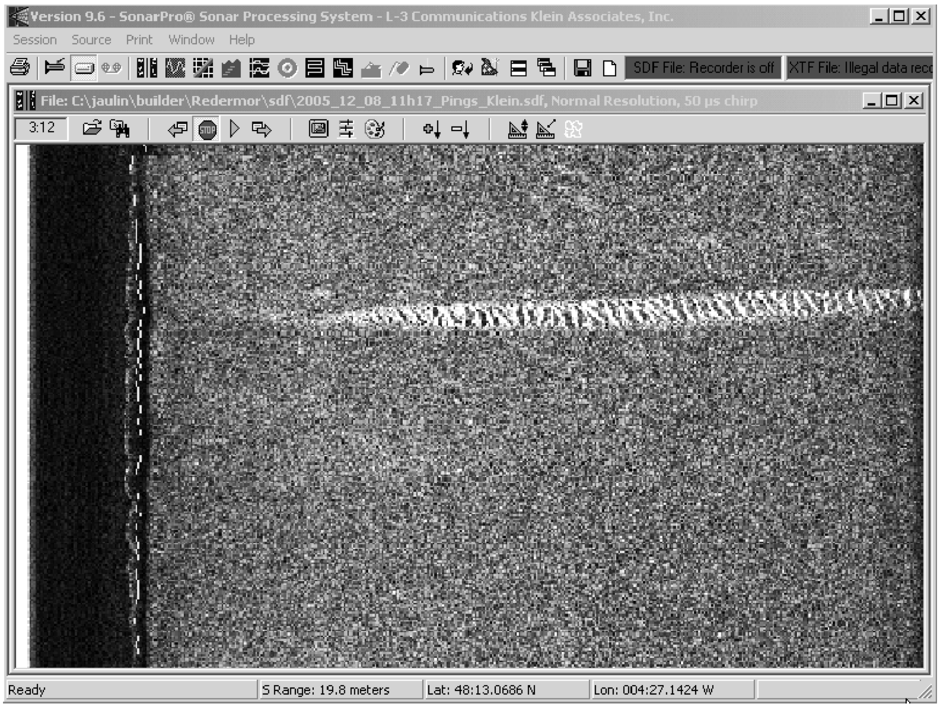
A GPS (Global positioning system) at the surface only.

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

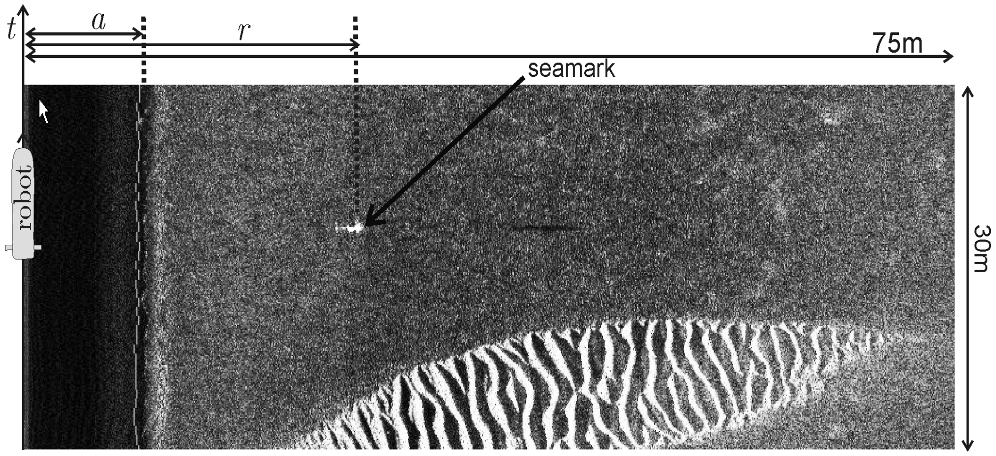
A sonar (KLEIN 5400 side scan sonar). Gives the distance r between the robot to the detected object.







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.

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

5.2 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
$\tau(i)$	7054	7092	7374	7748	9038	9688
$\sigma(i)$	1	2	1	0	1	5
$\tilde{r}(i)$	52.42	12.47	54.40	52.68	27.73	26.98

6	7	8	9	10	11
10024	10817	11172	11232	11279	11688
4	3	3	4	5	1
37.90	36.71	37.37	31.03	33.51	15.05

5.3 Constraints satisfaction problem

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

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

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

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

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

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

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

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

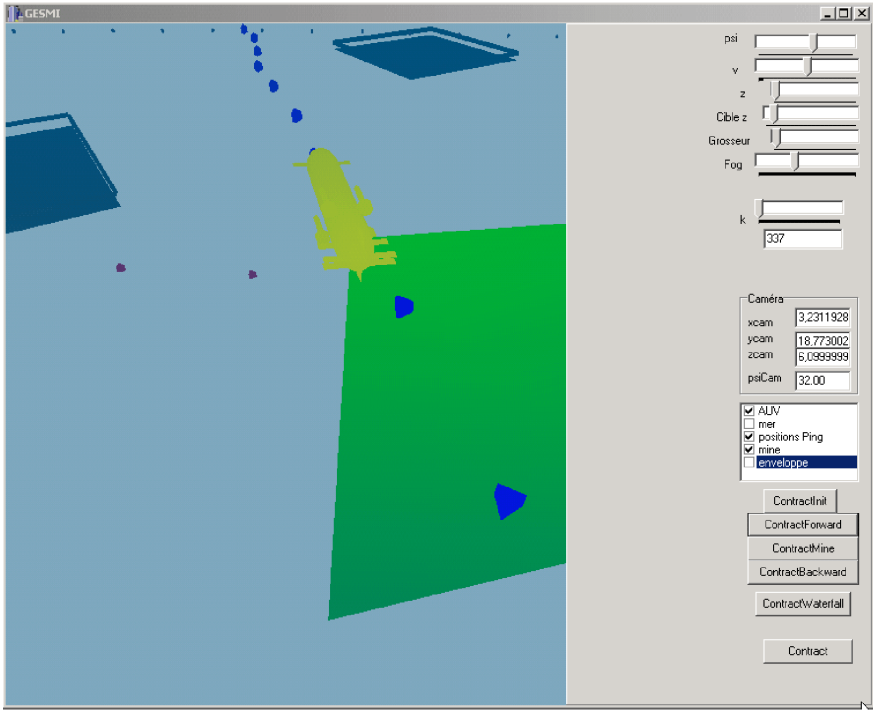
$$\dot{\mathbf{p}}(t) = \mathbf{R}(t).\mathbf{v}_r(t)$$

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

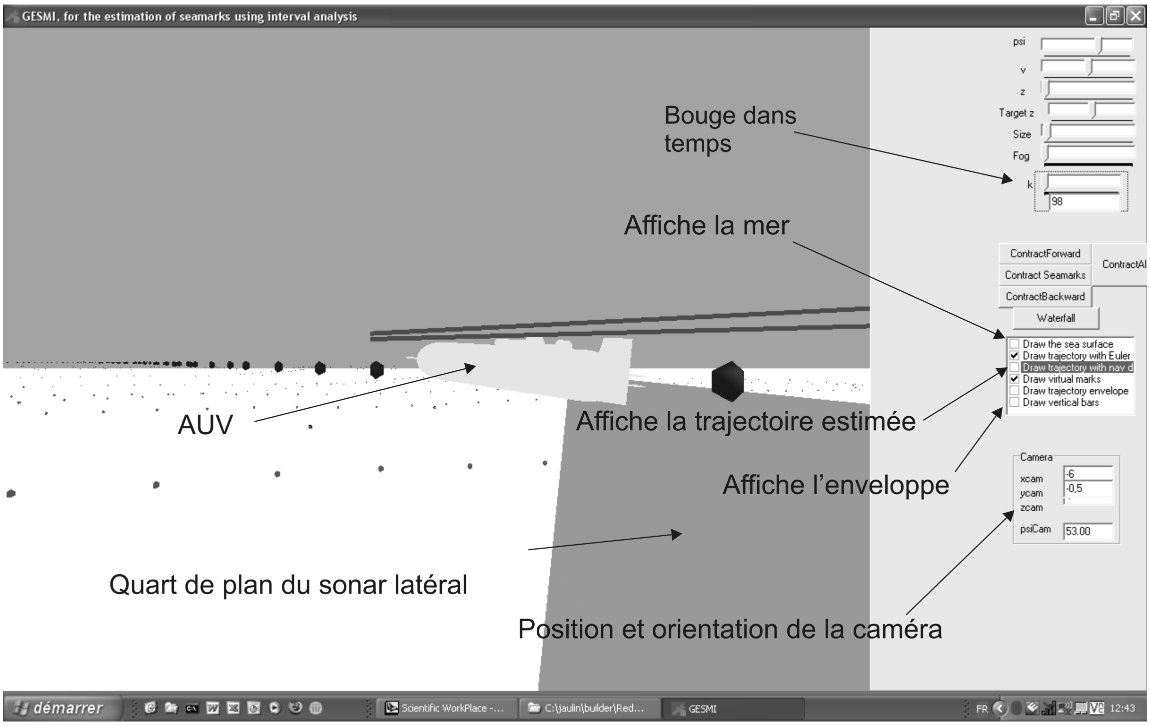
$$\mathbf{R}^{\mathsf{T}}(\tau(i))\,(\mathbf{m}(\sigma(i)) - \mathbf{p}(\tau(i))) \in [0] \times [0, \infty]^{\times 2},$$

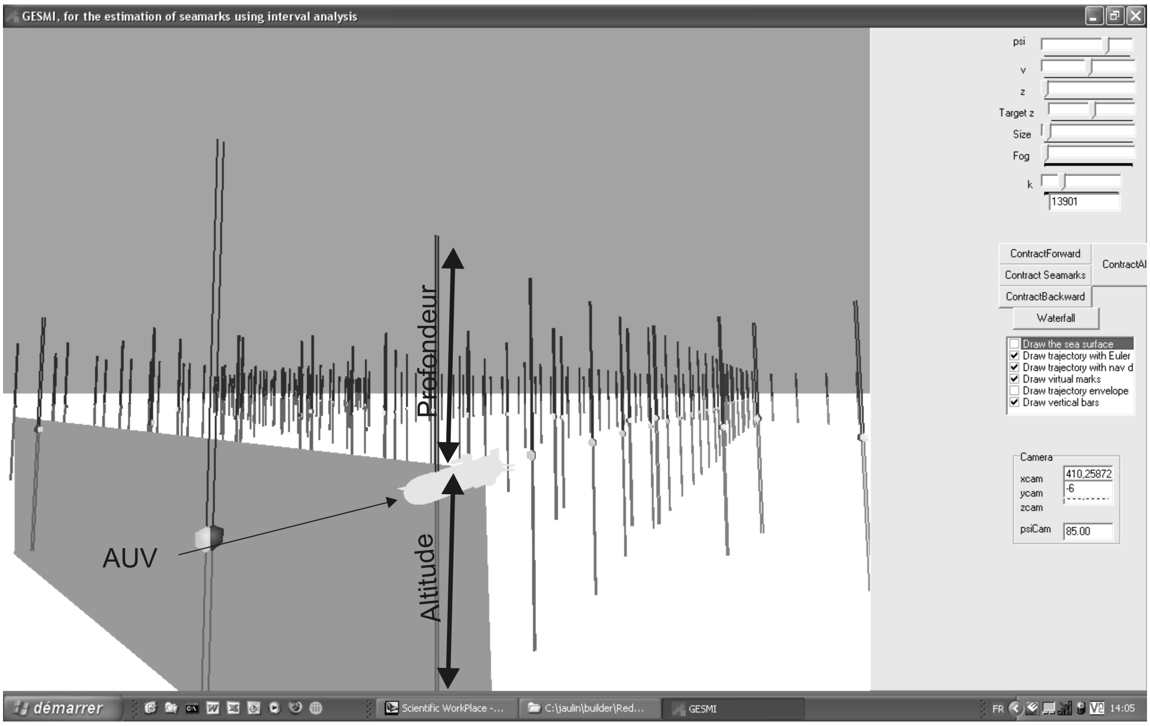
$$m_z(\sigma(i)) - p_z(\tau(i)) - a(\tau(i)) \in [-0.5, 0.5].$$

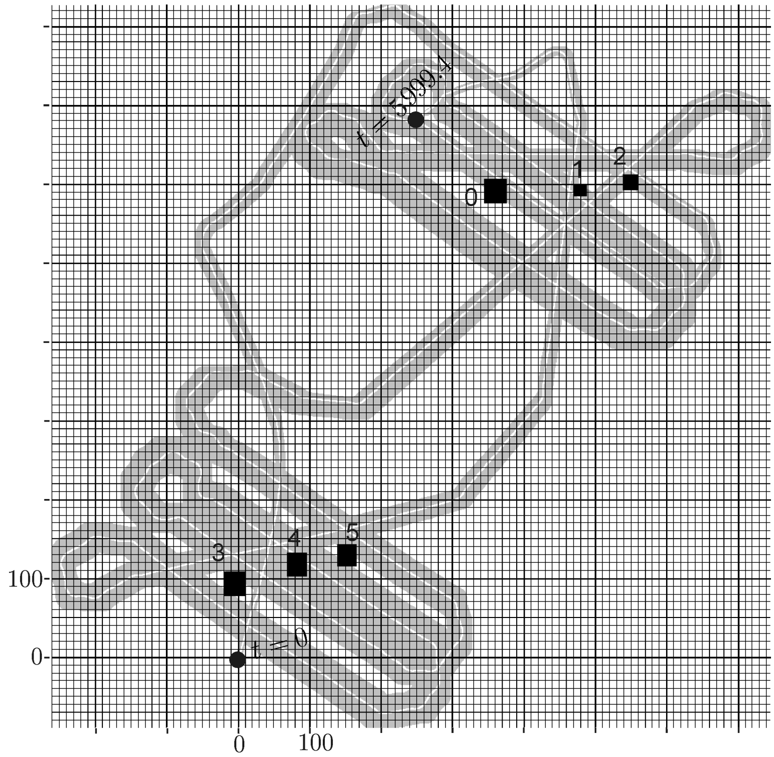
5.4 GESMI



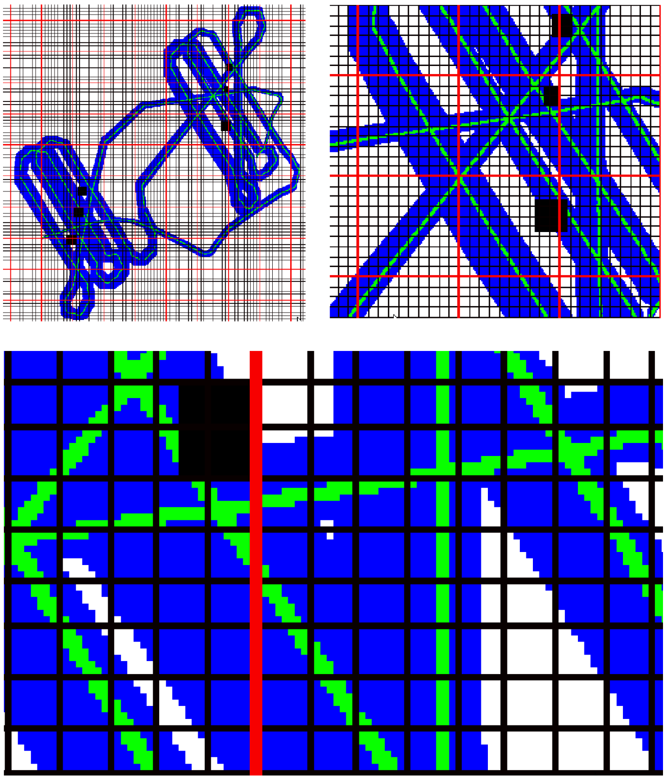
GESMI (Guaranteed Estimation of Sea Mines with Intervals)


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
Scientific WorkPlace -...
GESMI
FR
12:43


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



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



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
//----------------------------------------------------------------
```

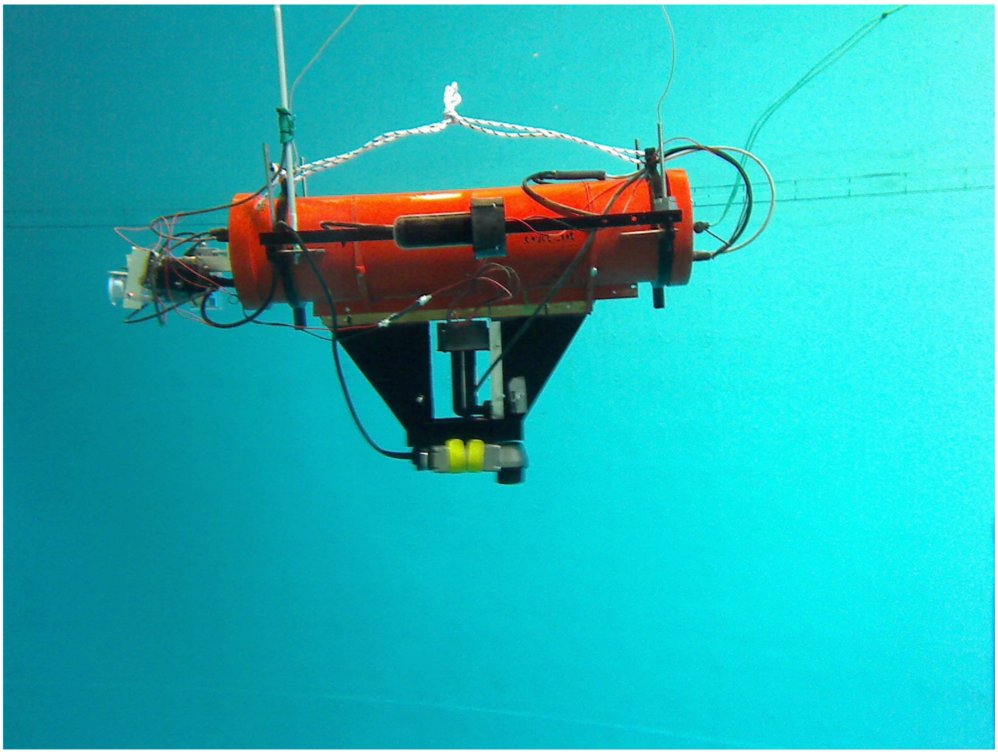
```
function R[3][3]=euler(phi,theta,psi)
  cphi = cos(phi);
  sphi = sin(phi);
  ctheta = cos(theta);
  stheta = sin(theta);
  cpsi = cos(psi);
  spsi = sin(psi);
  R[1][1]=ctheta*cpsi;
  R[1][2]=-cphi*spsi+stheta*cpsi*sphi;
  R[1][3]=spsi*sphi+stheta*cpsi*cphi;
  R[2][1]=ctheta*spsi;
  R[2][2]=cpsi*cphi+stheta*spsi*sphi;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

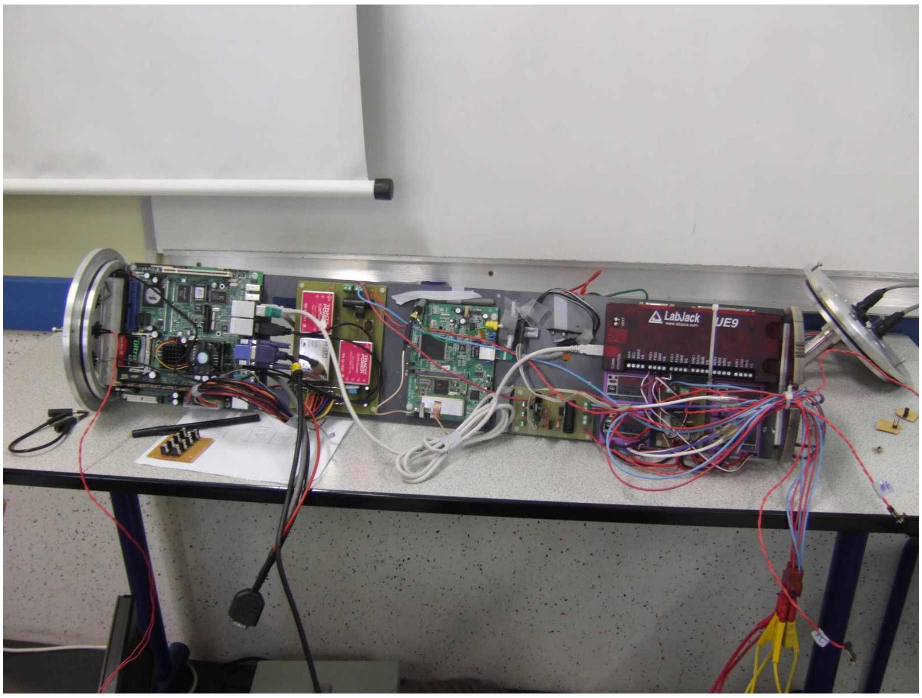
```
main
  p[1] :=read("gps_init.dat");
  v :=read("Quimper_v.dat");
  phi :=read("Quimper_phi.dat");
  theta :=read("Quimper_theta.dat");
  psi :=read("Quimper_psi.dat");
  init;
  fwd;
  bwd;
  column(p,px,1);
  column(p,py,2);
  print("--- Robot positions: ---");
  newplot("gesmi.dat");
  plot(px,py,color(rgb(1,1,1),rgb(0,0,0)));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

6 SAUC'ISSE



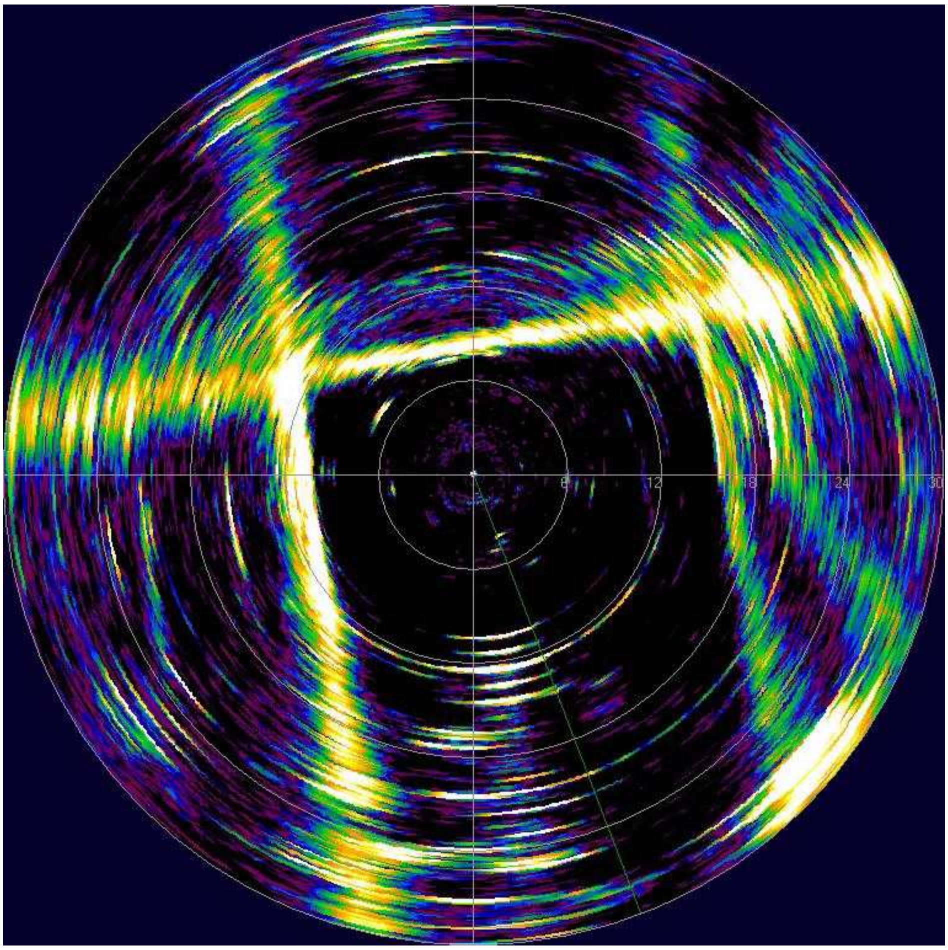
Robot SAUC'ISSE

Portsmouth, July 12-15, 2007.


LabJack
UE9

Montrer une vidéo

6.1 Localization with sonar


6
12
18
24
30

6.2 Set-membership approach

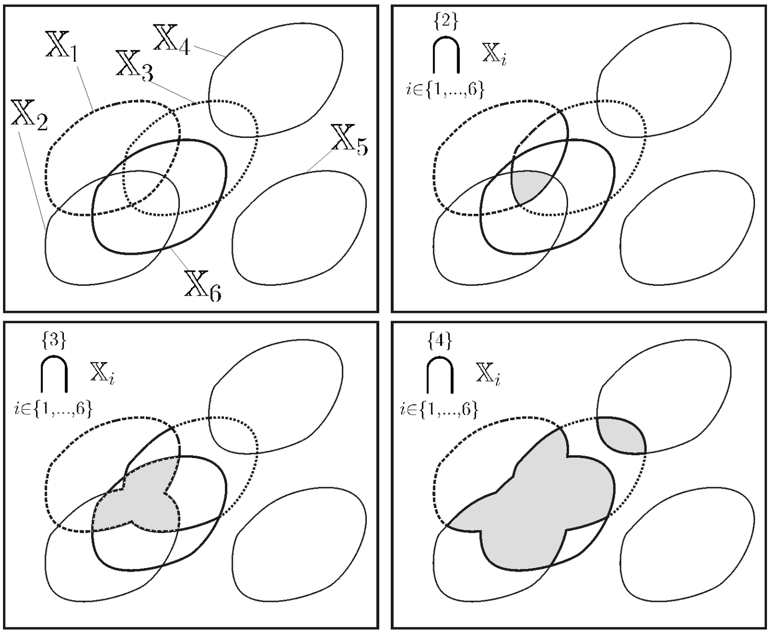
$$\begin{cases} \mathbf{x}(k+1) &= \mathbf{f}_k(\mathbf{x}(k), \mathbf{n}(k)) \\ \mathbf{y}(k) &= \mathbf{g}_k(\mathbf{x}(k)), \end{cases}$$

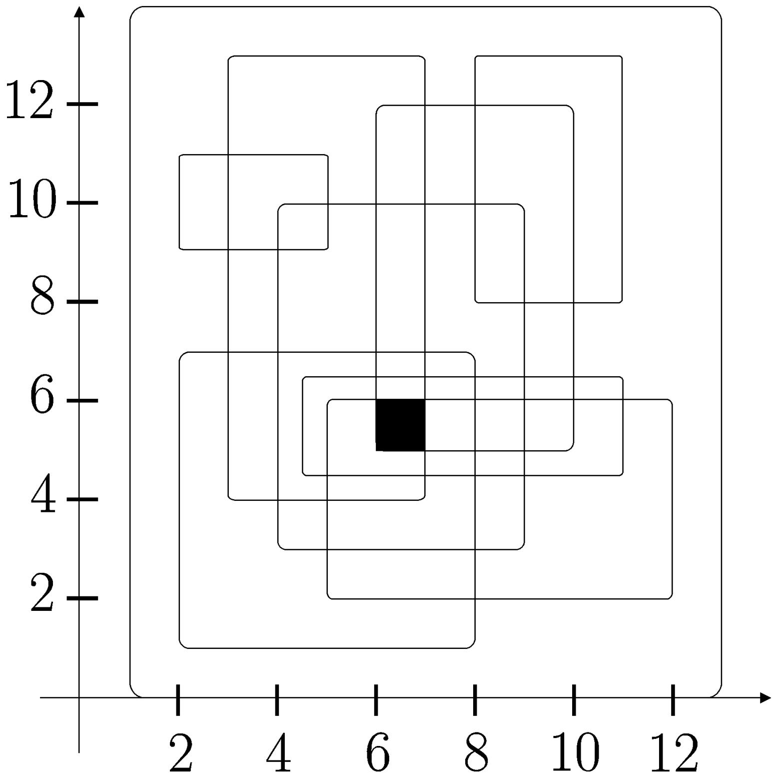
with $\mathbf{n}(k) \in \mathbb{N}(k)$ and $\mathbf{y}(k) \in \mathbb{Y}(k)$.

Without outliers

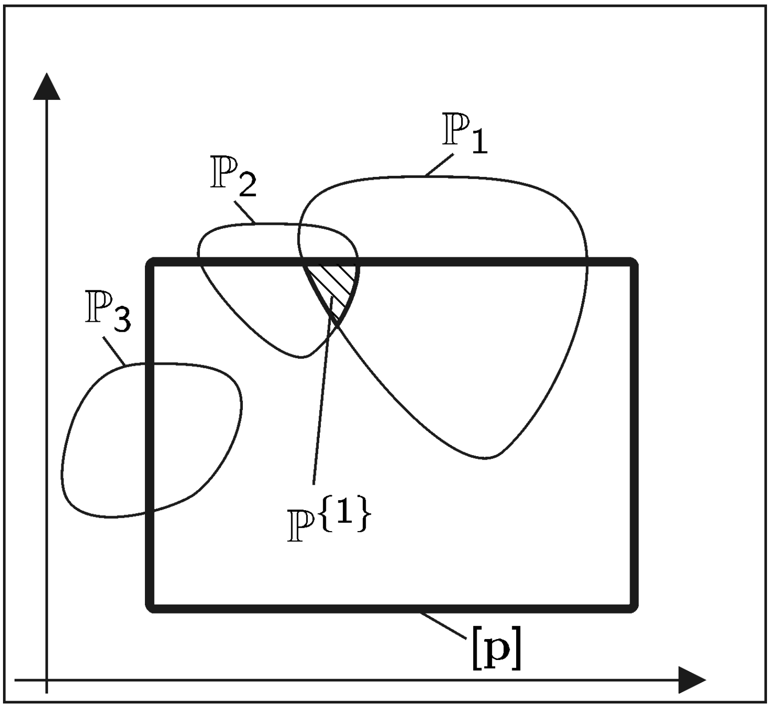
$$\mathbb{X}(k+1) = \mathbf{f}_k\left(\mathbb{X}(k) \cap \mathbf{g}_k^{-1}\left(\mathbb{Y}(k)\right),\quad \mathbb{N}\left(k\right)\right).$$

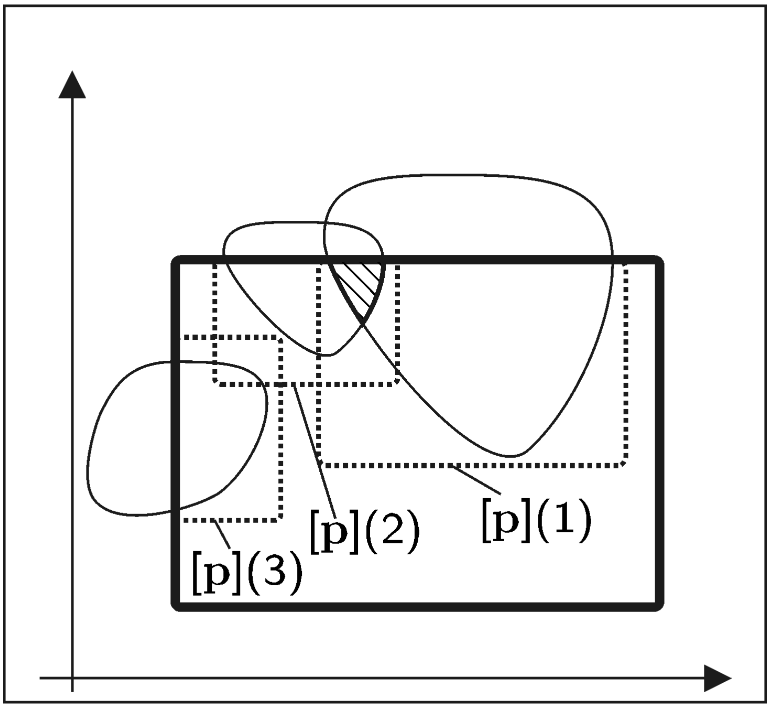
6.3 Relaxed intersection

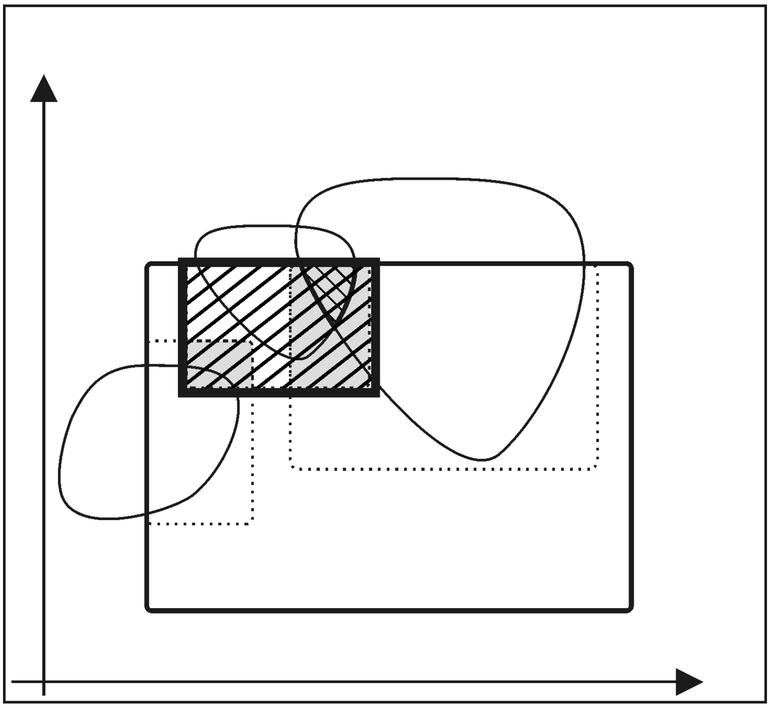

$\mathbb{X}_1$
$\mathbb{X}_2$
$\mathbb{X}_3$
$\mathbb{X}_4$
$\mathbb{X}_5$
$\mathbb{X}_6$
$\bigcap_{i\in\{1,\ldots,6\}}^{\{2\}} \mathbb{X}_i$
$\bigcap_{i\in\{1,\ldots,6\}}^{\{3\}} \mathbb{X}_i$
$\bigcap_{i\in\{1,\ldots,6\}}^{\{4\}} \mathbb{X}_i$

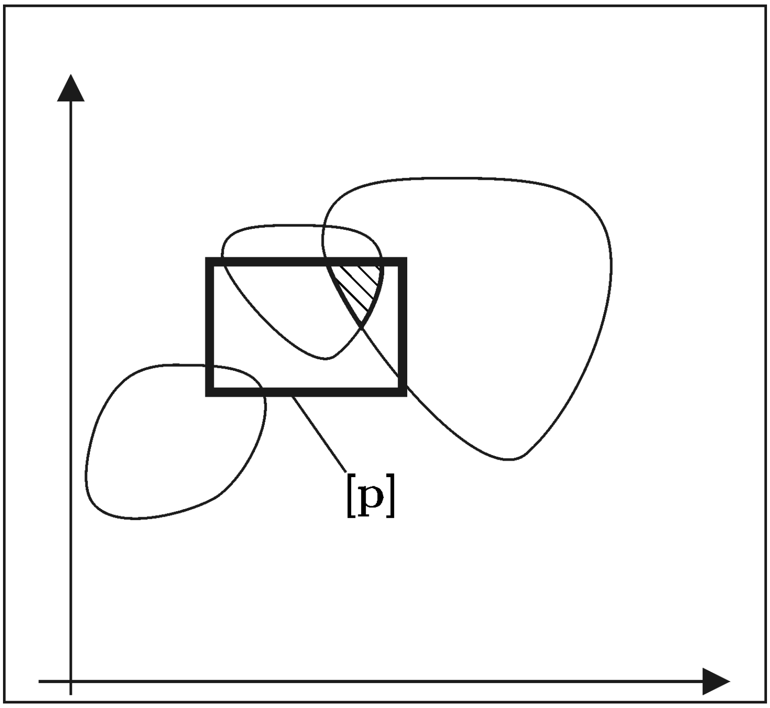


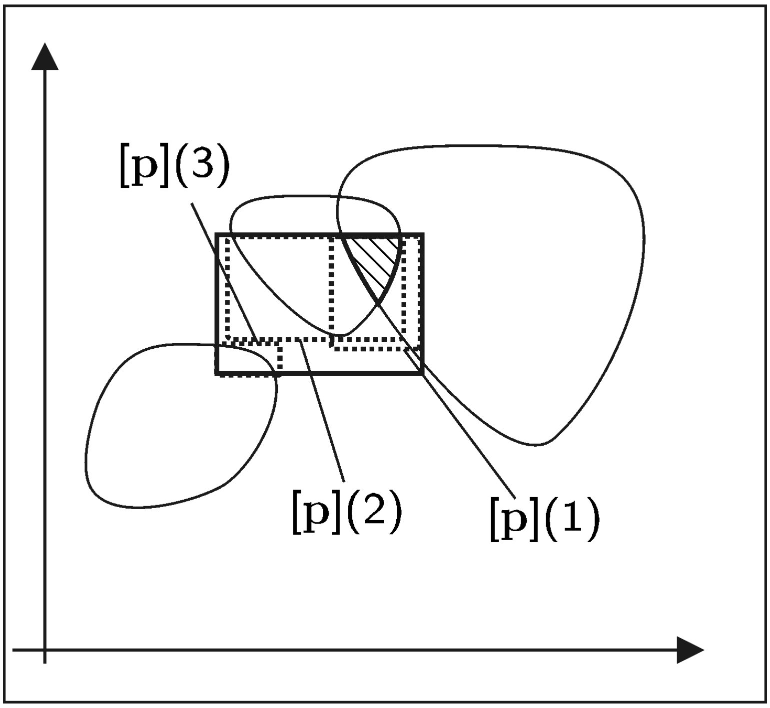
The black box is the 2-intersection of 9 boxes

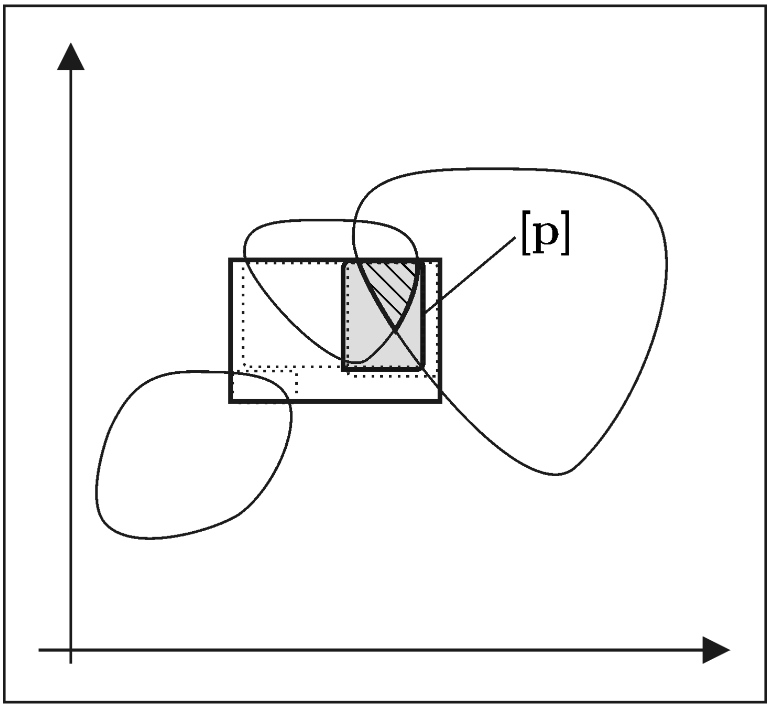

$\mathbb{P}_1$
$\mathbb{P}_2$
$\mathbb{P}_3$
$\mathbb{P}^{\{1\}}$
$[\mathbf{p}]$


[p](1)
[p](2)
[p](3)




[p]


[p](3)
[p](2)
[p](1)


[p]

Show the demo of Jan

6.4 Robust localization

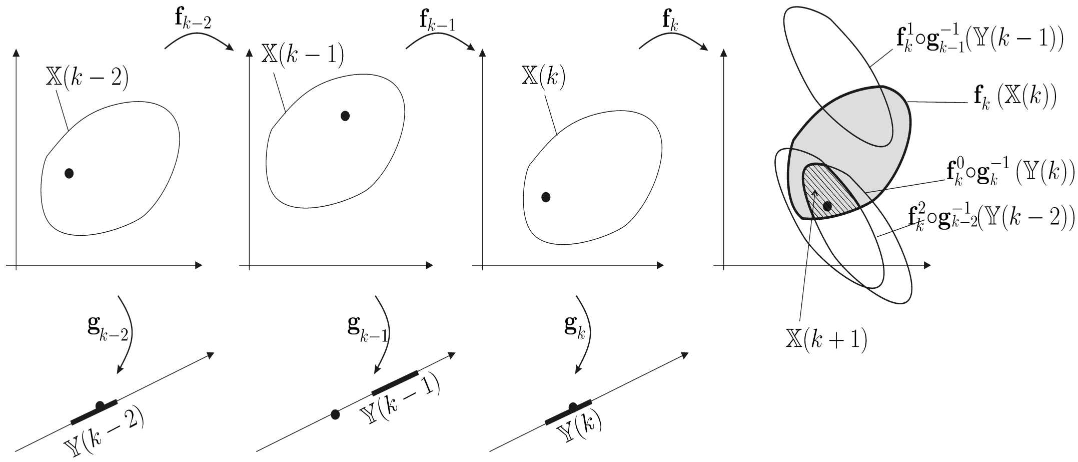
Define

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

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

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



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

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

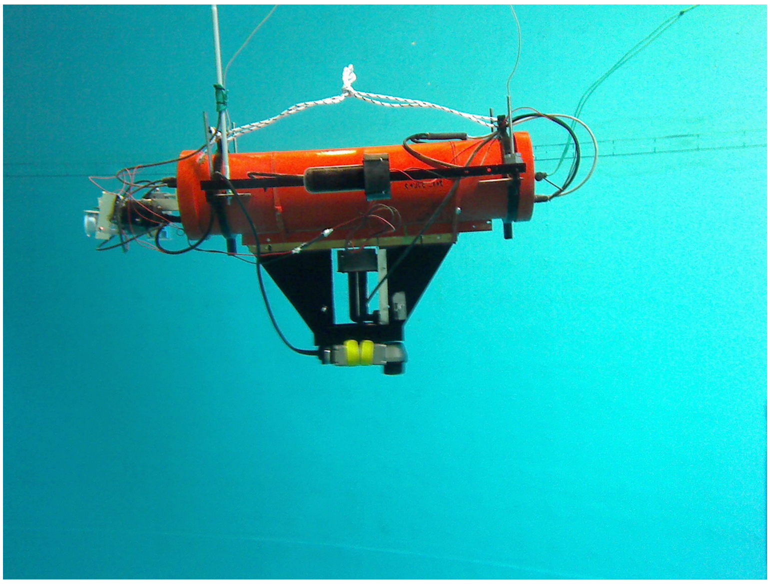
$$\Pr\left(\mathbf{x}\left(k\right) \in \mathbb{X}(k)\right) \geq \alpha \; * \; \Pr\left(\mathbf{x}\left(k-1\right) \in \mathbb{X}(k-1)\right)$$

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.

6.5 Application to localization



Sauc'isse robot inside a swimming pool

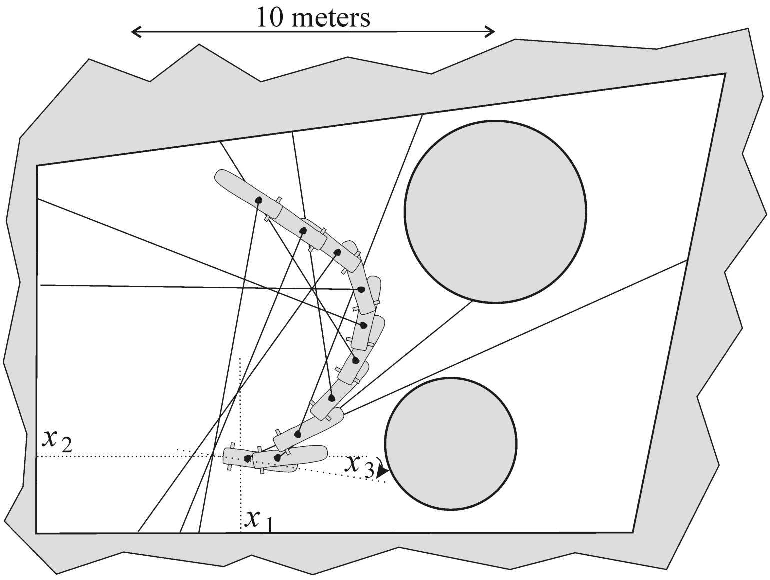
The robot evolution is described by

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

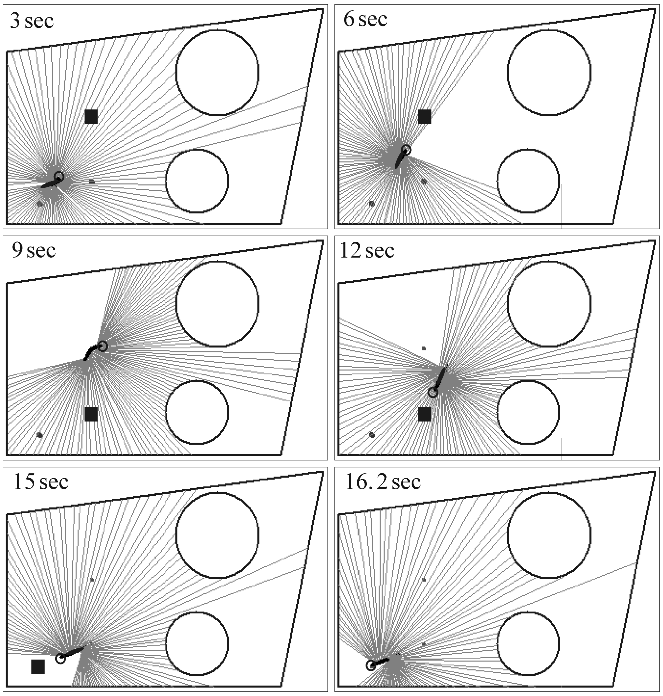
where x_1, x_2 are the coordinates of the robot center, x_3 is its orientation and x_4 is its speed. The inputs u_1 and u_2 are the accelerations provided by the propellers.

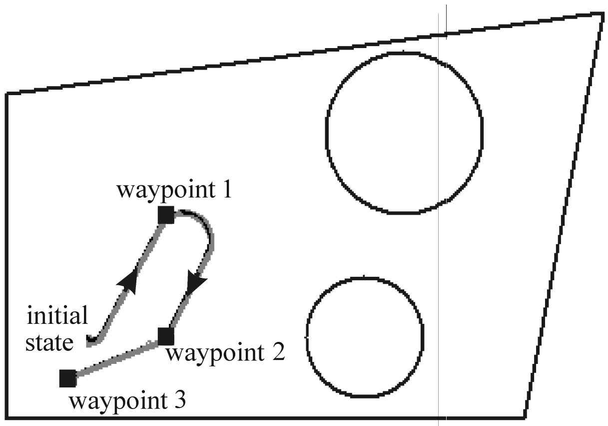
The system can be discretized by $\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k)$, where,

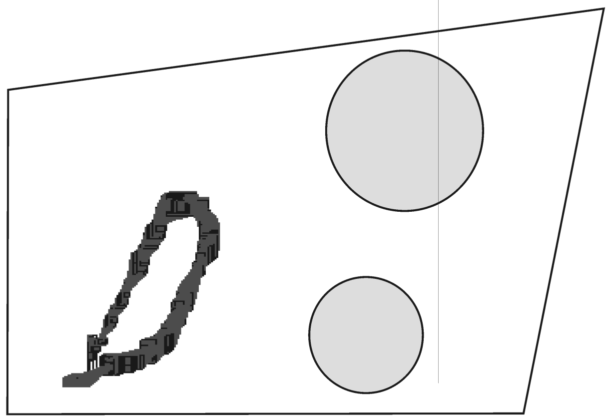
$$\mathbf{f}_k \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} x_1 + \delta . x_4 . \cos(x_3) \\ x_2 + \delta . x_4 . \sin(x_3) \\ x_3 + \delta . (u_2(k) - u_1(k)) \\ x_4 + \delta . (u_1(k) + u_2(k) - x_4) \end{pmatrix}$$



Underwater robot moving inside a pool


3 sec
6 sec
9 sec
12 sec
15 sec
16.2 sec

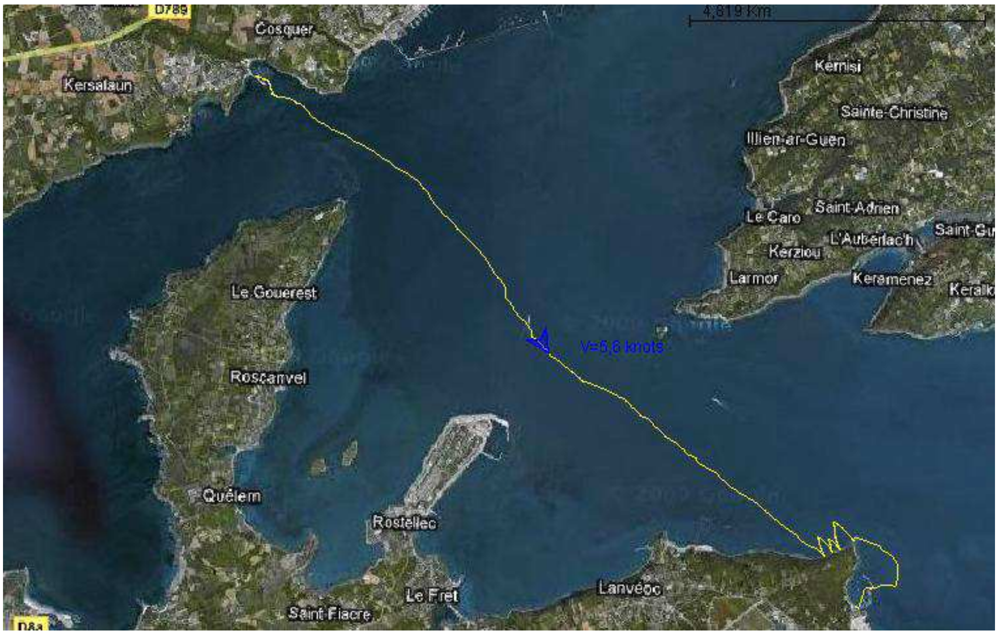

waypoint 1
initial state
waypoint 2
waypoint 3



Montrer la simu et la vidéo du concours

7 Breizh Spirit


BREIZH
ENSIETA
SPIRIT


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

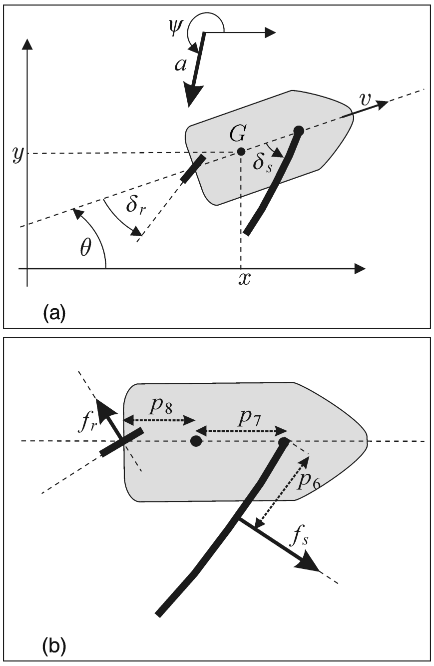
7.1 Sensors

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

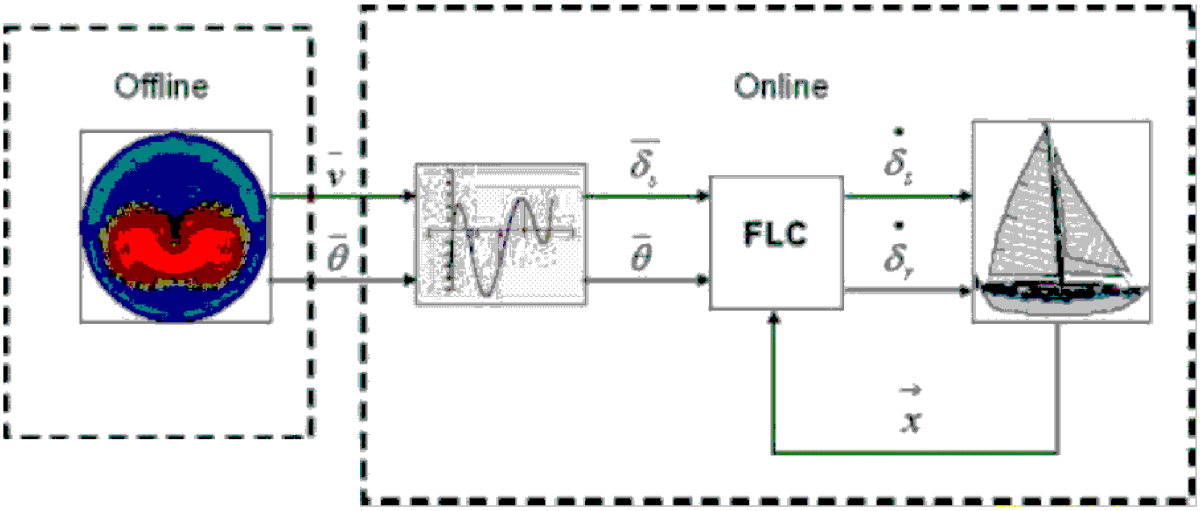
- *Unreliable sensors:* Anemometers, weather vane, dynamometers (they are directly in contact with wind, wave, salt, . . .) and can fail down at any time.

7.2 Normalized State equations

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


ψ
a
v
G
y
δs
δr
θ
x
(a)
fr
p8
p7
p6
fs
(b)

7.3 Control



Offline
Online
$\bar{v}$
$\bar{\theta}$
$\bar{\delta}_s$
$\bar{\theta}$
FLC
$\dot{\delta}_s$
$\dot{\delta}_r$
$\vec{x}$

7.4 Observer

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

If the system

$$\left\{\begin{array}{lcl}\dot{\mathbf{x}} & = & \mathbf{f}(\mathbf{x},\mathbf{u}) \\ \mathbf{y} & = & \mathbf{g}(\mathbf{x}),\end{array}\right.$$

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

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

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

- The *derivation step* computes symbolically $\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(r)}$ 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:

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

i.e.

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

As a consequence,

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

7.5 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

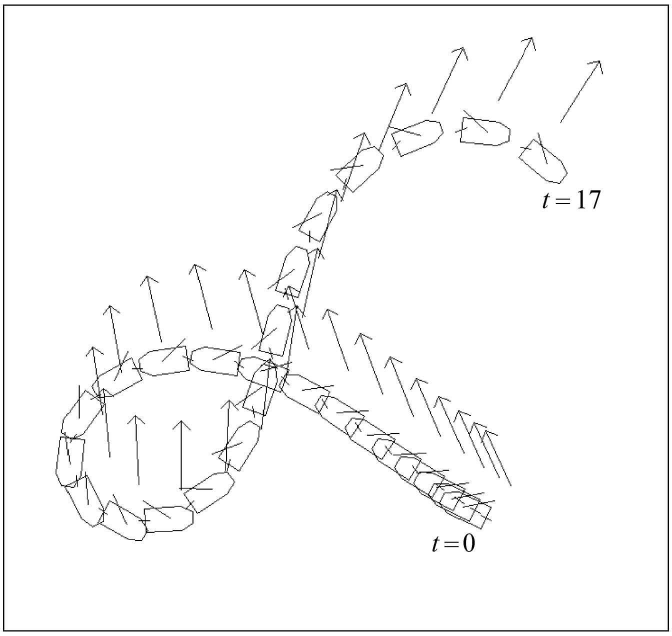
$$\underbrace{\begin{pmatrix} \theta \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{pmatrix}}_{\mathbf{z}} = \underbrace{\begin{pmatrix} \theta \\ v\sin\theta + a\sin\psi \\ v\sin\theta + a\sin\psi \\ \omega \\ (f_s.\sin\delta_s - f_r\sin u_1 - v)\cos\theta - \omega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

with

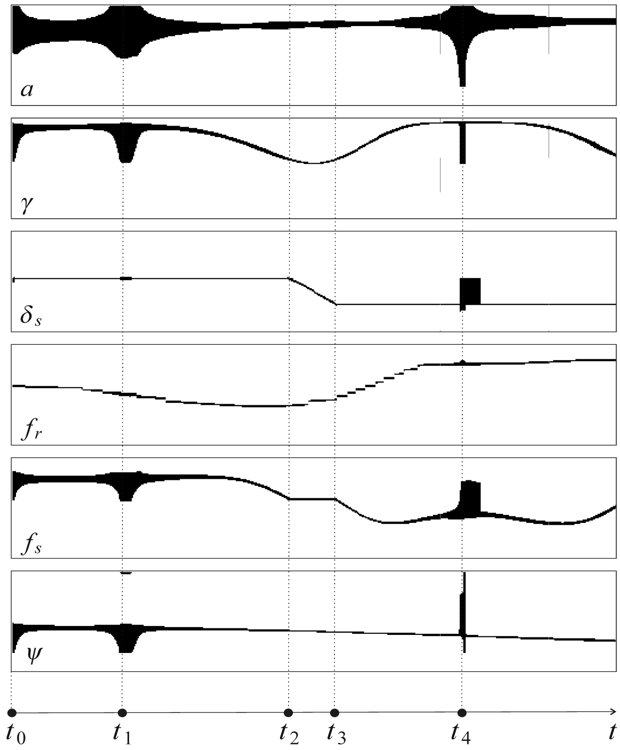
$$\mathbf{w} = \begin{pmatrix} \theta & v & \omega & a & \psi & u_1 & u_2 \end{pmatrix}^{\mathsf{T}}$$

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