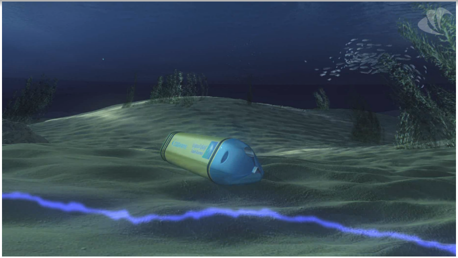
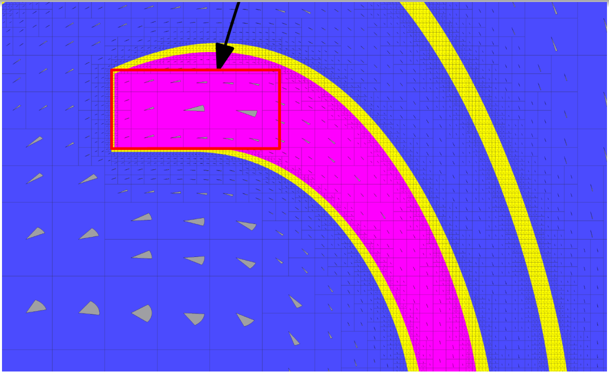
Interval Analysis for Cyber-Physical Systems

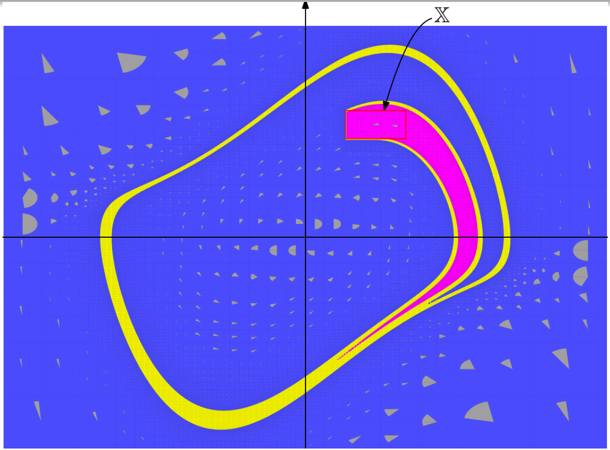
Luc Jaulin

August 7, 2017



https://youtu.be/ItlAi-UMup8




X

Intervals

Problem. Given $f : \mathbb{R}^n \to \mathbb{R}$ and a box $[\mathbf{x}] \subset \mathbb{R}^n$, prove that

$$\forall \mathbf{x} \in [\mathbf{x}], f(\mathbf{x}) \geq 0.$$

Interval arithmetic can solve efficiently this problem.

Example. Is the function

$$f(\mathbf{x}) = x_1 x_2 - (x_1 + x_2)\cos x_2 + \sin x_1 \cdot \sin x_2 + 2$$

always positive for $x_1, x_2 \in [-1, 1]$?

Set theory

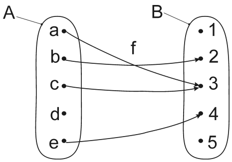
The *direct image* of $\mathbb{X}$ by f is

$$f(\mathbb{X}) \triangleq \{f(x) \mid x \in \mathbb{X}\}.$$

The *reciprocal image* of $\mathbb{Y}$ by f is

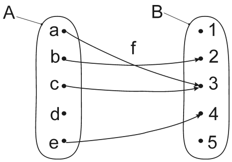
$$f^{-1}(\mathbb{Y}) \triangleq \{x \in \mathbb{X} \mid f(x) \in \mathbb{Y}\}.$$

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



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

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

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

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

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

This is consistent with the property

$$f^{-1}(f(\mathbb{Y})) \supset \mathbb{Y}.$$

Interval arithmetic

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

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

where $[\mathbb{A}]$ is the smallest interval which encloses $\mathbb{A} \subset \mathbb{R}$.

Exercise.

$$\begin{aligned} [-1,3]+[2,5] &= [?,?], \\ [-1,3]\cdot[2,5] &= [?,?], \\ [-2,6]/[2,5] &= [?,?]. \end{aligned}$$

Solution.

$$\begin{aligned} [-1,3]+[2,5] &= [1,8], \\ [-1,3].[2,5] &= [-5,15], \\ [-2,6]/[2,5] &= [-1,3]. \end{aligned}$$

Exercise. Compute

$$[-2,2]/[-1,1] \quad = [?,?].$$

Solution.

$$[-2,2]/[-1,1] \quad = [-\infty,\infty].$$

$$\begin{aligned}
[x^-,x^+]+[y^-,y^+] = \quad & [x^-+y^-,x^++y^+],\\
[x^-,x^+]\cdot[y^-,y^+] = \quad & [x^-y^-\wedge x^+y^-\wedge x^-y^+\wedge x^+y^+,\\
& x^-y^-\vee x^+y^-\vee x^-y^+\vee x^+y^+],
\end{aligned}$$

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

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

Exercise.

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

Solution.

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

```
from pyibex import *
A=Interval(-2,3)
B=Interval(-5,3)
Z=Interval.ALL_REALS

print('A=',A)
print('A^2=',sqr(A))
print('A*B=',A*B)
print('sin(sqr(Z)=',sqr(sin(Z)))


X=IntervalVector([[2,4],[3,5]])
print('X=',X)
f = Function("x1","x2","(x1-1)^2+(x2-2)^2")
C=f.eval(X)
print('C=',C)
```

```
A= [-2, 3]
A^2= [0, 9]
A*B= [-15, 10]
sin(sqr(Z)= [0, 1]
X= ([2, 4] ; [3, 5])
C= [2, 18]
```

Boxes

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

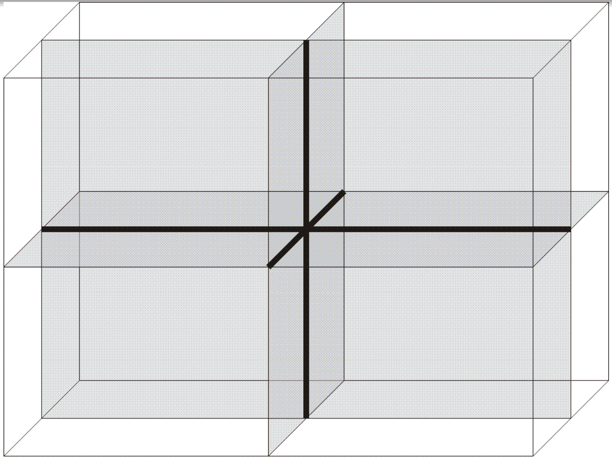
$$[\mathbf{x}] = [x_1^-, x_1^+] \times \cdots \times [x_n^-, x_n^+] = [x_1] \times \cdots \times [x_n].$$

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

The *width* $w([\mathbf{x}])$ is the length of the largest side.

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

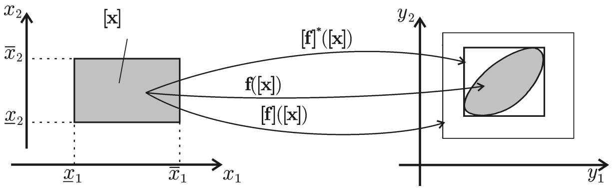
The *principal plane* of $[\mathbf{x}]$ is symmetric and perpendicular to the largest side.



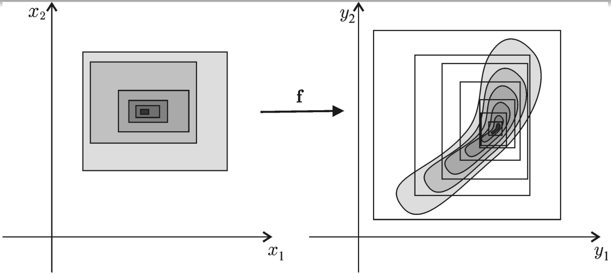
Inclusion function

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

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



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


x_2
x_1
f
y_2
y_1

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

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

For $[x] = [-3, 4]$, compute $[f]([x])$ and $f([x])$.

Solution. 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

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

is

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

If **f** is given by

Algorithm 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
    $z := x_2(z + k \cdot x_3);$
next;
$y_1 := z;$
$y_2 := \sin(zx_1);$

Its natural inclusion function is

Algorithm [f](in: [**x**], out: [**y**])

$[z] := [x_1];$
for $k := 0$ to 100
    $[z] := [x_2] \cdot ([z] + k \cdot [x_3]);$
next;
$[y_1] := [z];$
$[y_2] := \sin([z] * [x_1]);$

Set inversion

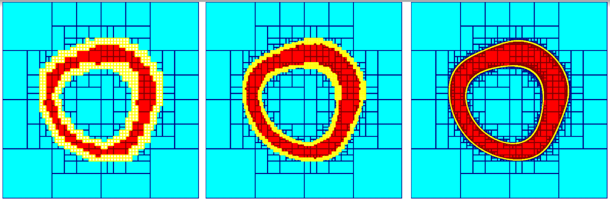
A subpaving of $\mathbb{R}^n$ is a set of non-overlapping boxes of $\mathbb{R}^n$.
Compact sets $\mathbb{X}$ can be bracketed between inner and outer subpavings:

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

Exercise. The set

$$\mathbb{X} = \{(x_1, x_2) \mid x_1^2 + x_2^2 + \sin(x_1 + x_2) \in [4, 9]\}$$

are approximated by $\mathbb{X}^-$ and $\mathbb{X}^+$ for different accuracies. Denote by $\mathbb{R}, \mathbb{Y}, \mathbb{B}$ the union of red, yellow, blue boxes. Denote by $\partial\mathbb{X}$ the boundary of $\mathbb{X}$.



$\mathbb{X}^- \cap \mathbb{B} = \emptyset$	yes or no
$\mathbb{X} \cap \mathbb{B} \neq \emptyset$	yes or no
$\mathbb{X}^+ = \mathbb{R} \cup \mathbb{Y}$	yes or no
$\partial\mathbb{X} \supset \mathbb{Y}$	yes or no

Solution. We have

$$\begin{array}{lcl} \mathbb{X}^- \cap \mathbb{B} = \emptyset & \rightarrow & \text{Yes} \\ \mathbb{X} \cap \mathbb{B} = \emptyset & \rightarrow & \text{No} \\ \mathbb{X}^+ = \mathbb{R} \cup \mathbb{Y} & \rightarrow & \text{Yes} \\ \partial\mathbb{X} \supset \mathbb{Y} & \rightarrow & \text{No. Instead, we have } \partial\mathbb{X} \subset \mathbb{Y} \end{array}$$

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

If $\mathbf{f}:\mathbb{R}^n \to \mathbb{R}^m$ and $\mathbb{Y} \subset \mathbb{R}^m$.

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

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

Boxes for which these tests failed, will be bisected, except if they are too small [4].

SIVIA

Algorithm Sivia(in: $[\mathbf{x}](0), \mathbf{f}, \mathbb{Y}$)

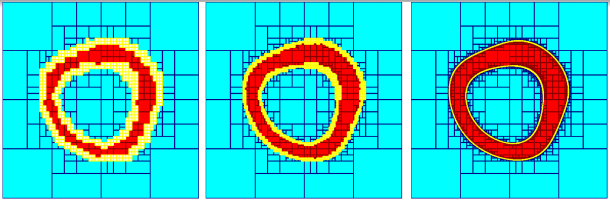
1 $\mathscr{L} := \{[\mathbf{x}](0)\}$;
2 pull $[\mathbf{x}]$ from $\mathscr{L}$;
3 if $[\mathbf{f}]([\mathbf{x}]) \subset \mathbb{Y}$, draw($[\mathbf{x}]$, 'red');
4 elseif $[\mathbf{f}]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset$, draw($[\mathbf{x}]$, 'blue');
5 elseif $w([\mathbf{x}]) < \varepsilon$,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

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

$$\mathbb{X}^- \subset \mathbb{X} \subset \underbrace{\mathbb{X}^- \cup \Delta\mathbb{X}}_{\mathbb{X}^+}.$$

Example: 4 rings

Contractors



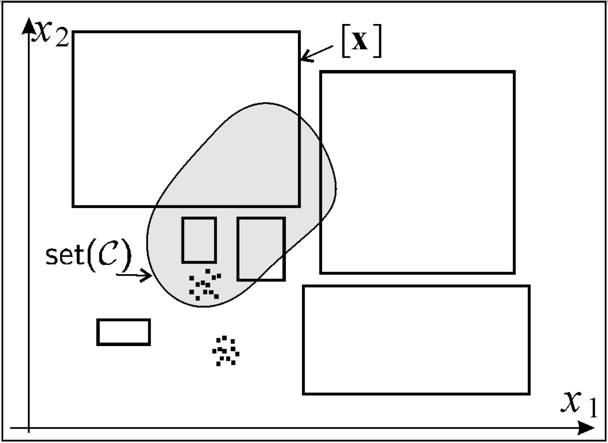
To characterize $\mathbb{X} \subset \mathbb{R}^n$, bisection algorithms bisect all boxes in all directions and become inefficient. Interval methods can still be useful if

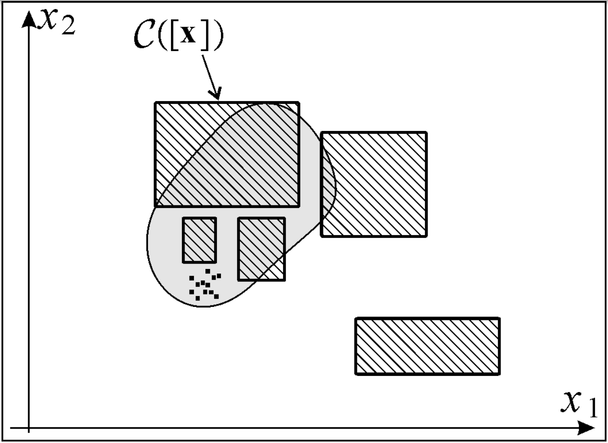
- the solution set $\mathbb{X}$ is small (optimization problem, solving equations),
- contraction procedures are used as much as possible,
- bisections are used only as a last resort.

Definition

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

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \begin{cases} \mathscr{C}([\mathbf{x}]) \subset [\mathbf{x}] & \text{(contractance)}, \\ \mathscr{C}([\mathbf{x}]) \cap \mathbb{X} = [\mathbf{x}] \cap \mathbb{X} & \text{(completeness)}. \end{cases}$$


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


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

The operator $\mathscr{C} : \mathbb{IR}^n \to \mathbb{IR}^n$ is a *contractor* for the equation $f(\mathbf{x}) = 0$, if

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \begin{cases} \mathscr{C}([\mathbf{x}]) \subset [\mathbf{x}] \\ \mathbf{x} \in [\mathbf{x}] \text{ et } f(\mathbf{x}) = 0 \Rightarrow \mathbf{x} \in \mathscr{C}([\mathbf{x}]) \end{cases}$$

Constraint projections

Exercice. Let x, y, z be 3 variables such that

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

Contract the intervals for x, y, z.

Solution.

$$
\begin{aligned}
[x] &= [2,5] \\
[y] &= [1,4] \\
[z] &= [6,9]
\end{aligned}
$$

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

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

Numerical method for projection

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

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

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 can be extended to other ternary constraints such as mult: $z = x \cdot y$, or equivalently

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

The resulting projection procedure becomes

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

For the binary constraint

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

the associated contractor is

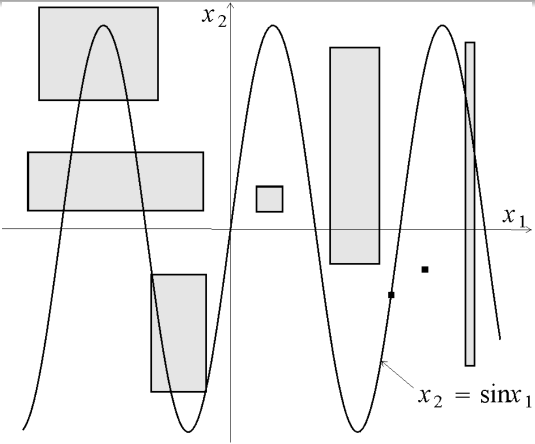
Algorithm pexp(inout: $[y],[x]$)

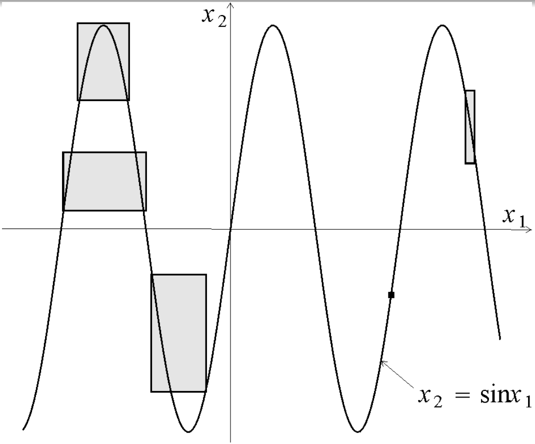
1 $[y] := [y] \cap \exp([x])$;
2 $[x] := [x] \cap \log([y])$.

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

Example. Consider the primitive equation:

$$x_2 = \sin x_1.$$


x_2
x_1
$x_2 = \sin x_1$


x_2
x_1
$x_2 = \sin x_1$

Decomposition

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

Decomposition

$$\begin{array}{l} x+\sin(xy)\leq 0, \\ x\in[-1,1], y\in[-1,1] \end{array}$$

can be decomposed into

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

Forward-backward contractor (HC4 revise)

For the equation

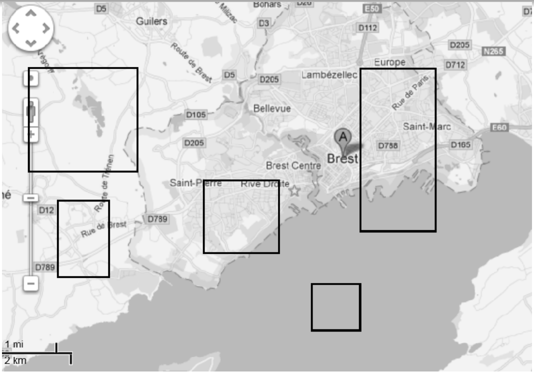
$$(x_1 + x_2) \cdot x_3 \in [1,2],$$

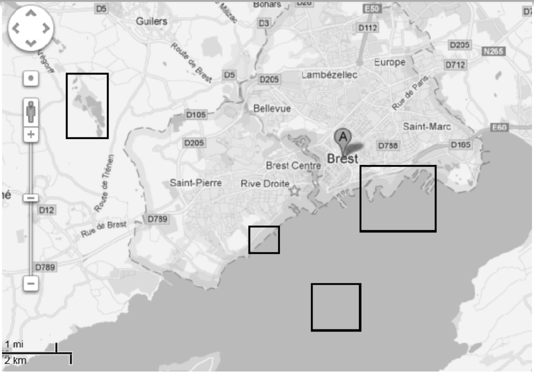
we have the following contractor:

algorithm $\mathscr{C}$ (inout $[x_1],[x_2],[x_3]$)	
$[a] = [x_1] + [x_2]$	% $a = x_1 + x_2$
$[b] = [a] \cdot [x_3]$	% $b = a \cdot x_3$
$[b] = [b] \cap [1,2]$	% $b \in [1,2]$
$[x_3] = [x_3] \cap \frac{[b]}{[a]}$	% $x_3 = \frac{b}{a}$
$[a] = [a] \cap \frac{[b]}{[x_3]}$	% $a = \frac{b}{x_3}$
$[x_1] = [x_1] \cap [a] - [x_2]$	% $x_1 = a - x_2$
$[x_2] = [x_2] \cap [a] - [x_1]$	% $x_2 = a - x_1$

Contractor on images

The robot with coordinates (x_1, x_2) is in the water.


Guilers
Bohars
Route de Brest
Lambézellec
Europe
Bellevue
Rue de Paris
Saint-Marc
Brest
Brest Centre
Saint-Pierre
Rive Droite
Rue de Brest
D5
D3
D112
D205
D712
N265
D105
E60
D788
D165
D12
D789
1 mi
2 km


Guilers
Bonars
D5
D3
D112
E50
D205
N265
Europe
D712
Lambézellec
Route de Brest
D205
Rue de Paris
Bellevue
D105
Saint-Marc
E60
D205
D788
D165
Brest
Brest Centre
Saint-Pierre
Rive Droite
Route de Trénen
D12
Rue de Brest
D789
D789
1 mi
2 km

Solving equations

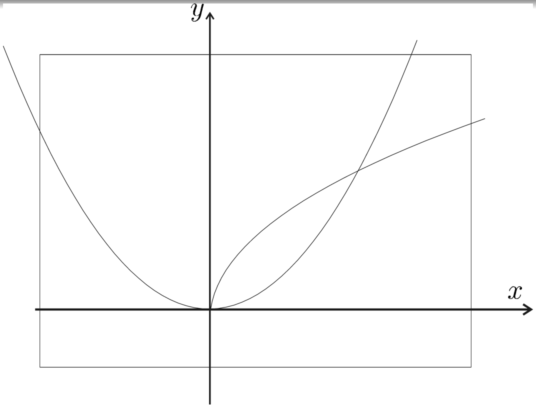
Consider the system of two equations.

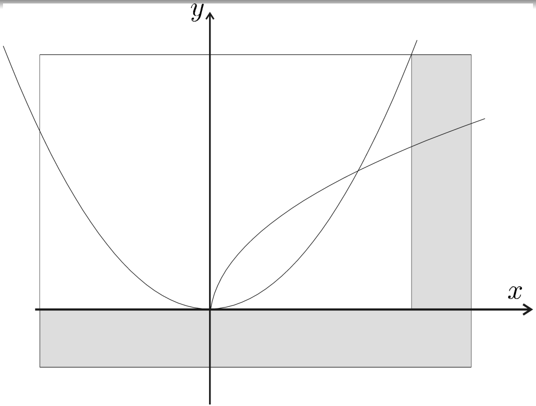
$$\begin{array}{rcl} y & = & x^2 \\ y & = & \sqrt{x}. \end{array}$$

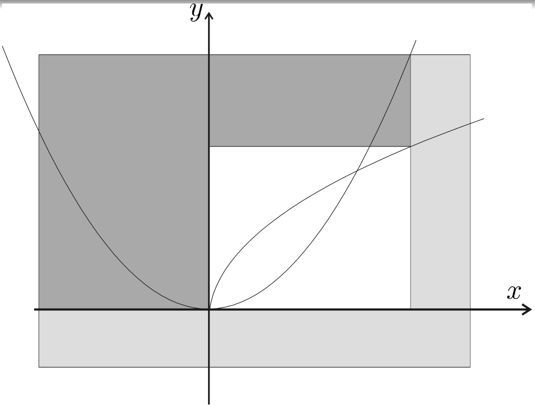
We can build two contractors

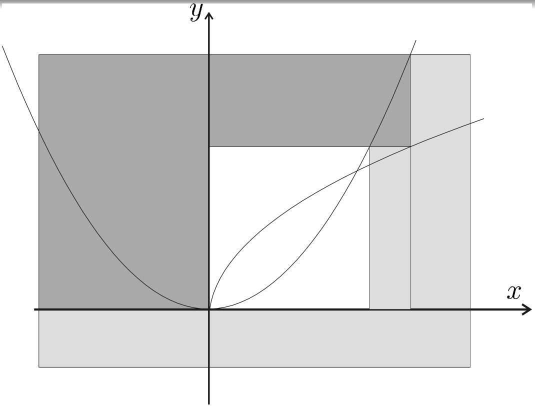
$$\mathscr{C}_1 : \begin{cases} [y] = [y] \cap [x]^2 \\ [x] = [x] \cap \sqrt{[y]} \end{cases} \quad \text{associated to } y = x^2$$

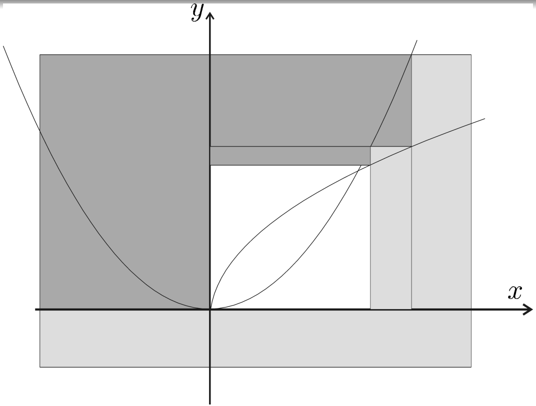
$$\mathscr{C}_2 : \begin{cases} [y] = [y] \cap \sqrt{[x]} \\ [x] = [x] \cap [y]^2 \end{cases} \quad \text{associated to } y = \sqrt{x}$$

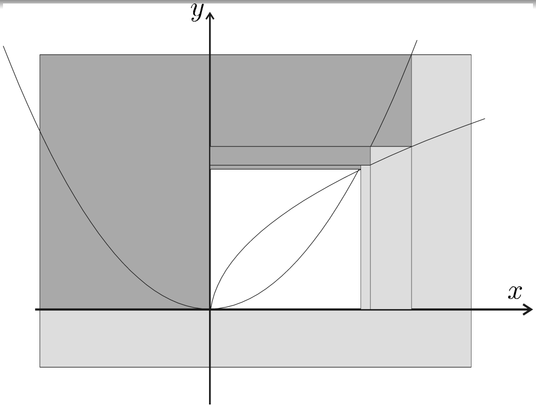

y
x

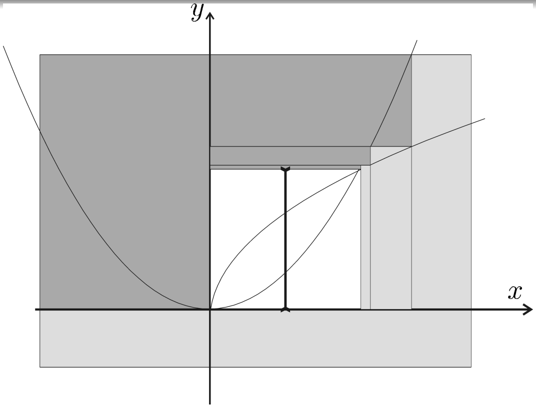

y
x

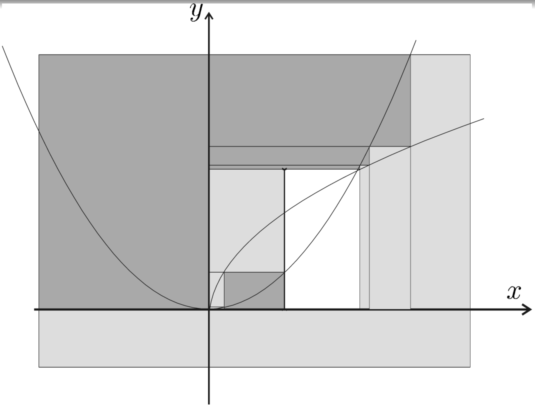

y
x

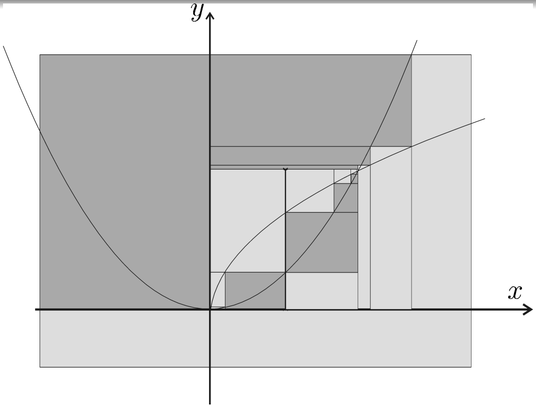

y
x


y
x


y
x


y
x

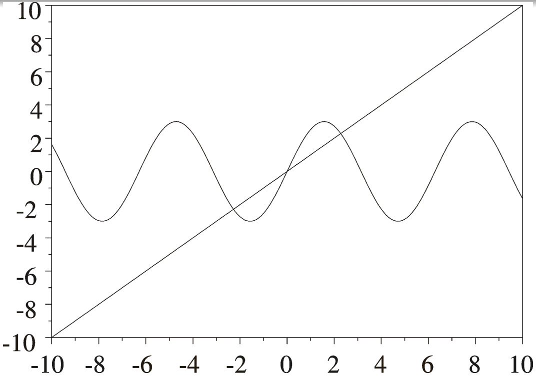

y
x

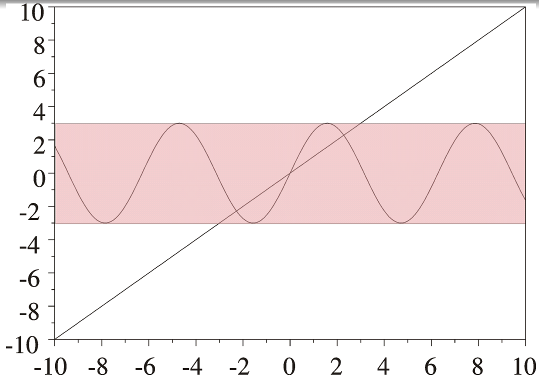

y
x

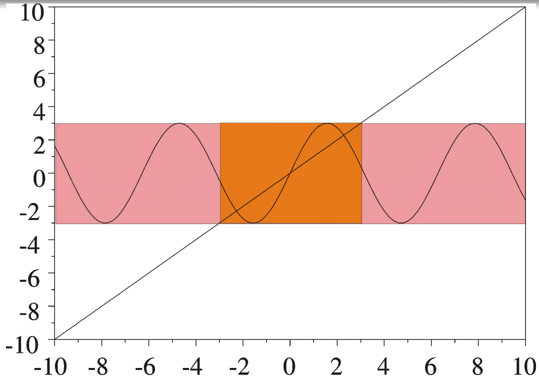
Example 2

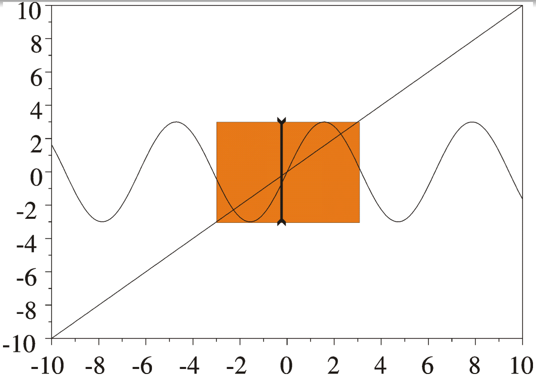
Exemple. Consider the system

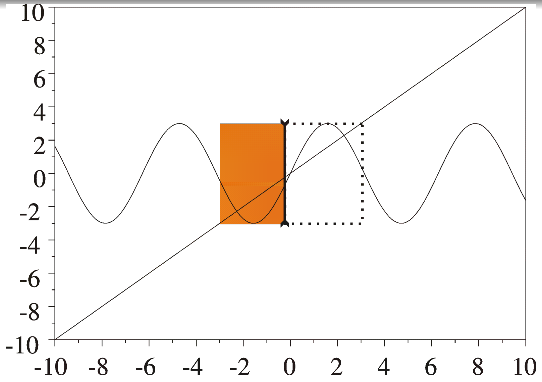
$$\left\{ \begin{array}{rcl} y & = & 3\sin(x) \\ y & = & x \end{array} \right. \qquad x \in \mathbb{R},\ y \in \mathbb{R}.$$

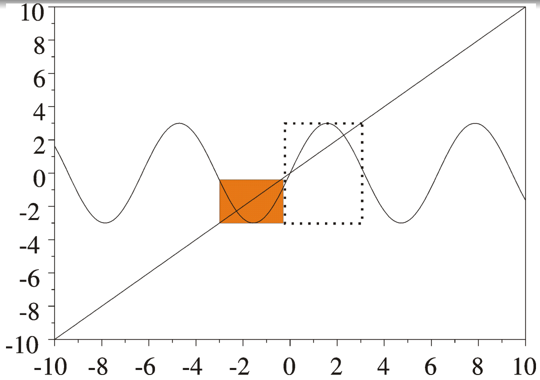

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

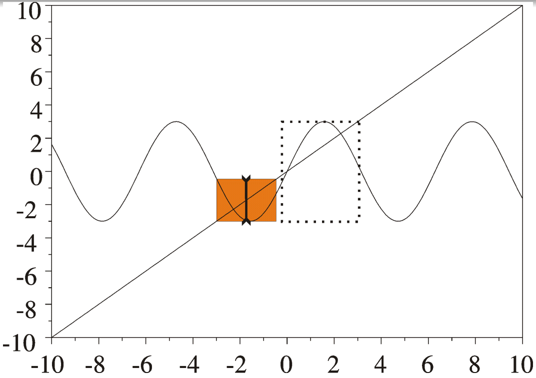

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10

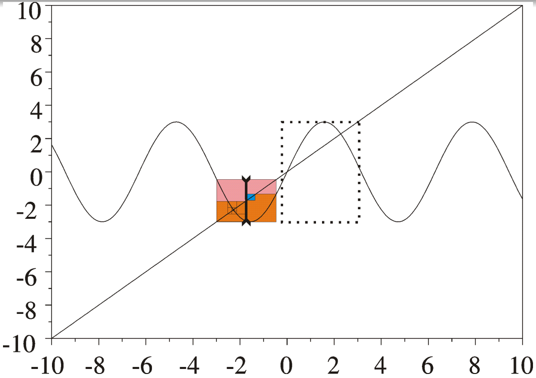

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10

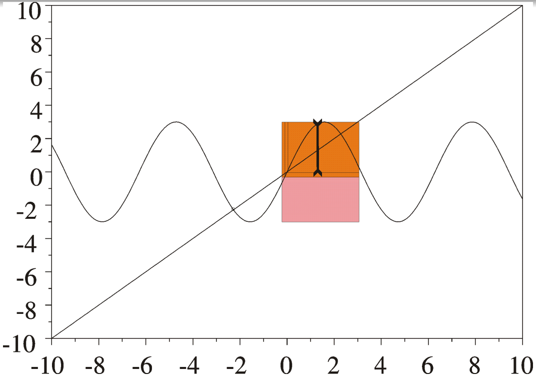

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10

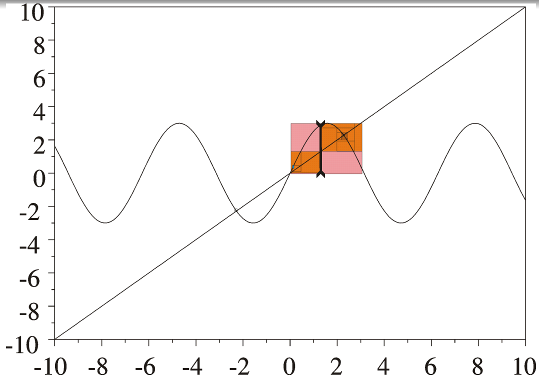

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10


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

Algebra

Some operations on contractors can be defined [1]

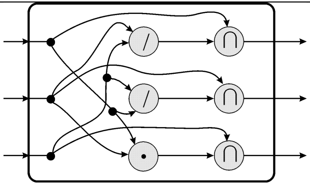
intersection	$(\mathscr{C}_1 \cap \mathscr{C}_2)([\mathbf{x}]) = \mathscr{C}_1([\mathbf{x}]) \cap \mathscr{C}_2([\mathbf{x}])$
union	$(\mathscr{C}_1 \cup \mathscr{C}_2)([\mathbf{x}]) = [\mathscr{C}_1([\mathbf{x}]) \cup \mathscr{C}_2([\mathbf{x}])]$
composition	$(\mathscr{C}_1 \circ \mathscr{C}_2)([\mathbf{x}])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

A link between matrices and contractors

$$\text{linear application} \quad \rightarrow \quad \text{matrices}$$

$$\mathscr{L}:\left\{\begin{array}{lll}\alpha & = & 2a+3h\\ \gamma & = & h-5a\end{array}\right. \quad \rightarrow \quad \mathbf{A}=\left(\begin{array}{cc}2 & 3\\ 1 & -5\end{array}\right)$$

We have a matrix algebra and Matlab.
We have: $\text{var}(\mathscr{L})=\{a,h\}$, $\text{covar}(\mathscr{L})=\{\alpha,\gamma\}$.
But we cannot write: $\text{var}(\mathbf{A})=\{a,h\}$, $\text{covar}(\mathbf{A})=\{\alpha,\gamma\}$.

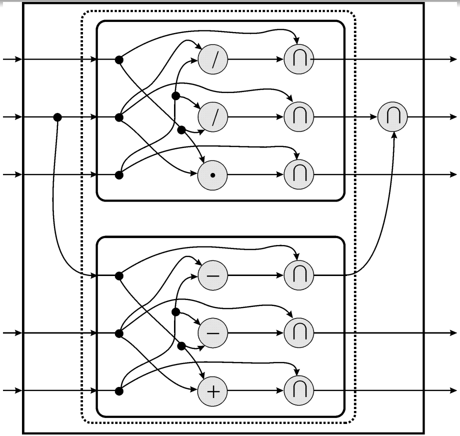
constraint	$\rightarrow$	contractor
$a \cdot b = z$	$\rightarrow$	

Contractor fusion

$$\left\{ \begin{array}{lcl} a \cdot b = z & \rightarrow & \mathscr{C}_1 \\ b + c = d & \rightarrow & \mathscr{C}_2 \end{array} \right.$$

Since b occurs in both constraints, we fuse the two contractors as:

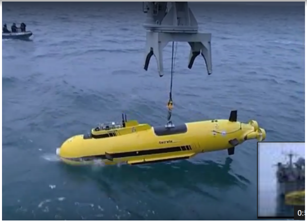
$$\begin{array}{rcl} \mathscr{C} & = & \mathscr{C}_1 \times \mathscr{C}_2 \rfloor_{(2,1)} \\ & = & \mathscr{C}_1 | \mathscr{C}_2 \text{ (for short)} \end{array}$$



SLAM

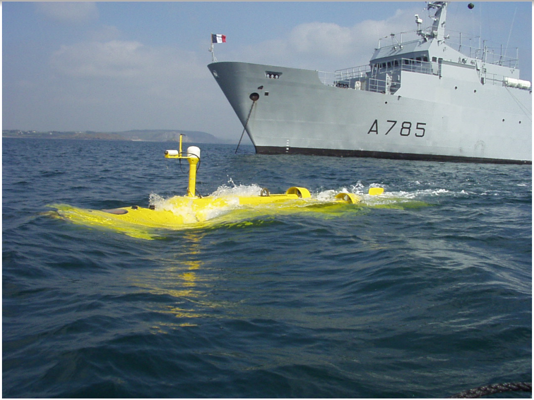
Formalisation [2]

$$\begin{cases} \dot{\mathbf{x}} & = & \mathbf{f}(\mathbf{x},\mathbf{u}) & \text{(evolution equation)} \\ \mathbf{y} & = & \mathbf{g}(\mathbf{x},\mathbf{u}) & \text{(observation equation)} \\ \mathbf{z}_i & = & \mathbf{h}(\mathbf{x},\mathbf{u},\mathbf{m}_i) & \text{(mark equation)} \end{cases}$$



Redermor (GESMA, Brest)

https://youtu.be/X0IqZxb-tFs

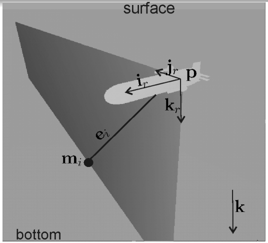

A785

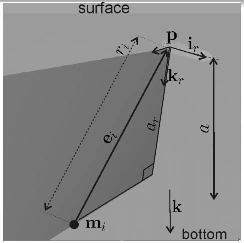
Sensors

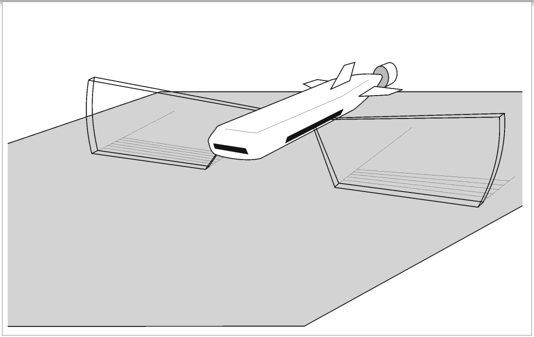
GPS (Global positioning system), only at the surface.

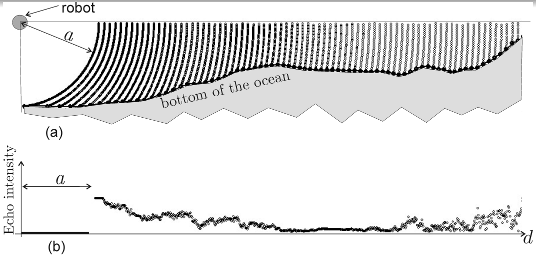
$$t_0 = 0000 \text{ s}, \quad \ell^0 = (-4.4582279°, 48.2129206°) \pm 2.5m$$
$$t_f = 6000 \text{ s}, \quad \ell^f = (-4.4546607°, 48.2191297°) \pm 2.5m$$

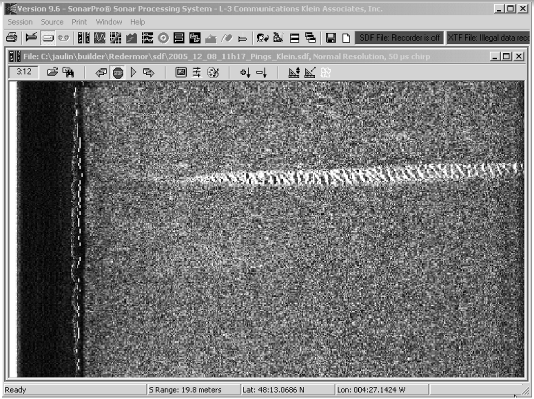
Sonar (KLEIN 5400 side scan sonar).


surface
bottom
$\mathbf{j}_r$
$\mathbf{p}$
$\mathbf{i}_r$
$\mathbf{k}_r$
$\mathbf{e}_i$
$\mathbf{m}_i$
$\mathbf{k}$

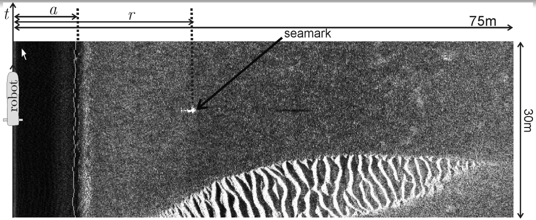

surface
$\mathbf{p}$
r_i
$\mathbf{i}_r$
$\mathbf{k}_r$
a_r
a
$\mathbf{e}_i$
$\mathbf{k}$
$\mathbf{m}_i$
bottom




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



Screenshot of SonarPro



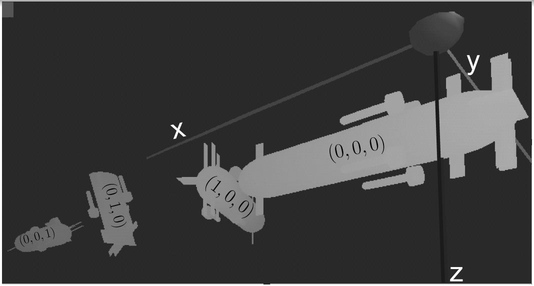
Mine detection with SonarPro

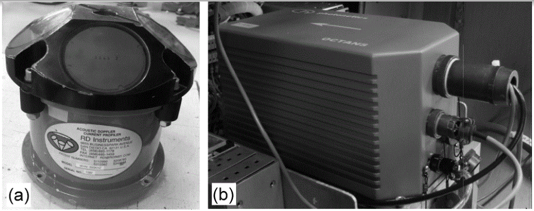
Loch-Doppler returns the speed robot $\mathbf{v}_r$.

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

Inertial central (Octans III from IXSEA).

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


y
x
(0, 0, 0)
(1, 0, 0)
(0, 1, 0)
(0, 0, 1)
z


(a)
ACOUSTIC DOPPLER
CURRENT PROFILER
RD Instruments
(b)
OCTANS

Six mines have been detected.

i	0	1	2	3	4	5
$\tau(i)$	1054	1092	1374	1748	3038	3688
$\sigma(i)$	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
4024	4817	5172	5232	5279	5688
4	3	3	4	5	1
37.90	36.71	37.37	31.03	33.51	15.05

Constraint Network

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

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

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

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

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

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

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

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

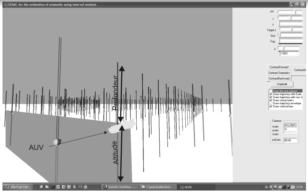
$$\dot{\mathbf{p}}(t)=\mathbf{R}(t).\mathbf{v}_{r}(t),$$

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

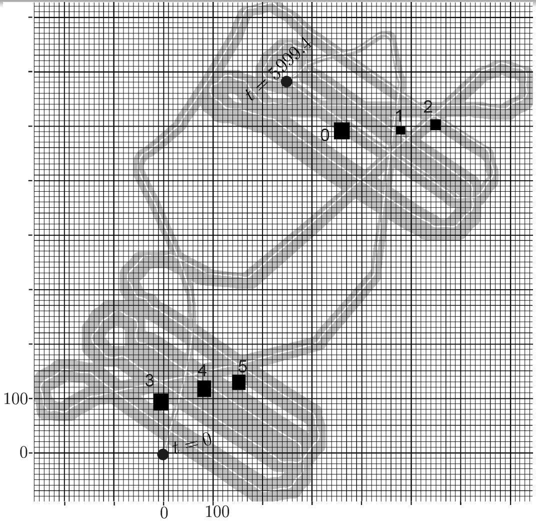
$$\mathbf{R}^{\mathsf{T}}(\tau(i))\,(\mathbf{m}(\sigma(i))-\mathbf{p}(\tau(i)))\in[0]\times[0,\infty]^{\times 2},$$

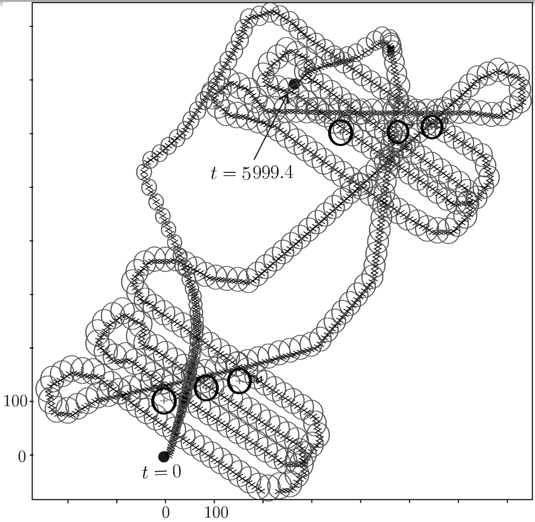
$$m_z(\sigma(i))-p_z(\tau(i))-a(\tau(i))\in[-0.5,0.5]$$

GESMI



https://youtu.be/IzJtAfAT7h4

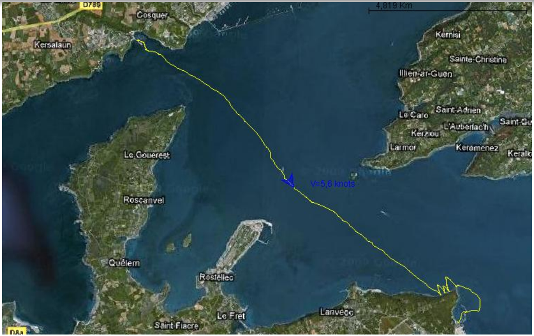

t = 5999.4
0
1
2
3
4
5
t = 0
100
0
0
100

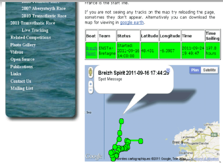

$t = 5999.4$
100
0
$t = 0$
0
100

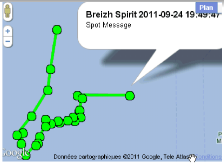
Dynamical systems

Saiboat robots


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=5,8 knots
Roscanvel
Quélern
Rostellec
Le Fret
Lanvéoc
Saint-Fiacre


2007 Aberystwyth Race
2010 Transatlantic Race
2011 Transatlantic Race
Live Tracking
Related Competitions
Photo Gallery
Videos
Open Source
Publications
Links
Contact Us
Mailing List
France is the start line.
If you are not seeing any tracks on the map try reloading the page, sometimes they don't appear. Alternatively you can download the map for viewing in google earth.
Boat | Team | Status | Latitude | Longitude | Time | Time Sailing
Breizh Spirit | ENSTA-Bretagne | Started: 2011-09-16 14:00:00 | 48.431 | -6.3907 | 2011-09-24 19:49:47 | 197.8 hours
Breizh Spirit 2011-09-16 17:44:26
Spot Message
Plan
Satellite


Breizh Spirit 2011-09-24 19:49:47
Spot Message
Plan
Données cartographiques ©2011 Google, Tele Atlas

Vaimos

Vaimos (IFREMER and ENSTA)

https://youtu.be/tmfkKNM76Qg

The robot satisfies a state equation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}).$$

With the controller $\mathbf{u} = \mathbf{g}(\mathbf{x})$, the robot satisfies an equation of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}).$$

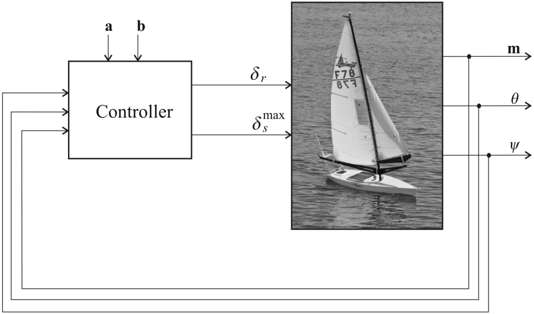
With all uncertainties, the robot satisfies

$$\dot{\mathbf{x}} \in \mathbf{F}(\mathbf{x})$$

which is *a differential inclusion*.

Controller

We follow a line following strategy [3]



Function in: $\mathbf{m},\theta,\psi,\mathbf{a},\mathbf{b}$; out: $\delta_r,\delta_s^{\max}$; inout: q

$e=\frac{\det(\mathbf{b}-\mathbf{a},\mathbf{m}-\mathbf{a})}{\|\mathbf{b}-\mathbf{a}\|}$

if $|e|>r$ then $q=\text{sign}(e)$

$\varphi=\text{atan2}(\mathbf{b}-\mathbf{a})$

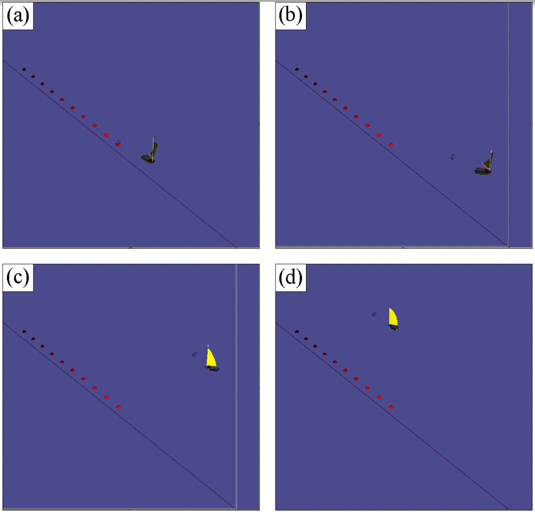
$\bar{\theta}=\varphi-\frac{1}{2}.\text{atan}\left(\frac{e}{r}\right)$

if $\cos\left(\psi-\bar{\theta}\right)+\cos\zeta<0$ then $\bar{\theta}=\pi+\psi-q.\zeta$.

$\delta_r=\text{atan}\left(\tan\frac{\theta-\bar{\theta}}{2}\right)$

$\delta_s^{\max}=\frac{\pi}{2}.\left(\frac{\cos\left(\psi-\bar{\theta}\right)+1}{2}\right)$.

Validation by simulation


(a)
(b)
(c)
(d)

Theoretical validation

When the wind is known, the sailboat with the heading controller is described by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}).$$

The system

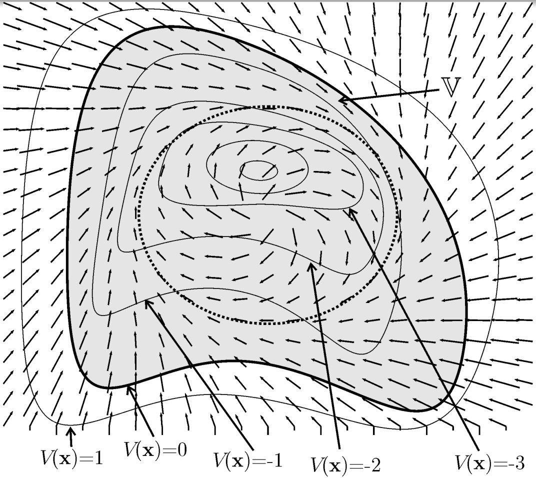
$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

is Lyapunov-stable (1892) is there exists $V(\mathbf{x}) \geq 0$ such that

$$\begin{array}{rcl} \dot{V}(\mathbf{x}) & < & 0 \text{ if } \mathbf{x} \neq \mathbf{0}, \\ V(\mathbf{x}) & = & 0 \text{ iff } \mathbf{x} = \mathbf{0}. \end{array}$$

Definition. Consider a differentiable function $V(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$. The system is V-stable if

$$\left(V(\mathbf{x}) \geq 0 \;\Rightarrow\; \dot{V}(\mathbf{x}) < 0 \right).$$


$\mathbb{V}$
$V(\mathbf{x})$=1
$V(\mathbf{x})$=0
$V(\mathbf{x})$=-1
$V(\mathbf{x})$=-2
$V(\mathbf{x})$=-3

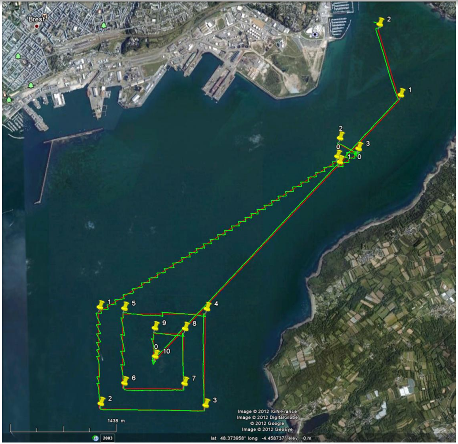
Theorem. We have

$$\left\{ \begin{array}{c} \frac{\partial V}{\partial \mathbf{x}}(\mathbf{x}).\mathbf{f}(\mathbf{x}) \geq 0 \\ V(\mathbf{x}) \geq 0 \end{array} \right. \text{ inconsistent } \Leftrightarrow \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \text{ is } V\text{-stable.}$$

Interval method could easily prove the V-stability.

Experimental validation

Brest


Brest
Image © 2012 IGN-France
Image © 2012 DigitalGlobe
© 2012 Google
Image © 2012 GeoEye
1438 m
2003
lat 48.373958° long -4.458737° élév 0 m

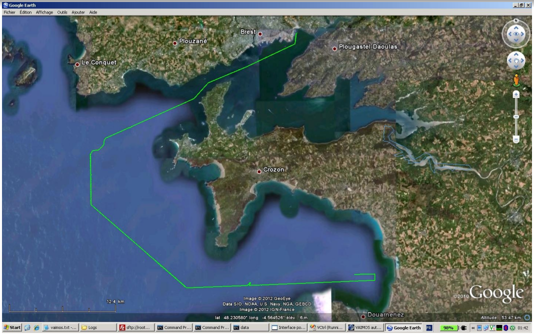
Brest-Douarnenez. January 17, 2012, 8am

Vaimos (IFREMER and ENSTA)

https://youtu.be/XxQ_KWI1q74

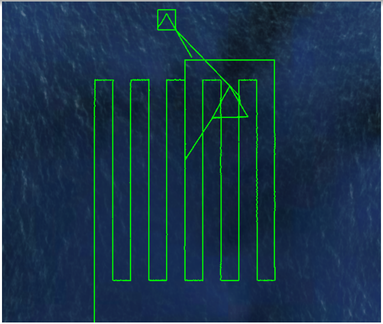

F7
8


miniji
F78


Google Earth
Fichier Édition Affichage Outils Ajouter Aide
Brest
Plouzané
Plougastel-Daoulas
Le Conquet
Crozon
Douarnenez
12.4 km
Image © 2012 GeoEye
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Image © 2012 IGN-France
©2010 Google
Altitude 53.47 km

Middle of Atlantic ocean

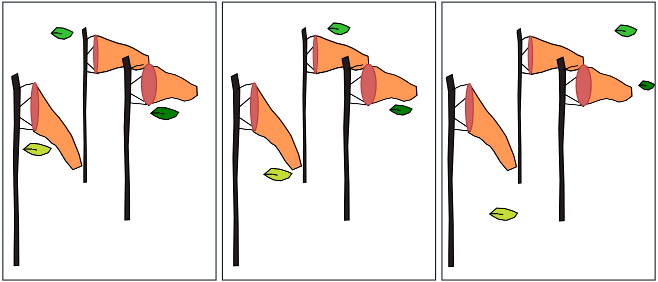
https://youtu.be/pb_KhcYZI_A



350 km made by Vaimos in 53h, September 6-9, 2012.

Eulerian state estimation

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$$



Leaves: Lagrangian view of the wind; Flags: an Eulerian view [8]

Eulerian state estimation [7] can be formalized as:

$$\begin{array}{llc} (i) & \dot{\mathbf{x}}(t)=\mathbf{f}(\mathbf{x}(t)) & \text{(evolution)} \\ (ii) & \mathbf{x}(t_i)\in\mathbb{X}_i\subset\mathbb{R}^n & \text{(event)} \\ (iii) & \forall(i,j)\in\mathbb{J}, t_i\leq t_j & \text{(precedence)} \end{array}$$

Bracket the set $\mathbb{X}\subset\mathbb{R}^n$ of all feasible $\mathbf{x}(t)$.

Invariant sets

Denote by φ the flow map of our system, *i.e.*, with $\mathbf{x}_0 = \mathbf{x}(0)$, the system reaches $\varphi(t, \mathbf{x}_0)$ at time t.

A set $\mathbb{A}$ is *positive invariant* if

$$\mathbf{x} \in \mathbb{A}, t \geq 0 \Longrightarrow \varphi(t, \mathbf{x}) \in \mathbb{A}.$$

The set of all positive invariant sets is a lattice, *i.e.*, the union and the intersection are closed.
Thus, the notion of *largest positive invariant set* contained in $\mathbb{X}$ can be defined.

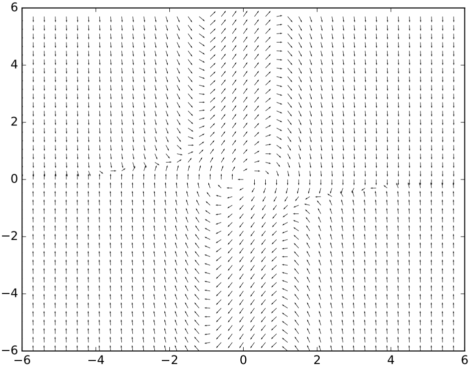
The largest positive invariant set included in $\mathbb{X}$ is:

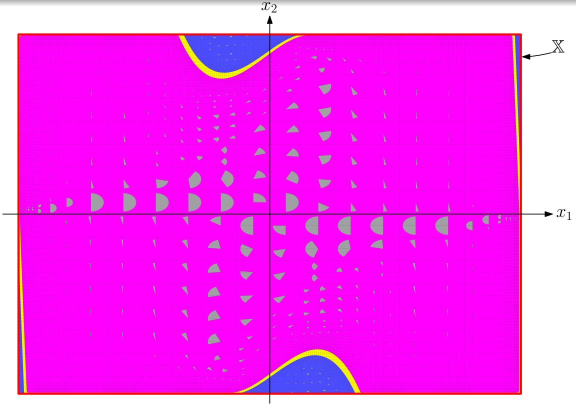
$$Inv^{+}(\mathbf{f},\mathbb{X}) = \{\mathbf{x}_0 \mid \forall t \geq 0, \varphi(t,\mathbf{x}_0) \in \mathbb{X}\}.$$

Mazes allow us to compute an inner and an outer approximation for $Inv^{+}(\mathbf{f},\mathbb{X})$.

As an illustration, consider the Van der Pol system:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \left(1 - x_1^2\right) \cdot x_2 - x_1 \end{cases}$$


6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6



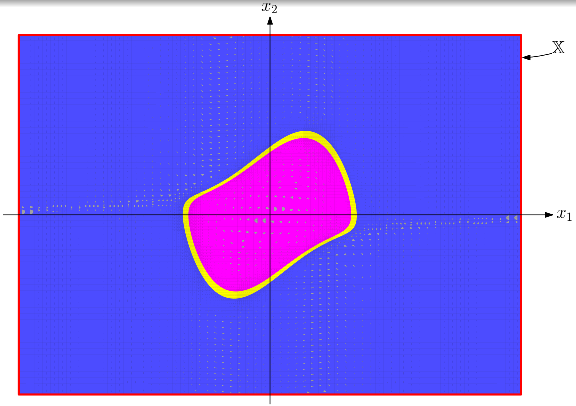
Largest positive invariant set $Inv^+(\mathbf{f}, \mathbb{X})$

Largest negative invariant set.

$$Inv^{-}(\mathbf{f},\mathbb{X})=\{\mathbf{x}_0 \mid \forall t\leq 0, \varphi(t,\mathbf{x}_0)\in\mathbb{X}\}.$$

We have

$$Inv^{-}(\mathbf{f},\mathbb{X})=Inv^{+}(-\mathbf{f},\mathbb{X}).$$



Largest negative invariant set $Inv^{-}(\mathbf{f}, \mathbb{X})$

Largest invariant set

$$Inv(\mathbf{f},\mathbb{X}) = \{\mathbf{x}_0 \mid \forall t \in \mathbb{R}, \varphi(t,\mathbf{x}_0) \in \mathbb{X}\}.$$

We have

$$Inv(\mathbf{f},\mathbb{X}) \;=\; Inv^{+}(-\mathbf{f},\mathbb{X}) \cap Inv^{+}(\mathbf{f},\mathbb{X}).$$

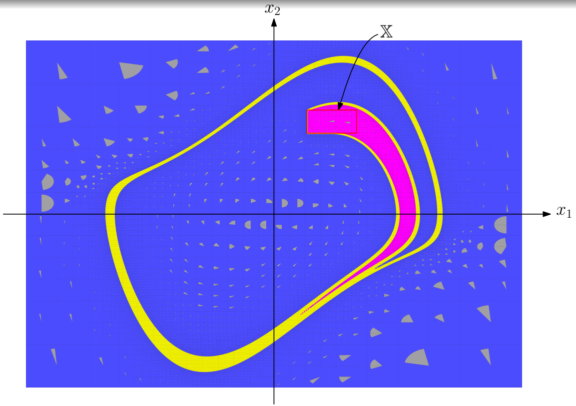
Thus $Inv(\mathbf{f},\mathbb{X})$ can be defined in terms of largest positive invariant sets.

Forward reach set

$$Forw(\mathbf{f},\mathbb{X}) = \{\mathbf{x} \mid \exists t \geq 0, \exists \mathbf{x}_0 \in \mathbb{X}, \varphi(t,\mathbf{x}_0) = \mathbf{x}\}.$$

We have

$$Forw(\mathbf{f},\mathbb{X}) \quad = \quad \overline{Inv^+(-\mathbf{f},\overline{\mathbb{X}})} \ .$$



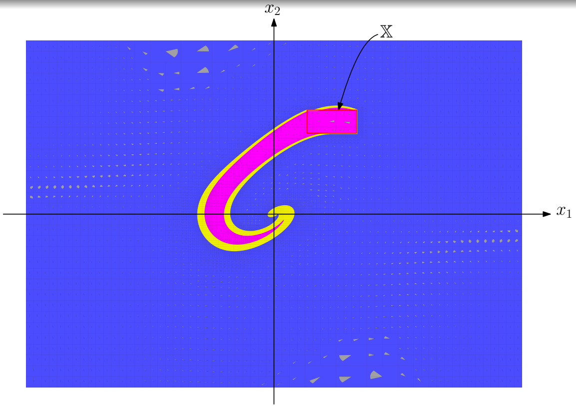
$$Forw(\mathbf{f}, \mathbb{X}) \text{ for } \mathbb{X} = [0.4, 1.0] \times [1.4, 1.8]$$

Backward reach set.

$$Back(\mathbf{f},\mathbb{X}) = \{\mathbf{x}_0 \mid \exists t \geq 0, \varphi(t,\mathbf{x}_0) \in \mathbb{X}\}.$$

Since

$$Back(\mathbf{f},\mathbb{X}) \quad = \quad \overline{Inv^{+}(\mathbf{f},\overline{\mathbb{X}})} \;.$$

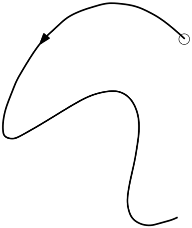


$Back(\mathbf{f}, \mathbb{X})$ for $\mathbb{X} = [0.4, 1.0] \times [1.4, 1.8]$.

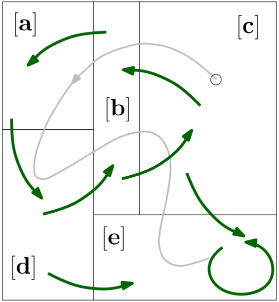
Maze

An ***interval*** is *a domain* which encloses a real number.
A ***polygon*** is *a domain* which encloses a vector of $\mathbb{R}^n$.
A ***maze*** is *a domain* which encloses a path.

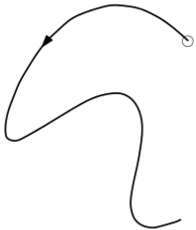
A maze is a set of paths.



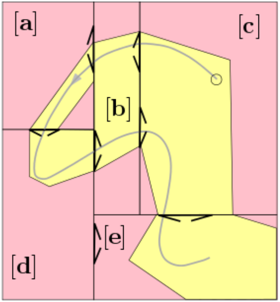
$\in$



Mazes can be made more accurate:



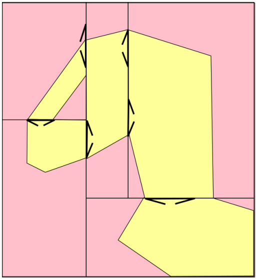
$\in$



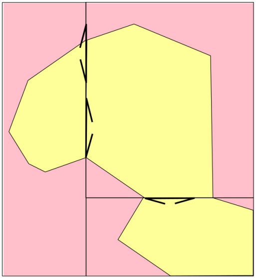
Here, a **maze** $\mathscr{L}$ is composed of [6][5].

- A paving $\mathscr{P}$
- Doors between adjacent boxes

The set of mazes forms a lattice with respect to $\subset$.



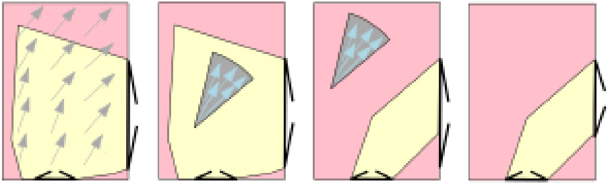
$\subset$



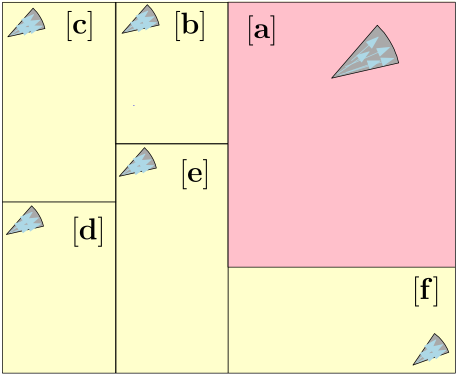
Inner approximation

Target contractor. If a box $[\mathbf{x}]$ of $\mathscr{P}$ is outside $\mathbb{X}$ then remove $[\mathbf{x}]$ and close all doors entering in $[\mathbf{x}]$.

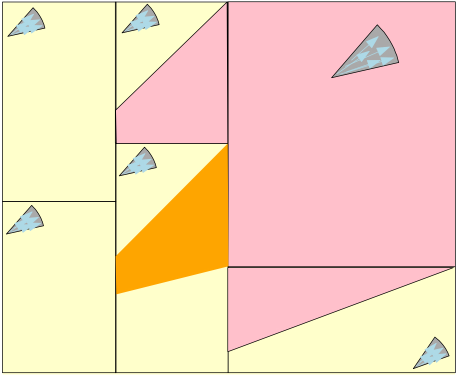
Flow contractor. For each box [x] of $\mathscr{P}$, we contract using the constraint $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$.



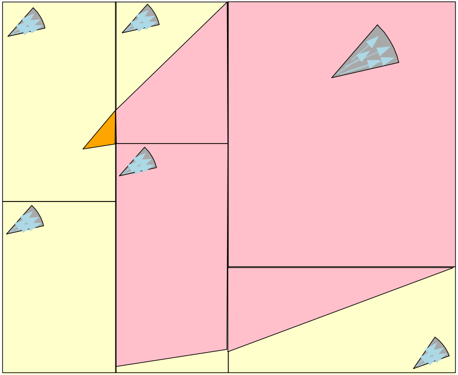
Propagation



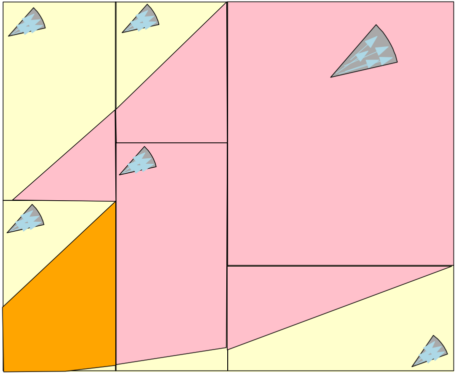
Yellow area: $\mathbb{X}$



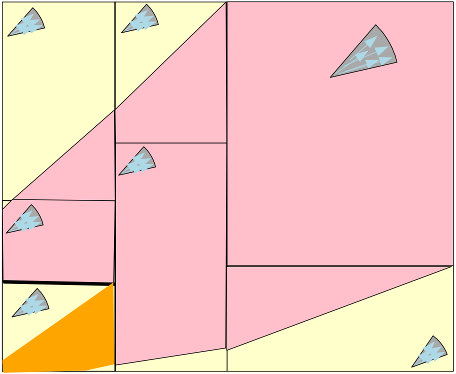
The red parts have been deleted



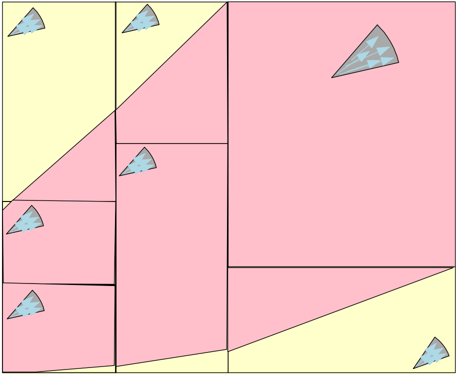
The yellow area is contracted



At each step, the yellow area encloses $Inv^+(\mathbb{X})$

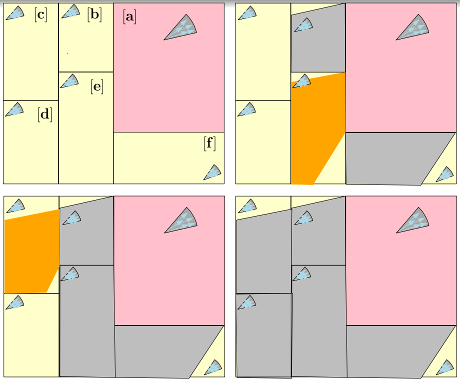


At each step, the red area is outside $Inv^{+}(\mathbb{X})$



The yellow area encloses $Inv^{+}(\mathbb{X})$

Inflation propagation



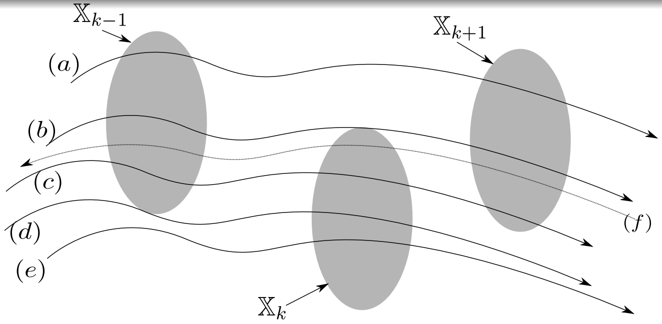
An interpretation can be given only when the fixed point is reached.
The yellow area is an inner approximation of $Inv^{+}(\mathbb{X})$

Eulerian filter

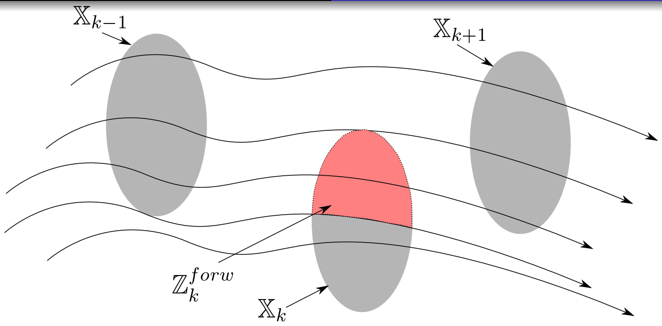
Define ℓ sets $\mathbb{X}_0, \mathbb{X}_1, \dots, \mathbb{X}_\ell$ of the state space. Define $\mathbb{Z}_k^{forw}$ the set of all state vectors $\mathbf{x}(t)$ inside $\mathbb{X}_k$ that have visited $\mathbb{X}_0, \mathbb{X}_1, \dots, \mathbb{X}_{k-1}$. We have

$$\mathbb{Z}_{k+1}^{forw} = Forw\left(\mathbb{Z}_k^{forw}\right) \cap \mathbb{X}_{k+1}$$

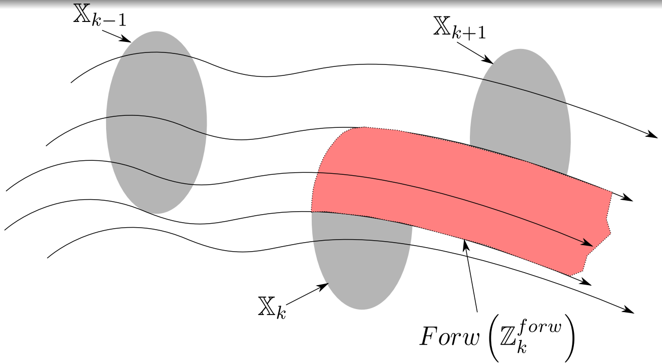
with $\mathbb{Z}_0^{forw} = \mathbb{X}_0$.



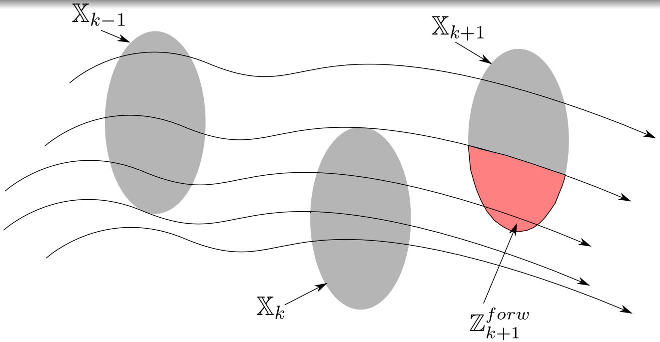
The trajectories (b),(c) are consistent with the sets $\mathbb{X}_{k-1}, \mathbb{X}_k, \mathbb{X}_{k+1}$



Set $\mathbb{Z}_k^{forw}$ of all $\mathbf{x}(t)$ in $\mathbb{X}_k$ that have already visited $\mathbb{X}_0, \mathbb{X}_1, \dots, \mathbb{X}_{k-1}$



$Forw\left(\mathbb{Z}_k^{forw}\right)$ corresponds to all states $\mathbf{x}(t)$ that have visited $\mathbb{X}_0, \mathbb{X}_1, \dots, \mathbb{X}_k$



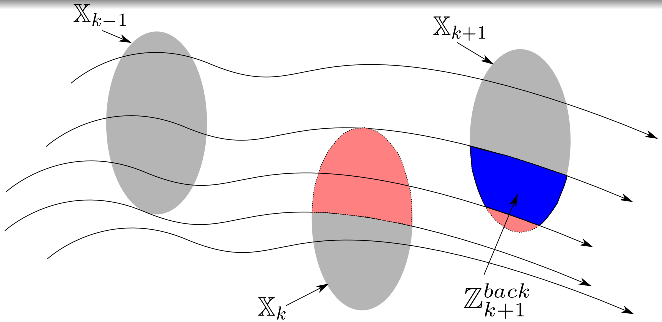
Set $\mathbb{Z}_{k+1}^{forw}$ of all states $\mathbf{x}(t)$ in $\mathbb{X}_{k+1}$ that have already visited $\mathbb{X}_0, \mathbb{X}_1, \dots, \mathbb{X}_k$

Eulerian smoother

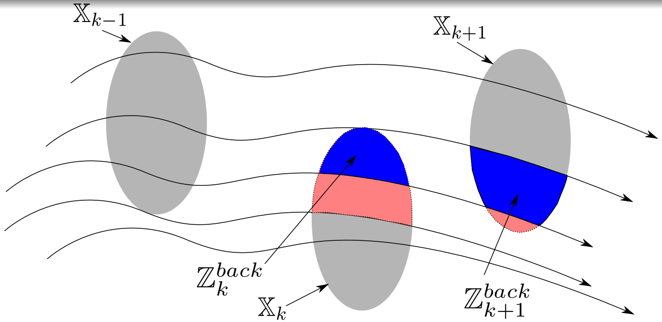
Define the set $\mathbb{Z}_k^{back}$ of all states $\mathbf{x}(t)$ inside $\mathbb{X}_k$ that have visited $\mathbb{X}_0, \mathbb{X}_1, \dots, \mathbb{X}_{k-1}$ in the past and will visit $\mathbb{X}_{k+1}, \dots, \mathbb{X}_\ell$ in the future. We have

$$\mathbb{Z}_k^{back} = Back\left(\mathbb{Z}_{k+1}^{back}\right) \cap \mathbb{Z}_k^{forw}$$

with $\mathbb{Z}_\ell^{back} = \mathbb{Z}_\ell^{forw}$. The will be called the *Eulerian smoother.*



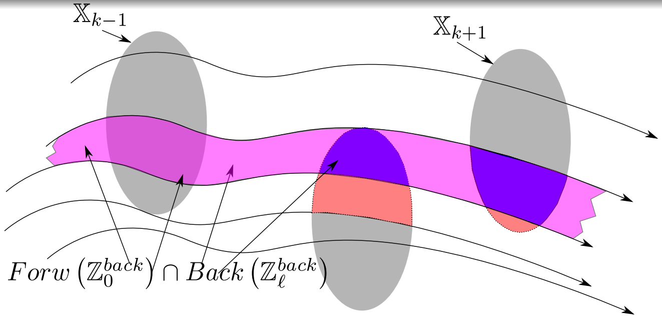
Set $\mathbb{Z}_{k+1}^{back}$ of all states $\mathbf{x}(t)$ inside $\mathbb{Z}_{k+1}^{forw}$ that will visit $\mathbb{X}_{k+2}, \ldots, \mathbb{X}_\ell$



Set $\mathbb{Z}_k^{back}$ of all states $\mathbf{x}(t)$ inside $\mathbb{Z}_k^{forw}$ that will visit $\mathbb{X}_{k+1}, \dots, \mathbb{X}_\ell$

The set of trajectories that started inside $\mathbb{X}_0$ and visited the sets $\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_{\ell-1}$sequentially, and that ended in $\mathbb{X}_\ell$ can thus be enclosed by

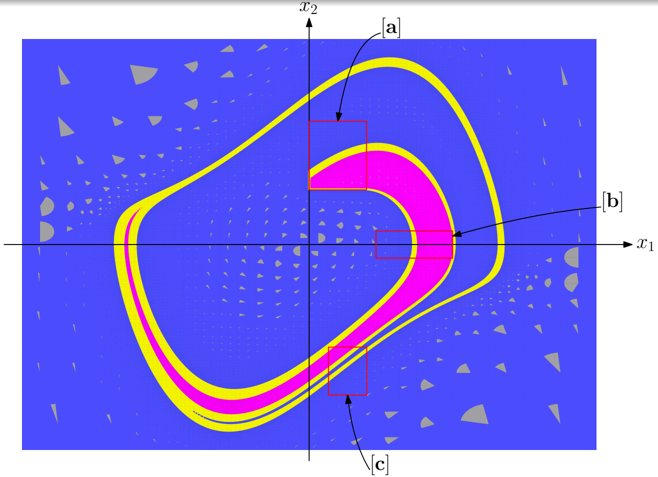
$$Forw\left(\mathbb{Z}_0^{back}\right) \cap Back\left(\mathbb{Z}_\ell^{back}\right).$$



Set $Forw\left(\mathbb{Z}_0^{back}\right) \cap Back\left(\mathbb{Z}_\ell^{back}\right)$ enclosing the trajectory consistent with the past and future visits

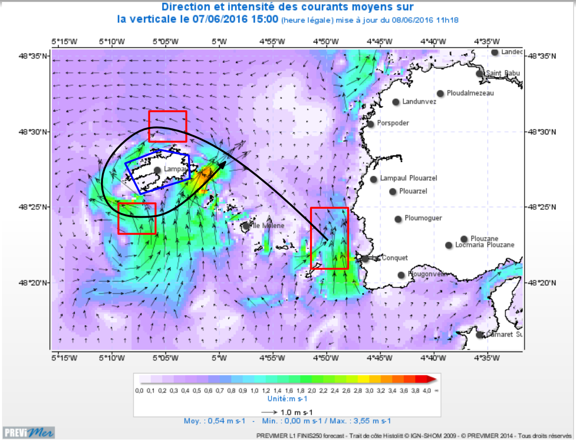
Example. Take the Van der Pol system with

$$\begin{aligned} \mathbb{X}_0 &= [\mathbf{a}] = [0, 0.6] \times [0.8, 1.8] \\ \mathbb{X}_1 &= [\mathbf{b}] = [0.7, 1.5] \times [-0.2, 0.2] \\ \mathbb{X}_2 &= [\mathbf{c}] = [0.2, 0.6] \times [-2.2, -1.5] \end{aligned}$$



Feasible states associated to the Eulerian state estimation problem

An application of Eulerian state estimation moving taking advantage of ocean currents.



Visiting the three red boxes using a buoy that follows the currents is an Eulerian state estimation problem

G. Chabert and L. Jaulin.
Contractor Programming.
Artificial Intelligence, 173:1079–1100, 2009.

L. Jaulin.
Range-only SLAM with occupancy maps; A set-membership approach.
IEEE Transaction on Robotics, 27(5):1004–1010, 2011.

L. Jaulin and F. Le Bars.
An Interval Approach for Stability Analysis; Application to Sailboat Robotics.
IEEE Transaction on Robotics, 27(5), 2012.

L. Jaulin and E. Walter.
Guaranteed nonlinear parameter estimation from bounded-error data via interval analysis.

Mathematics and Computers in Simulation, 35(2):123–137, 1993.

T. Le Mézo L. Jaulin and B. Zerr.
Bracketing the solutions of an ordinary differential equation with uncertain initial conditions.
Applied Mathematics and Computation, 2017.

T. Le Mézo, L. Jaulin, and B. Zerr.
An interval approach to compute invariant sets.
IEEE Transaction on Automatic Control, 2017.

T. Le Mézo, L. Jaulin, and B. Zerr.
Eulerian state estimation.
In *SWIM'17, Manchester, UK*, 2017.

Ian Mitchell, Alexandre M. Bayen, and Claire J. Tomlin.
Validating a Hamilton-Jacobi Approximation to Hybrid System Reachable Sets.

In Maria Domenica Di Benedetto and Alberto Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, number 2034 in Lecture Notes in Computer Science, pages 418–432. Springer Berlin Heidelberg, 2001.