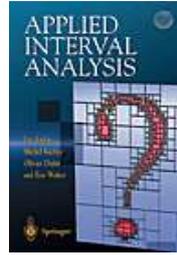


Méthodes ensemblistes pour la localisation



Luc Jaulin, Jan Sliwka, Fabrice Le Bars, Olivier Reynet
ENSTA-Bretagne, le mardi 29 mars 2011

1 Approche ensembliste

1.1 Calcul par intervalles

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

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

Par exemple,

$$\begin{aligned} [-1, 3] + [2, 5] &= [1, 8], \\ [-1, 3] \cdot [2, 5] &= [-5, 15], \\ [-1, 3] / [2, 5] &= \left[-\frac{1}{2}, \frac{3}{2}\right], \end{aligned}$$

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

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

Par exemple,

$$\sin([0, \pi]) = [0, 1],$$

$$\text{sqr}([-1, 3]) = [-1, 3]^2 = [0, 9],$$

$$\text{abs}([-7, 1]) = [0, 7],$$

$$\text{sqrt}([-10, 4]) = \sqrt{[-10, 4]} = [0, 2],$$

$$\log([-2, -1]) = \emptyset.$$

1.2 Projection de contraintes

Soient x, y, z trois variables telles que

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

Les valeurs < 2 pour x , < 1 pour y et > 9 pour z sont inconsistantes.

1.3 Méthode numérique de projection

En effet, puisque $x \in [-\infty, 5]$, $y \in [-\infty, 4]$, $z \in [6, \infty]$ et $z = x + y$, nous avons

$$z = x + y \Rightarrow z \in [6, \infty] \cap ([-\infty, 5] + [-\infty, 4]) \\ = [6, \infty] \cap [-\infty, 9] = [6, 9].$$

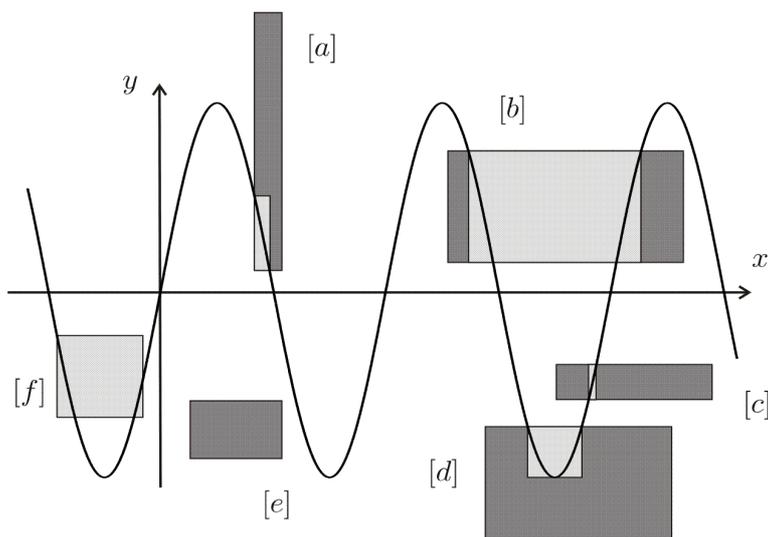
$$x = z - y \Rightarrow x \in [-\infty, 5] \cap ([6, \infty] - [-\infty, 4]) \\ = [-\infty, 5] \cap [2, \infty] = [2, 5].$$

$$y = z - x \Rightarrow y \in [-\infty, 4] \cap ([6, \infty] - [-\infty, 5]) \\ = [-\infty, 4] \cap [1, \infty] = [1, 4].$$

Pour la contrainte

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

le problème est un peu plus difficile.



1.4 Algorithme de propagation-bissection

Exemple. Cherchons à résoudre.

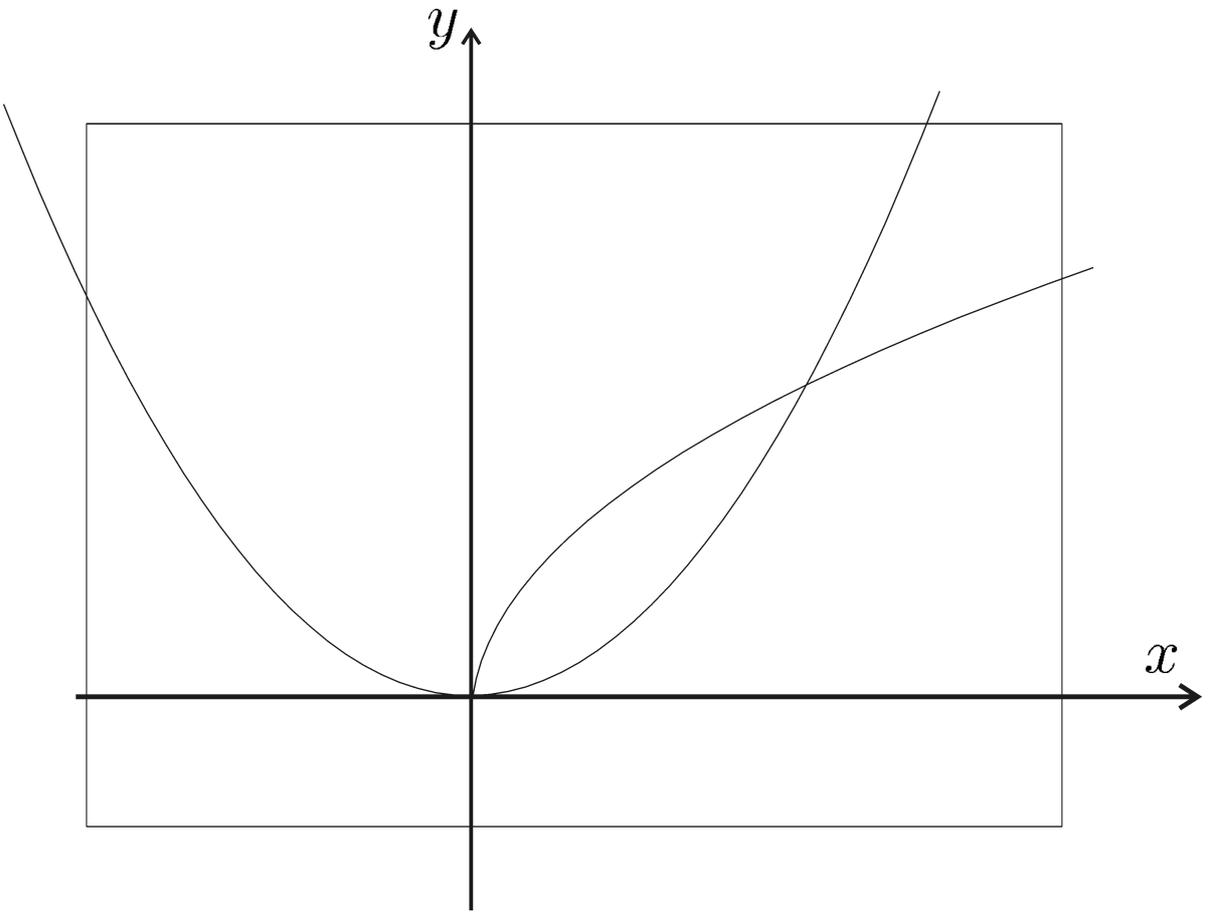
$$y = x^2$$

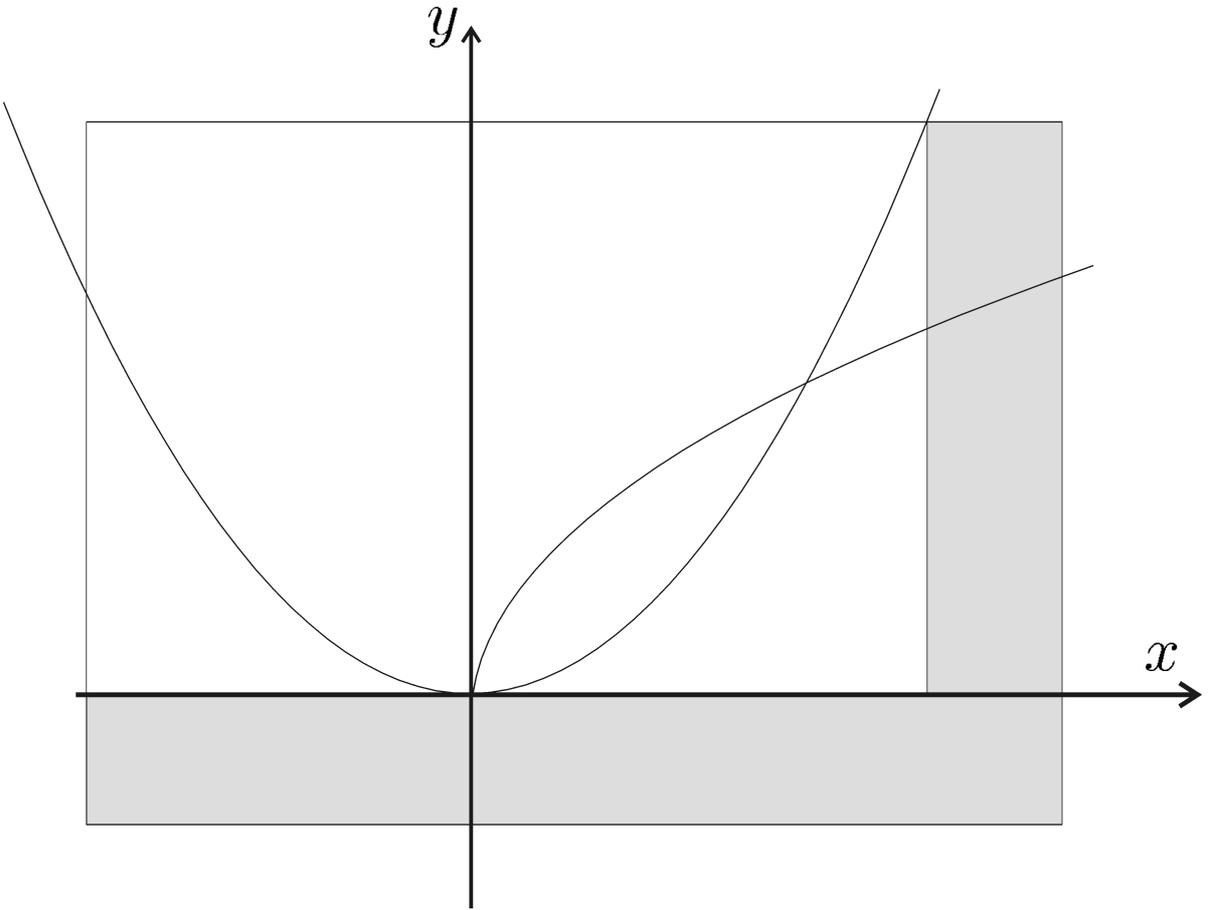
$$y = \sqrt{x}.$$

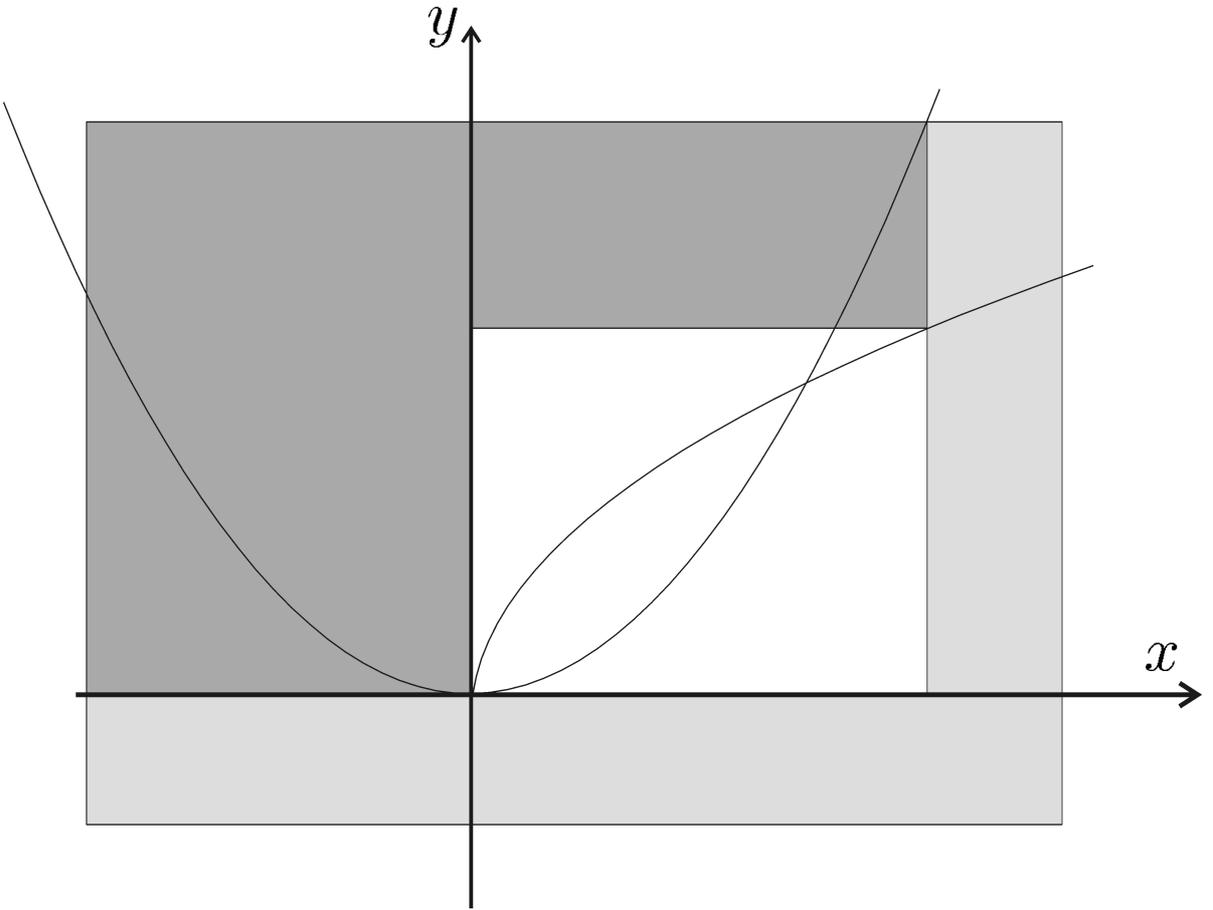
On a deux contracteurs

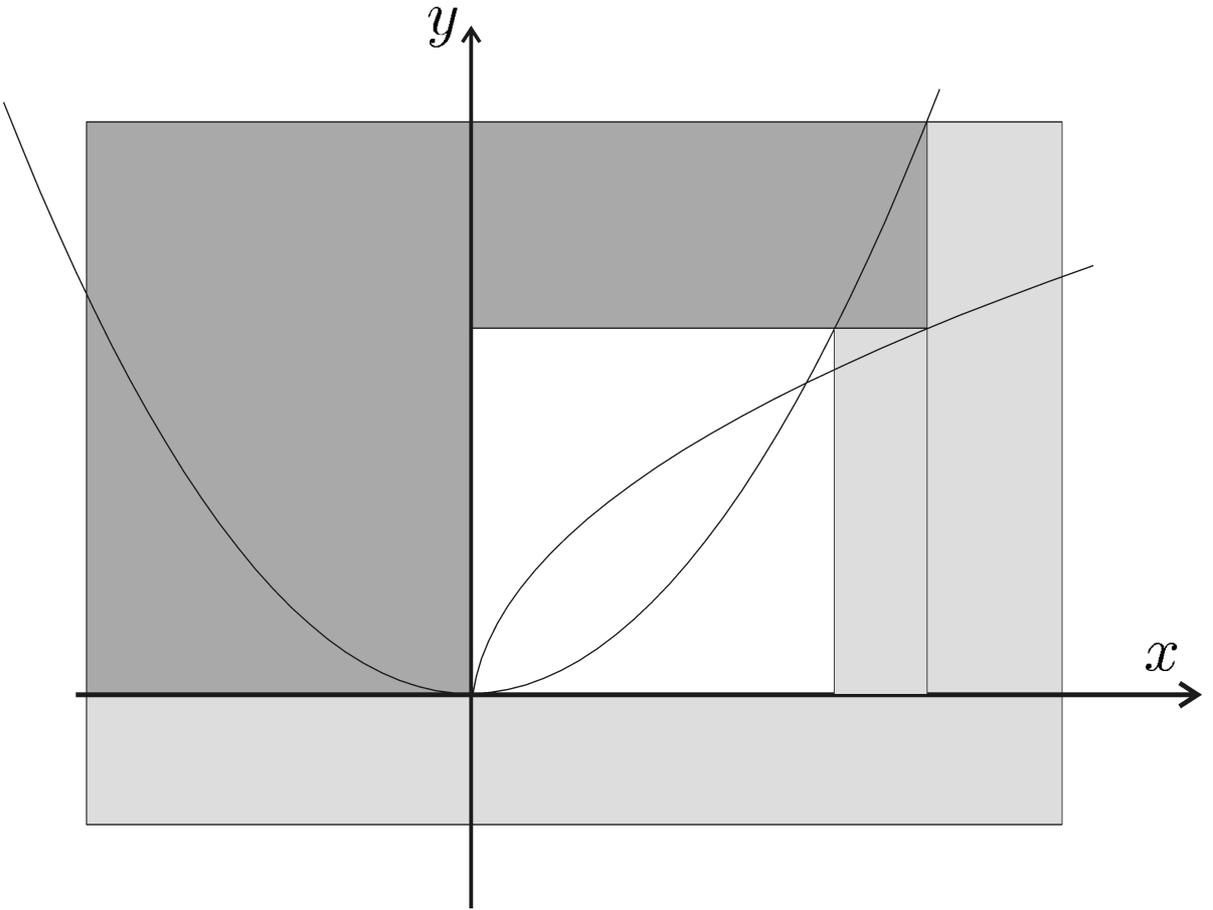
$$\mathcal{C}_1 : \begin{cases} [y] = [y] \cap [x]^2 \\ [x] = [x] \cap \sqrt{[y]} \end{cases} \text{ associé à } y = x^2$$

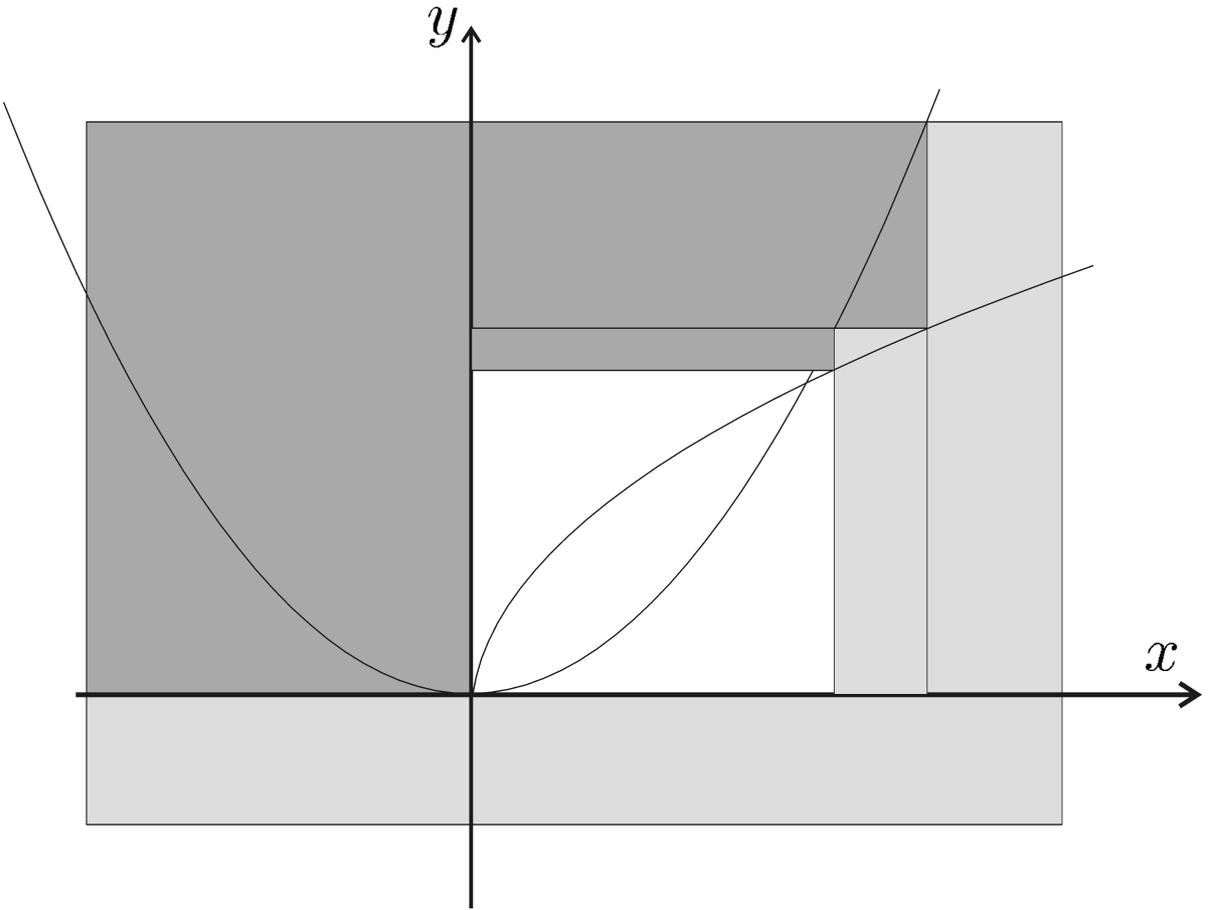
$$\mathcal{C}_2 : \begin{cases} [y] = [y] \cap \sqrt{[x]} \\ [x] = [x] \cap [y]^2 \end{cases} \text{ associé à } y = \sqrt{x}$$

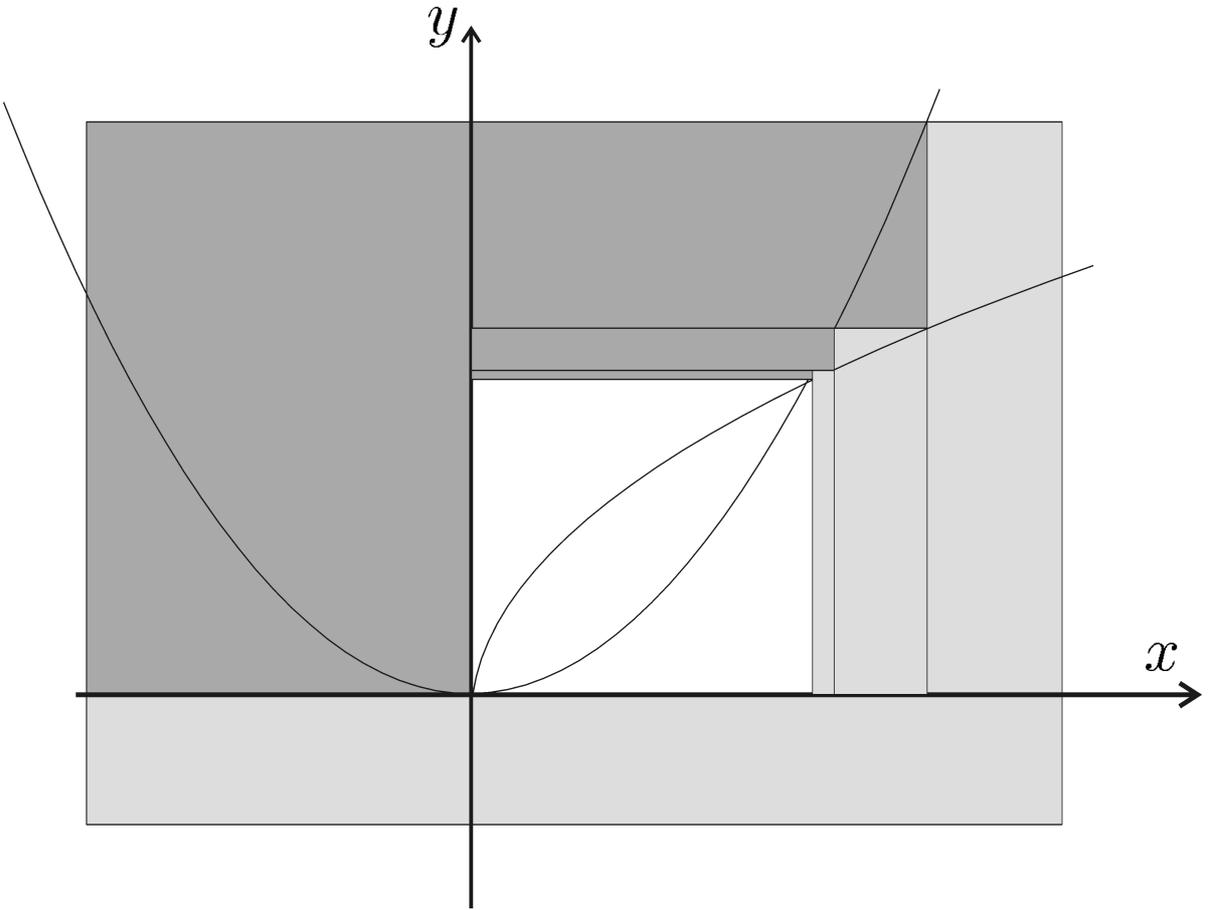


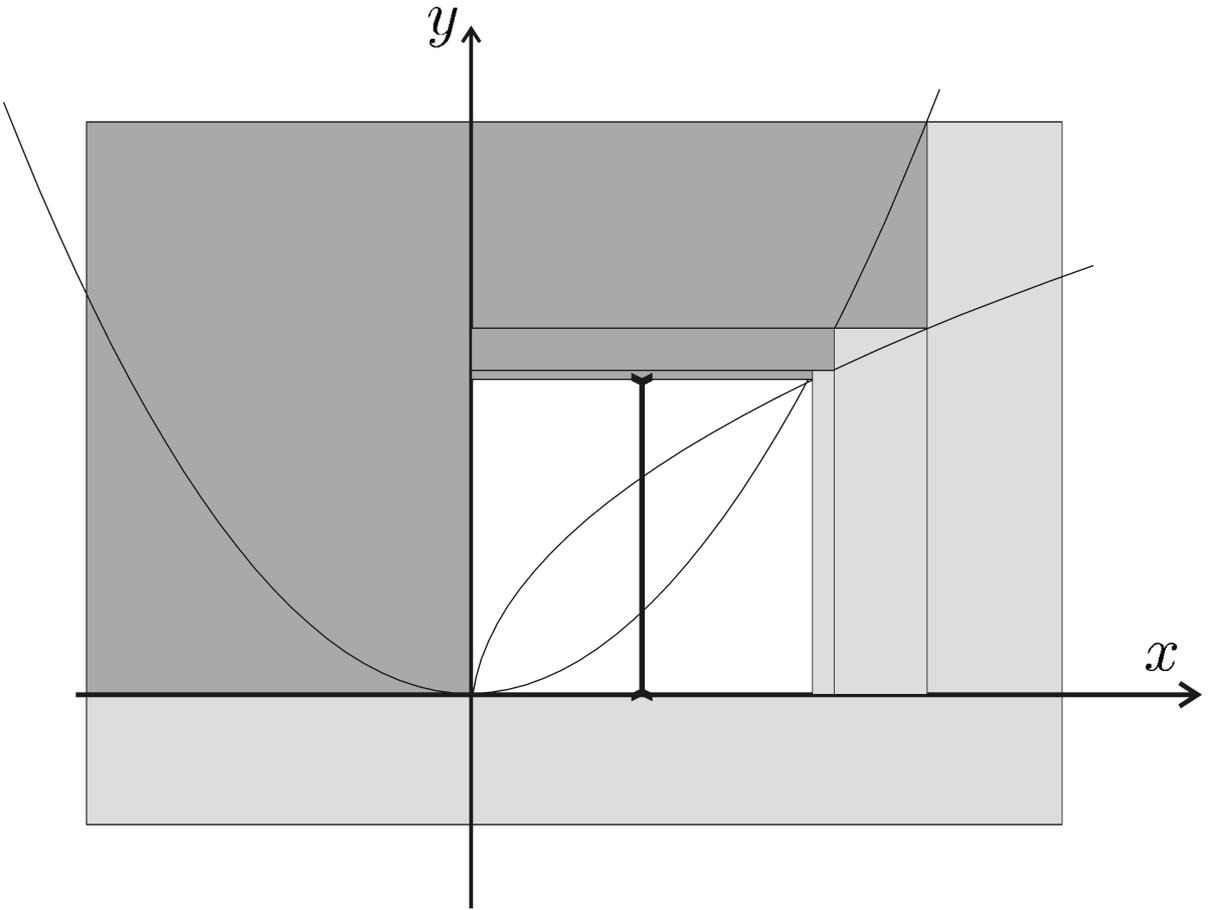


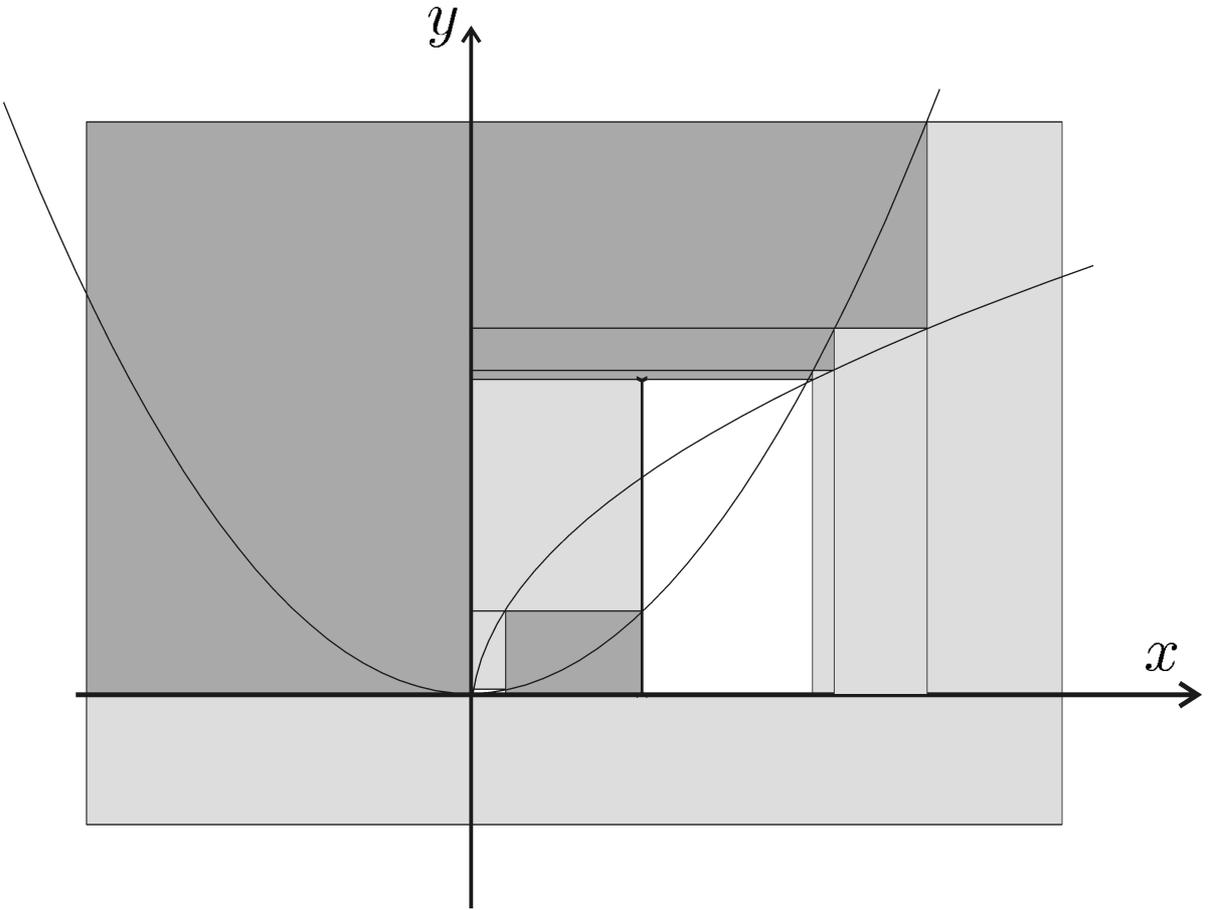


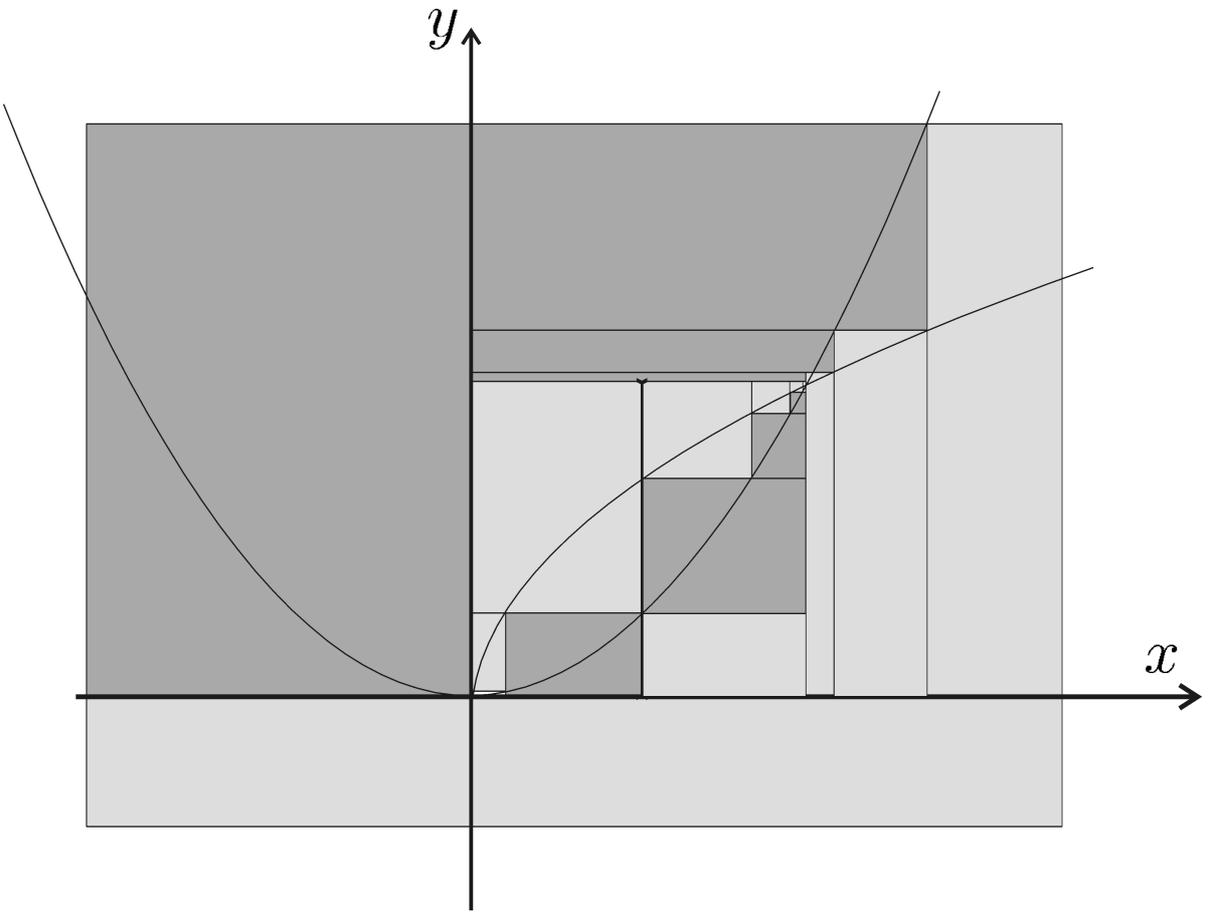












1.5 Décomposition

Pour les contraintes plus complexes, il nous faut effectuer une décomposition

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

se décompose en

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

1.6 QUIMPER

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

Quimper est un langage interprété pour le calcul ensembliste.

Un programme Quimper se décrit par un ensemble de contracteurs.

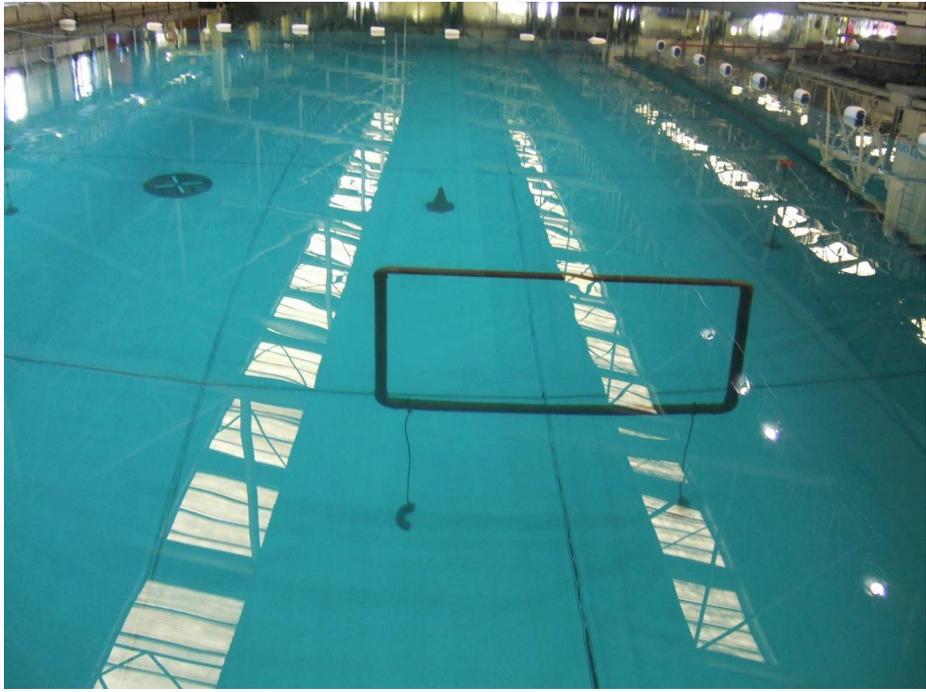
Logiciel libre disponible sur

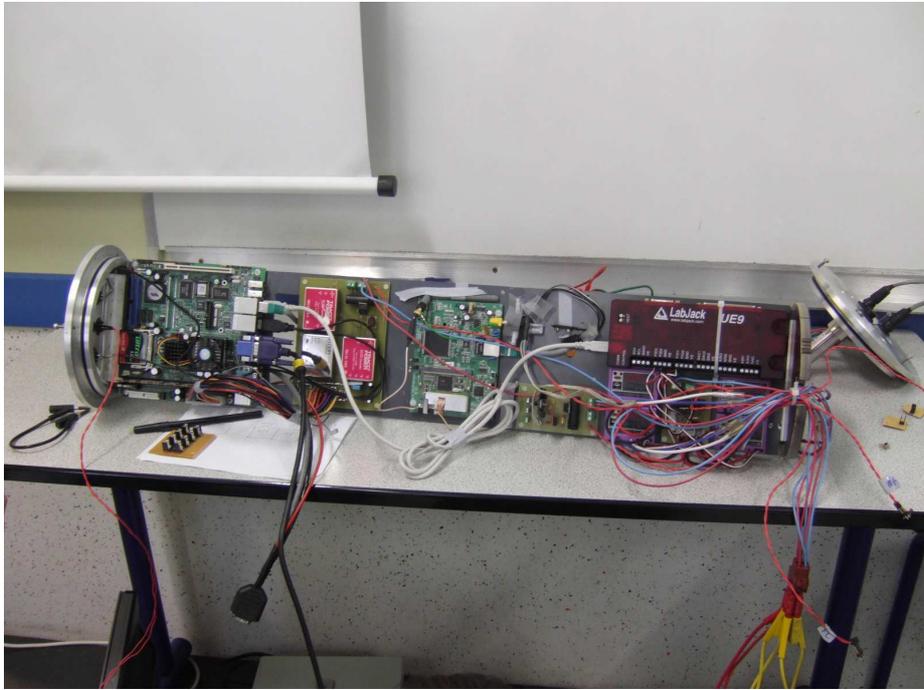
<http://ibex-lib.org/>

2 SAUCE



Portsmouth, July 12-15, 2007.

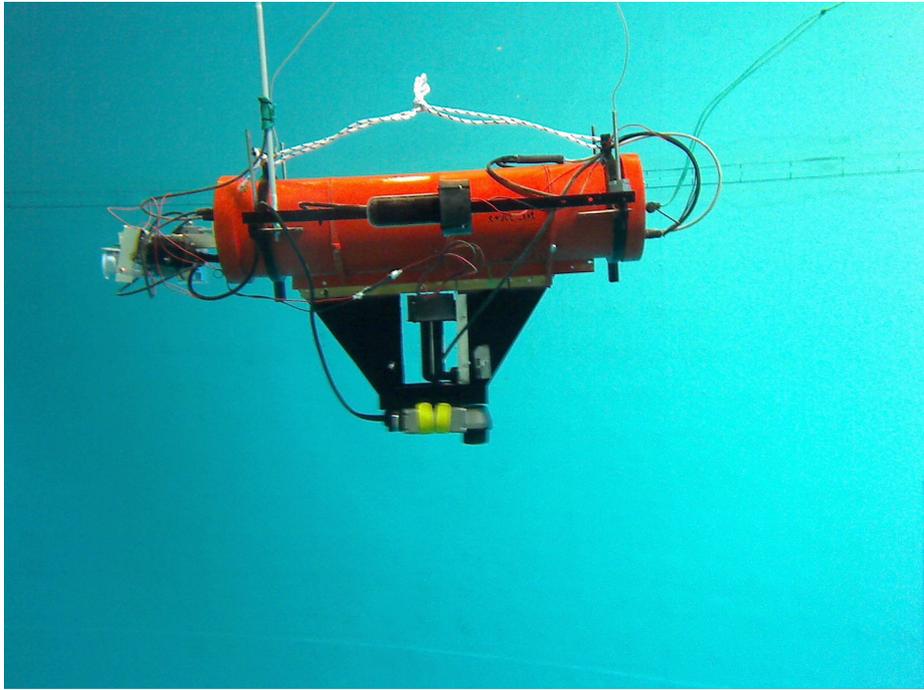




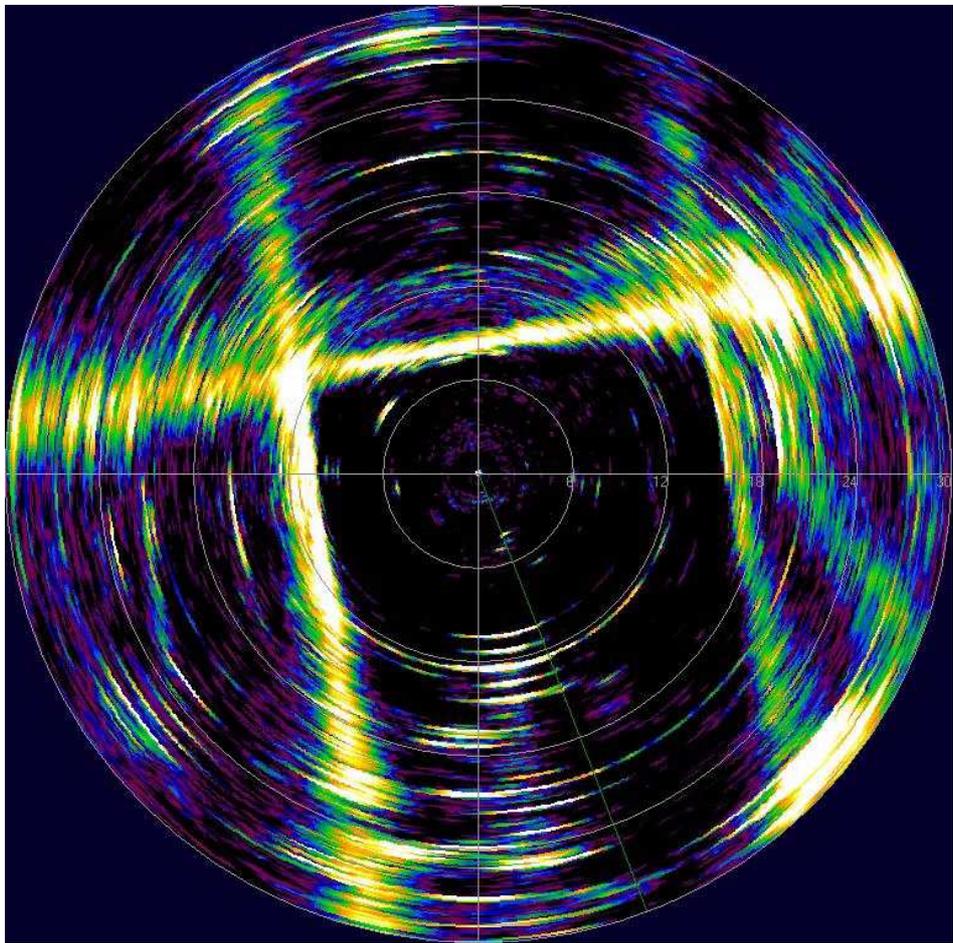


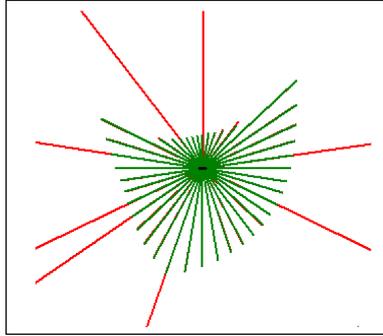
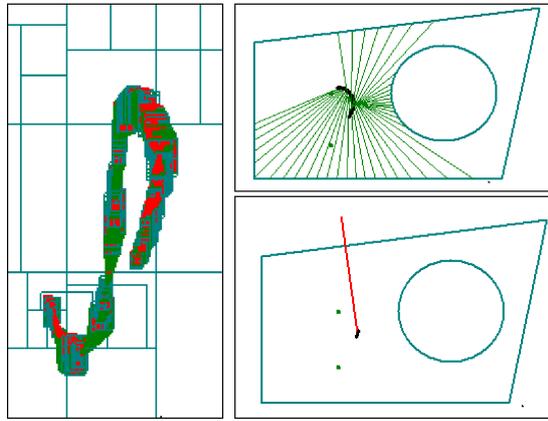






Robot Sauc'isse dans une piscine

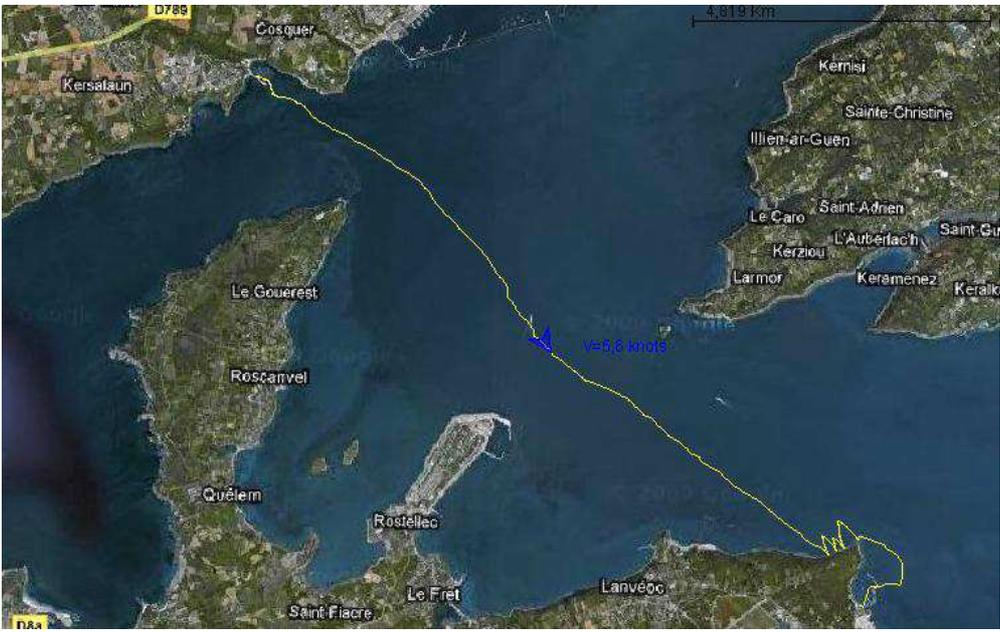




Montrer les images de la Marina

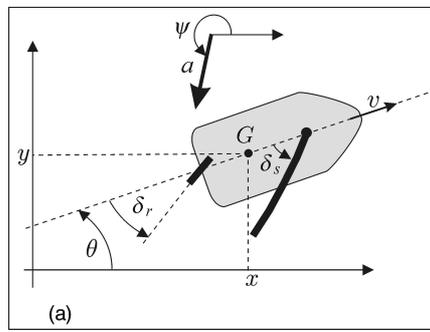
3 Robot voilier



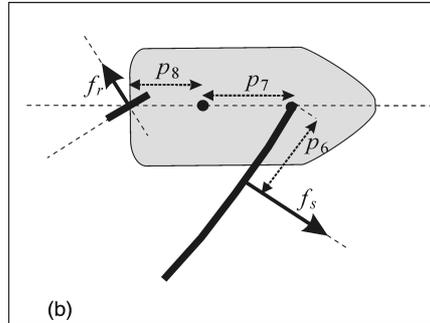


3.1 Normalized State equations

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



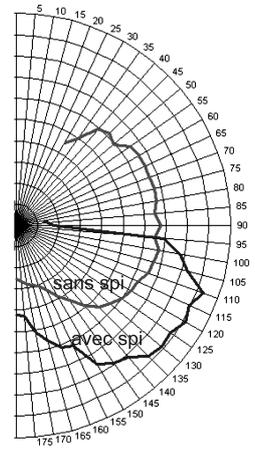
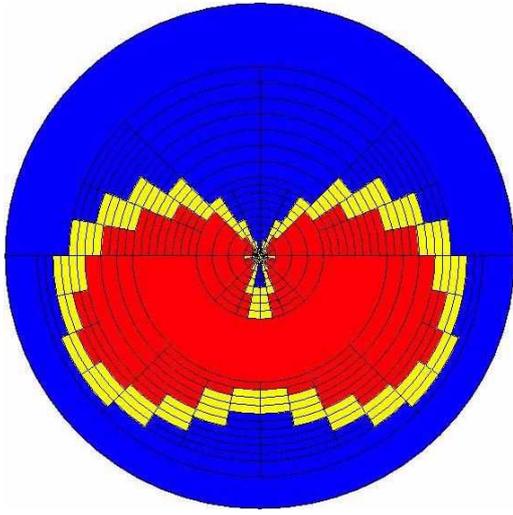
(a)

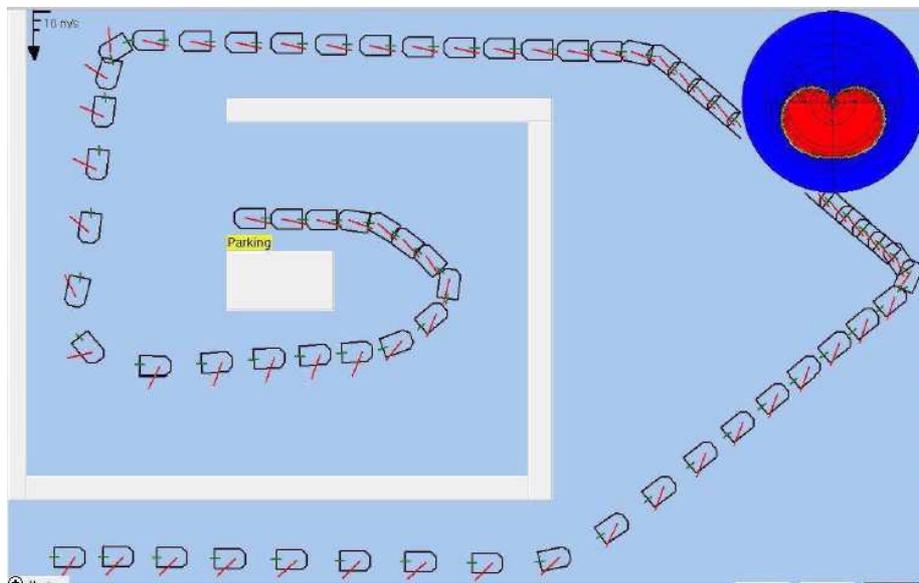


(b)

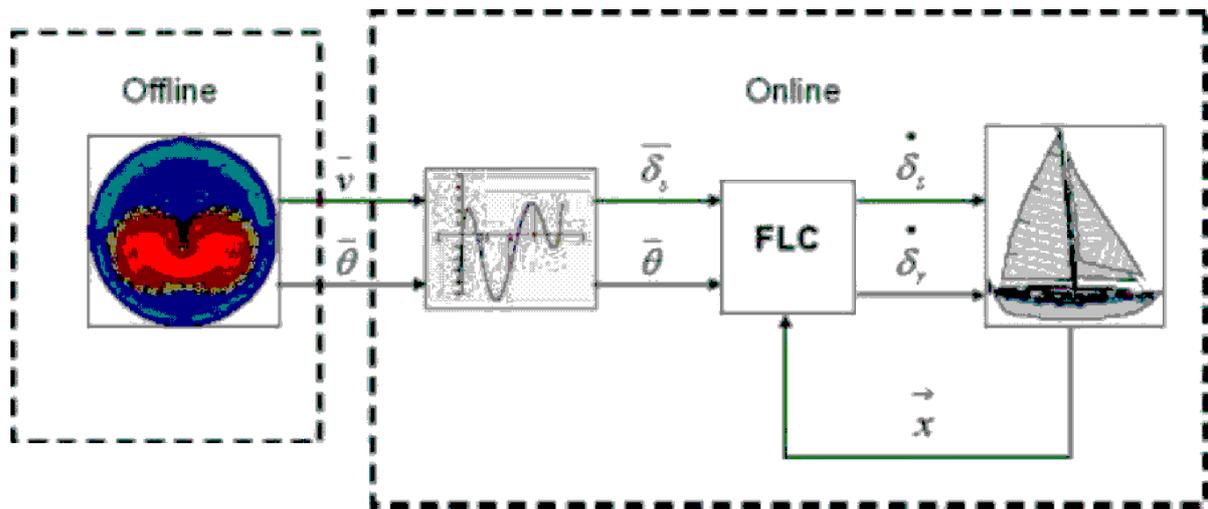
3.2 Polar speed diagram

$$\mathbb{W} = \{ (\theta, v) \mid \begin{array}{l} \exists(\omega, u_1, u_2, f_s, f_r, \delta_r, \delta_s) \\ \omega = 0, u_1 = 0, u_2 = 0 \\ f_s \sin \delta_s - f_r \sin \delta_r - v = 0 \\ (1 - \cos \delta_s) f_s - \cos \delta_r f_r = 0 \\ f_s = a \cos(\theta + \delta_s) - v \sin \delta_s \\ f_r = v \sin \delta_r \end{array} \}.$$



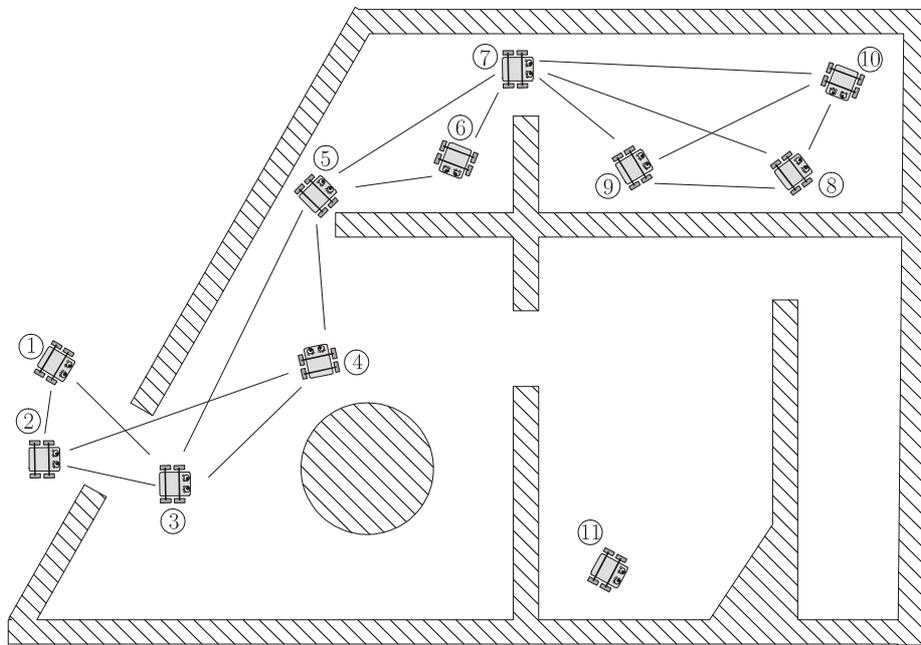


3.3 Control



4 CAROTTE





5 CSPs

Un CSP est constitué.

- d'un ensemble de variables $\mathcal{V} = \{x_1, \dots, x_n\}$,
- d'un ensemble de contraintes $\mathcal{C} = \{c_1, \dots, c_m\}$
- d'un ensemble d'intervalles $\{[x_1], \dots, [x_n]\}$.

- 1) Les informations sur la localisation d'un robot et sur sa carte peuvent être représentées par un CSP.
- 2) Les contraintes c_i représentent les relations entre les variables. On leur associe un contracteur.
- 3) Les CSP se distribuent facilement entre différents robots.
- 4) La méthode ne linéarise pas.
- 5) Elle permet de prendre en compte des variables discrètes (entières, booléennes, ...).
- 6) Elle est robuste par rapport aux outliers.
- 7) Elle se parallélise aisément.