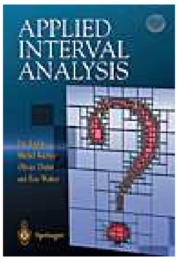
Nonlinear control using interval constraints propagation

2nd Franco Japanese Workshop on Constraint Programming

November 14th-16th, 2005, Le Croisic, France



Luc Jaulin, Laboratoire E^3I^2

ENSIETA, 2 rue François Verny, 29806 Brest Cédex 09

Objective of our team: Promote interval methods and constraint propagation within the robotics community (build solvers, solve applications, ...)

1 Interval constraints propagation

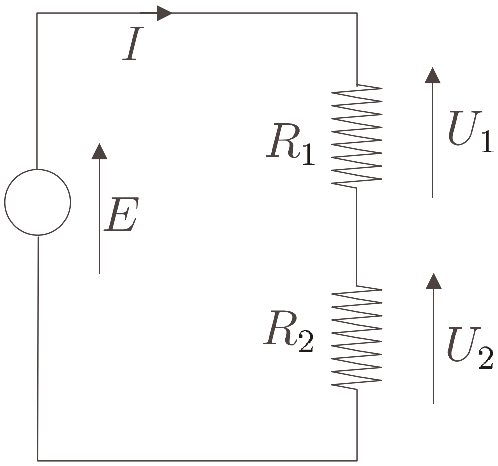
A CSP is composed of

- a set of variables $\mathcal{V} = \{x_1, \ldots, x_n\}$ of $\mathbb{R}$.
- a set of constraints $\mathcal{C} = \{c_1, \ldots, c_m\}$,
- a set of interval domains $\{[x_1], \ldots, [x_n]\}$.

Interval constraints propagation techniques make it possible to

- contract the domains $[x_i]$ without loosing solutions,
- prove that the CSP is inconsistent or infallible,
- prove that there exists a unique solution,
- prove that the solution set is connected.

1.1 Example : estimation



$$E \in [23V, 26V],\ I \in [4A, 8A],\ U_1 \in [10V, 11V],$$
$$U_2 \in [14V, 17V],\ P \in [124W, 130W],$$

$$P = EI;\ E = (R_1 + R_2)\, I;\ U_1 = R_1 I;$$
$$U_2 = R_2 I;\ E = U_1 + U_2.$$

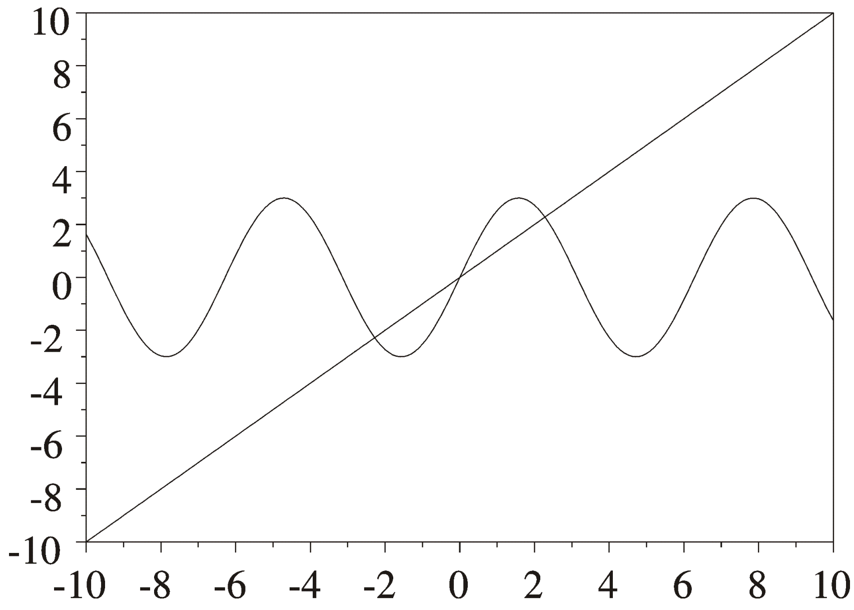
IntervalPeeler gets

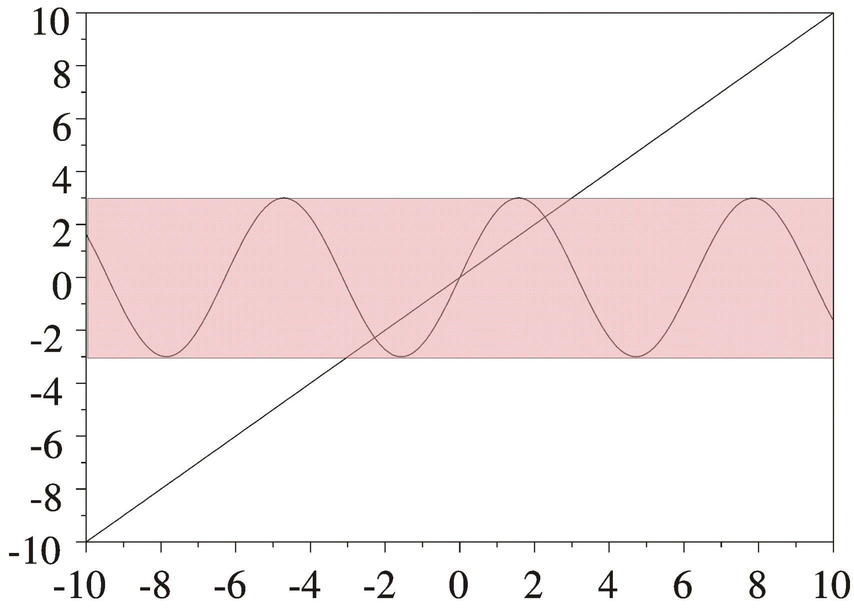
$$\begin{aligned} R_1 &\in [1.84\Omega, 2.31\Omega]\,,\ R_2 \in [2.58\Omega, 3.35\Omega], \\ I &\in [4.769A, 5.417A]\,,\ U_1 \in [10V; 11V]\,, \\ U_2 &\in [14V; 16V]\,,\ E \in [24V; 26V]\,, \\ P &\in [124W, 130W]\,. \end{aligned}$$

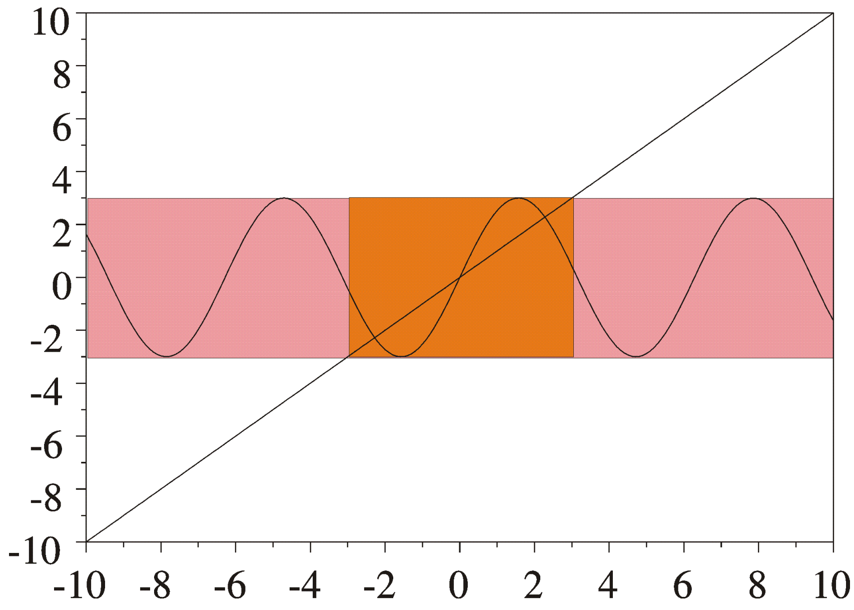
1.2 Solving nonlinear equations

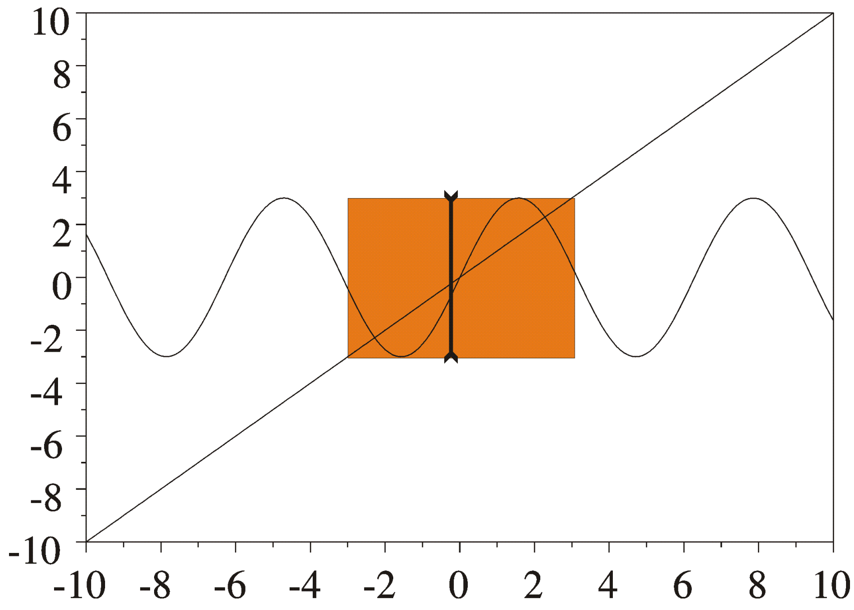
Consider the system

$$
\begin{aligned}
y &= 3\sin(x) \\
y &= x
\end{aligned}
$$

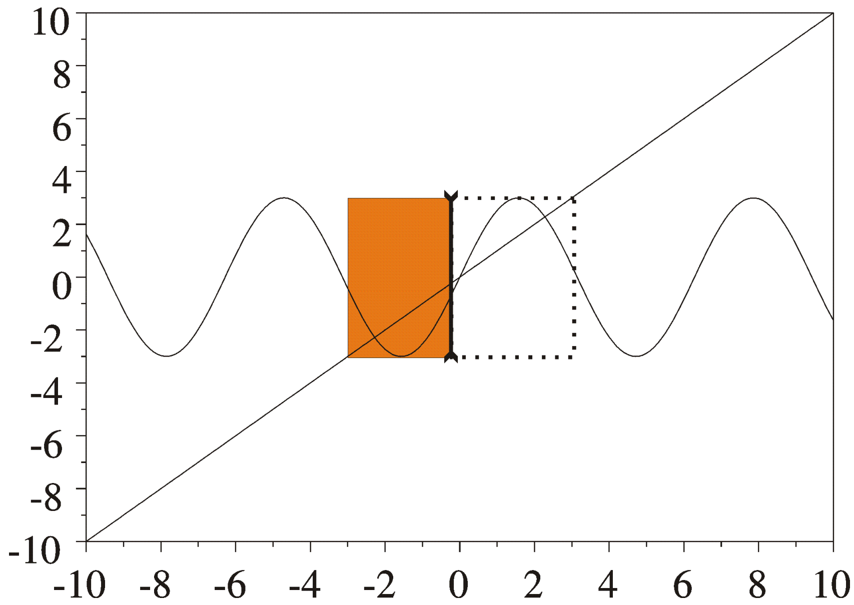

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10

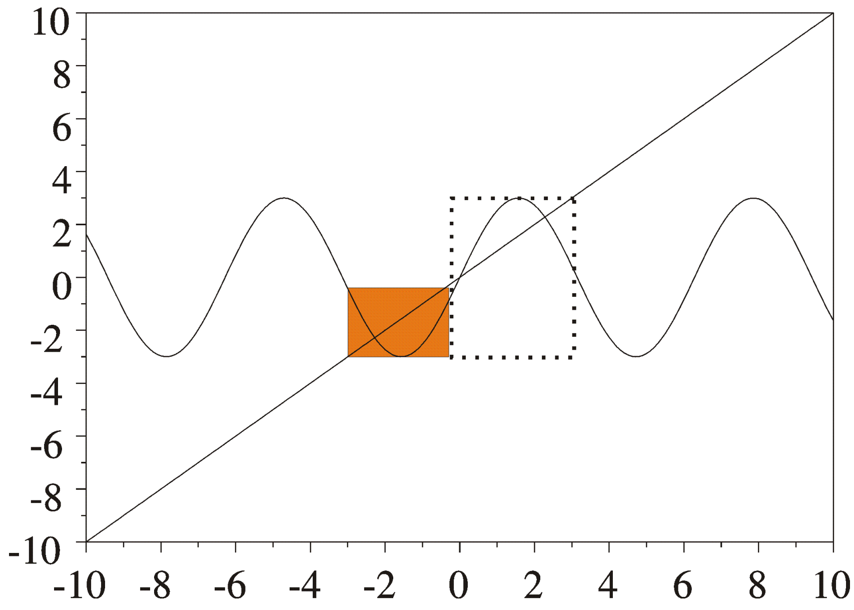

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10


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

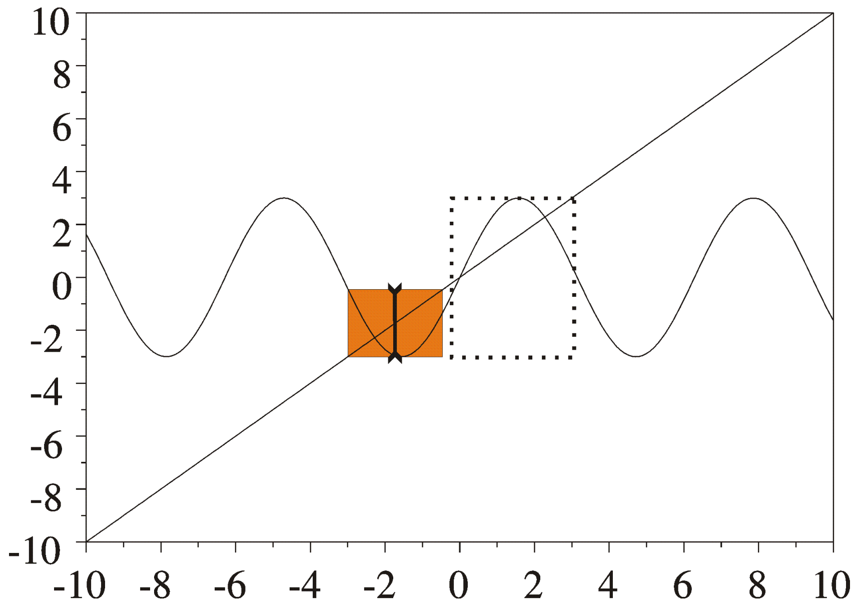


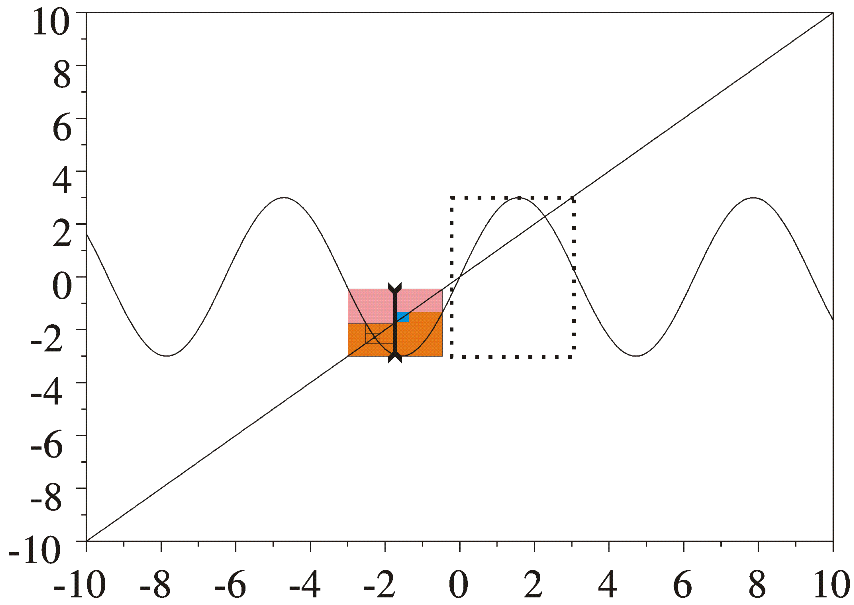
10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10


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

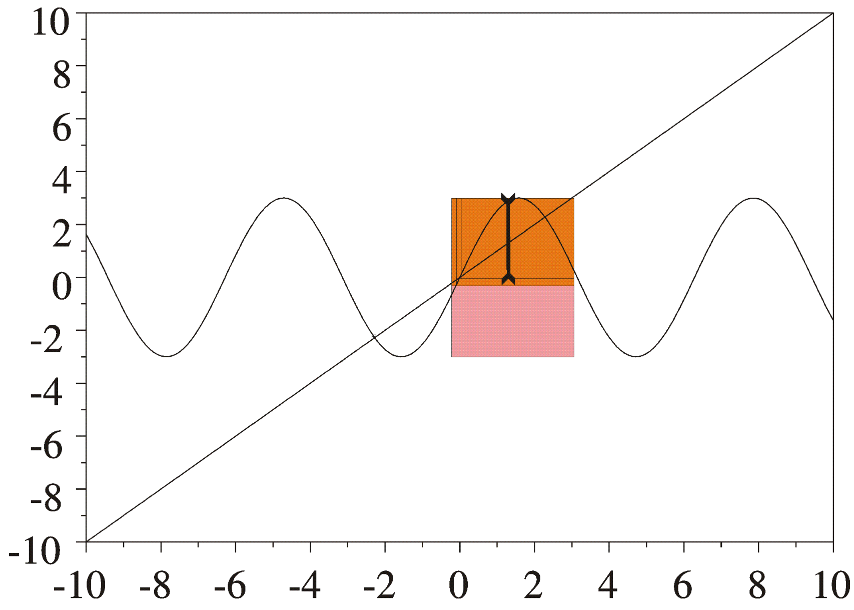


10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10

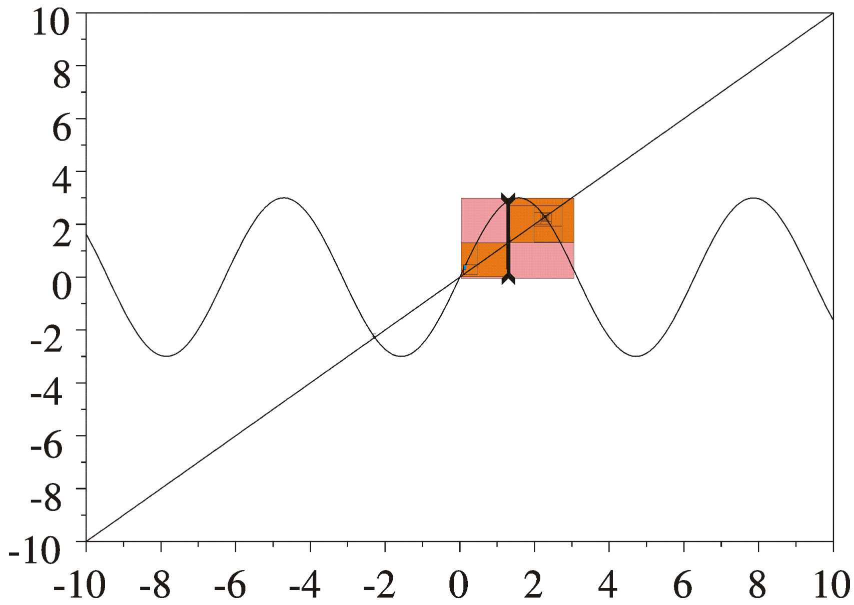

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10



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



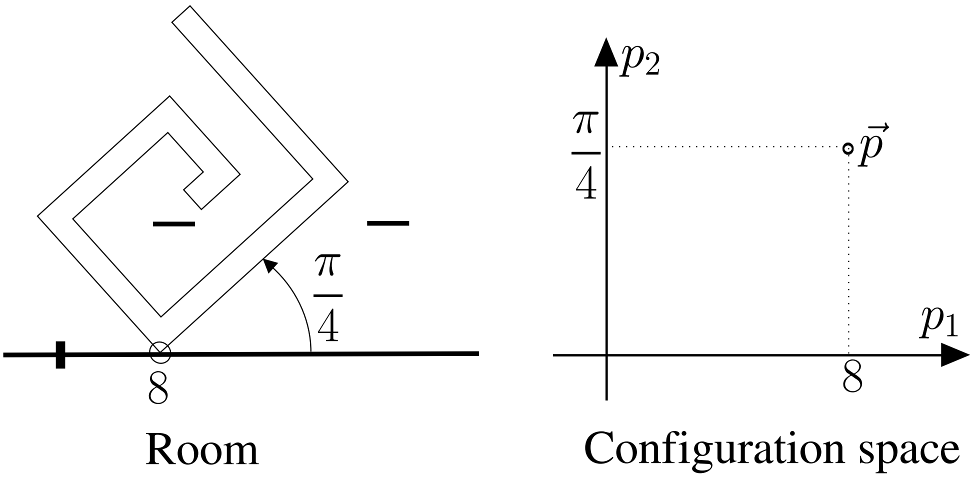
10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10

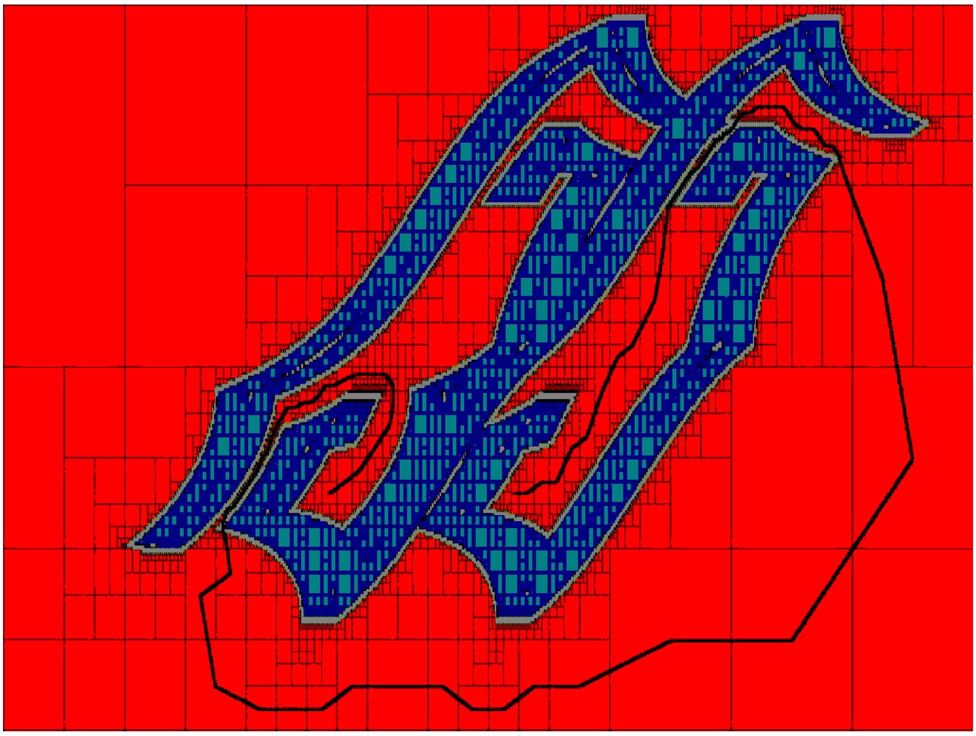


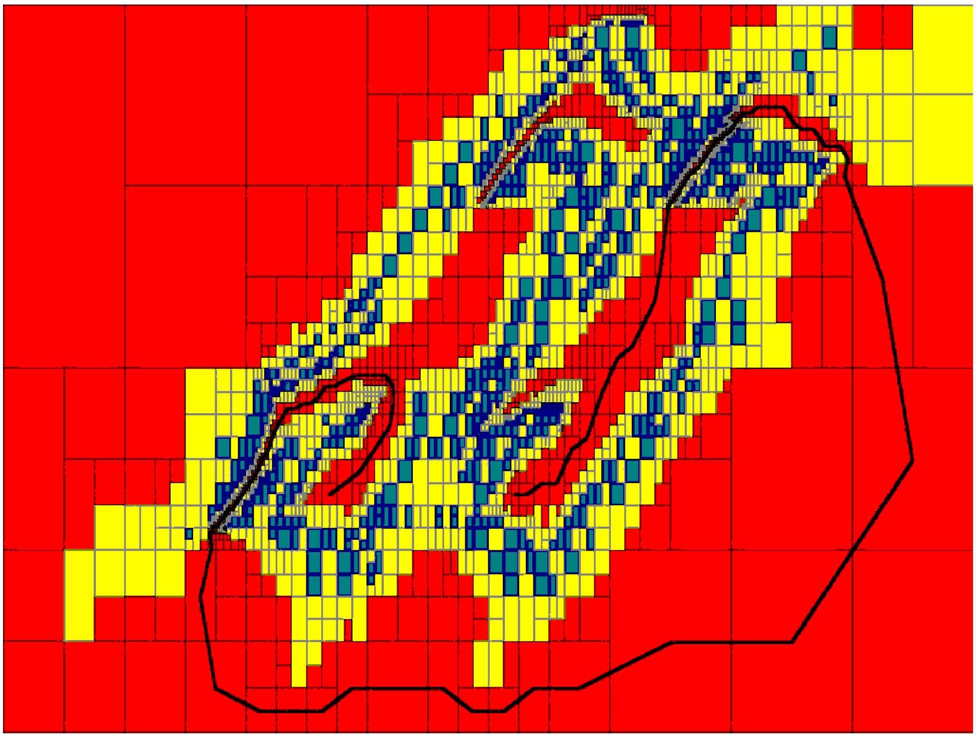
10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10

2 Application to path planning

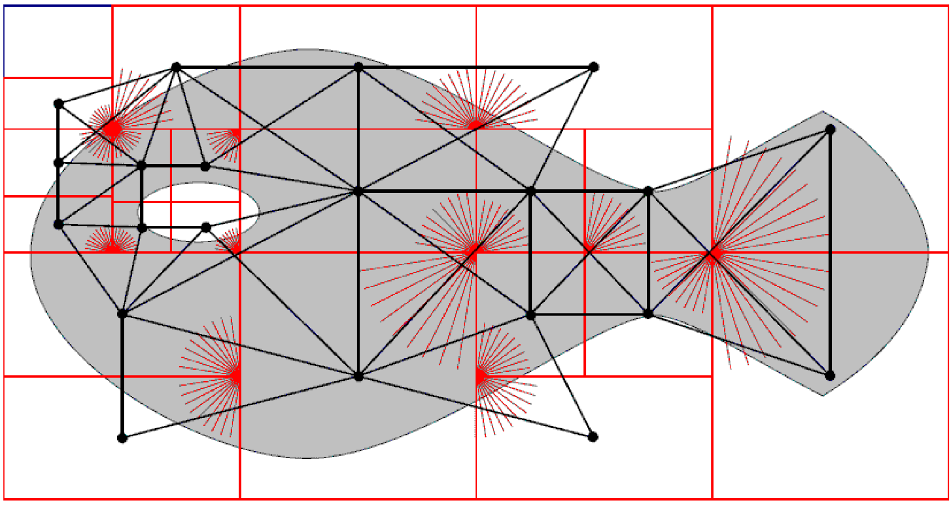
(collaboration with N. Denanoue and B. Cottenceau)

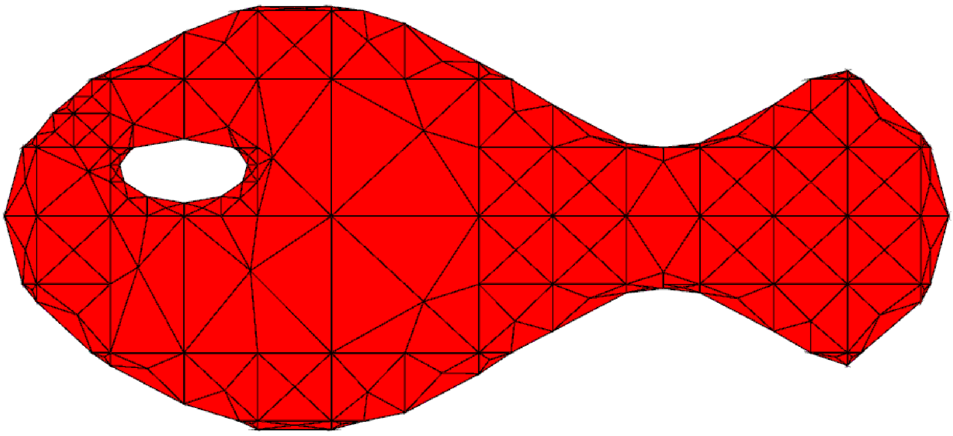






$$\mathbb{S} \stackrel{\text{def}}{=} \left\{ (x, y) \in \mathbb{R}^2 \mid \begin{pmatrix} x^2 + 4y^2 - 16 \\ 2\sin x - \cos y + y^2 - \frac{3}{2} \\ -(x + \frac{5}{2})^2 - 4(y - \frac{2}{5})^2 + \frac{3}{10} \end{pmatrix} \leq 0 \right\},$$





3 Control of a sailboat

(Collaboration with M. Dao, M. Lhommeau, P. Herrero, J. Vehi and M. Sainz).

Projection of an equality.

Consider the set

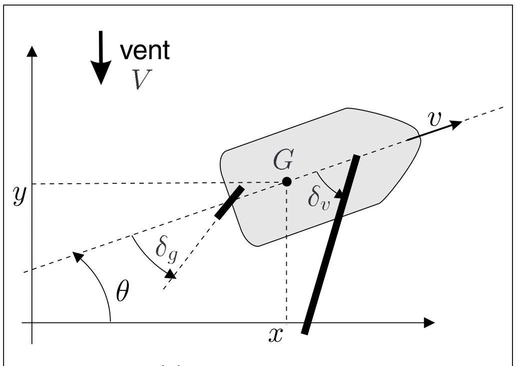
$$\mathbb{S} \overset{\text{def}}{=} \{\mathbf{p} \in \mathbf{P} \mid \exists \mathbf{q} \in \mathbf{Q}, f(\mathbf{p}, \mathbf{q}) = 0\},$$

where $\mathbf{P}$ and $\mathbf{Q}$ are boxes and f is continuous.

Since $\mathbf{Q}$ is a connected set and f is continuous, we have

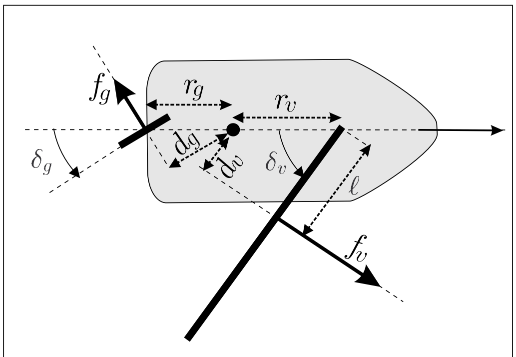
$$\mathbb{S} = \left\{\mathbf{p} \in \mathbf{P} \mid \exists\, (\mathbf{q}_1, \mathbf{q}_2) \in \mathbf{Q}^2, f(\mathbf{p}, \mathbf{q}_1) \leq 0,\ f(\mathbf{p}, \mathbf{q}_2) \geq 0\right\}.$$

Sailboat :



vent
V
v
G
y
δ_v
δ_g
θ
x

(a)



f_g
r_g
r_v
d_g
d_v
δ_g
δ_v
ℓ
f_v

(b)

$$
\begin{cases}
\dot{x} &= v \cos\theta, \\
\dot{y} &= v \sin\theta - \beta V, \\
\dot{\theta} &= \omega, \\
\dot{\delta}_s &= u_1, \\
\dot{\delta}_r &= u_2, \\
\dot{v} &= \frac{f_s \sin\delta_s - f_r \sin\delta_r - \alpha_f v}{m}, \\
\dot{\omega} &= \frac{(\ell - r_s \cos\delta_s) f_s - r_r \cos\delta_r f_r - \alpha_\theta \omega}{J}, \\
f_s &= \alpha_s \left(V \cos(\theta + \delta_s) - v \sin\delta_s \right), \\
f_r &= \alpha_r v \sin\delta_r.
\end{cases}
$$

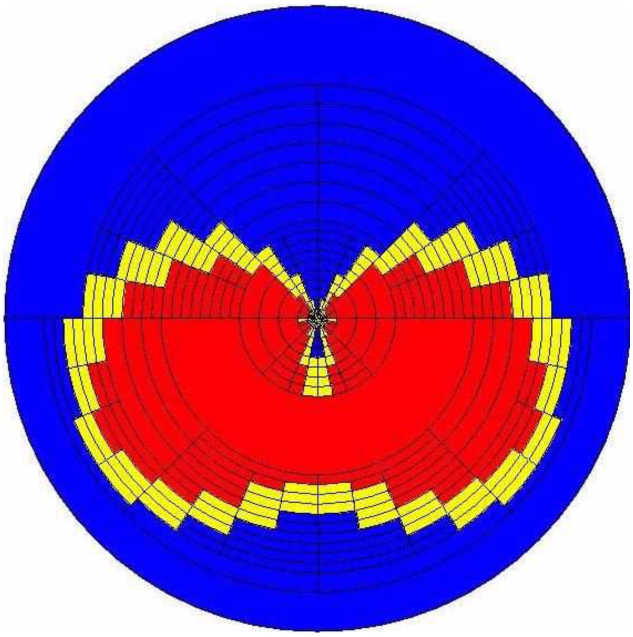
Polar speed diagram of a sailboat.

The set of feasible chosen input vectors is

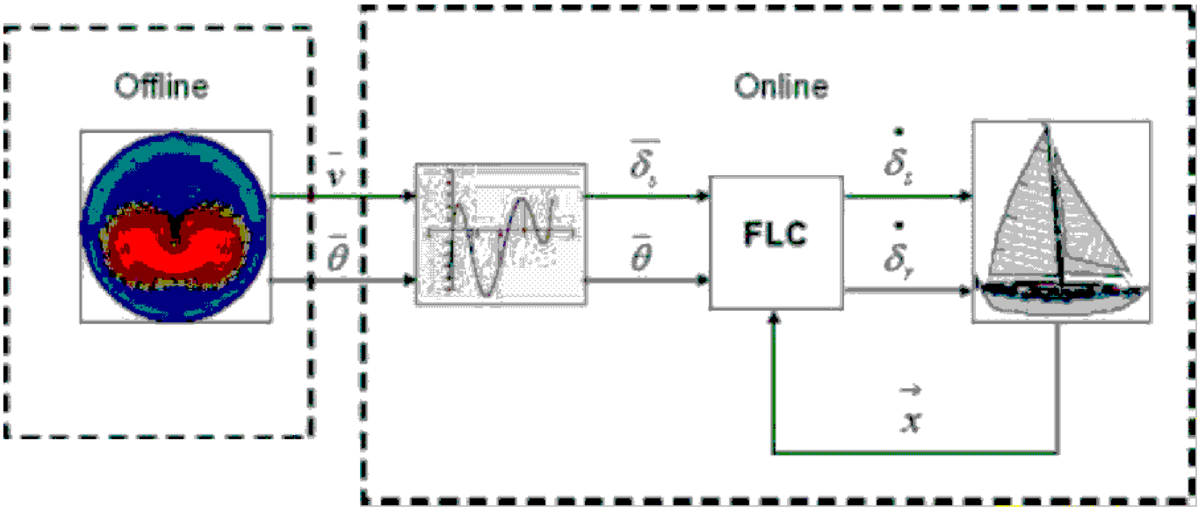
$$\mathbb{W} = \left\{ \quad (\theta, v) \mid \quad \begin{array}{l} \exists (f_s, f_r, \delta_r, \delta_s) \\ 0 = \frac{f_s \sin \delta_s - f_r \sin \delta_r - \alpha_f v}{m} \\ 0 = \frac{(\ell - r_s \cos \delta_s) f_s - r_r \cos \delta_r f_r}{J} \\ f_s = \alpha_s \left(V \cos \left(\theta + \delta_s \right) - v \sin \delta_s \right) \\ f_r = \alpha_r v \sin \delta_r \end{array} \right\}.$$

An elimination of f_s, f_r and δ_r yields

$$\mathbb{W} = \{ \quad (\theta, v) \mid \exists \delta_s \in [-\tfrac{\pi}{2}, \tfrac{\pi}{2}], \left(\frac{(\alpha_r + 2\alpha_f) v + 2\alpha_s v \sin^2 \delta_s}{V} - 2\alpha_s \cos(\theta + \delta_s) \sin \delta_s \right)^2 + \left(\frac{2\alpha_s}{r_r} (\ell - r_s \cos \delta_s) \left(\cos(\theta + \delta_s) - \frac{v}{V} \sin \delta_s \right) \right)^2 - \alpha_r^2 \frac{v^2}{V^2} = 0 \ \}$$



This picture has been obtained using modal interval techniques.



Offline
Online
FLC

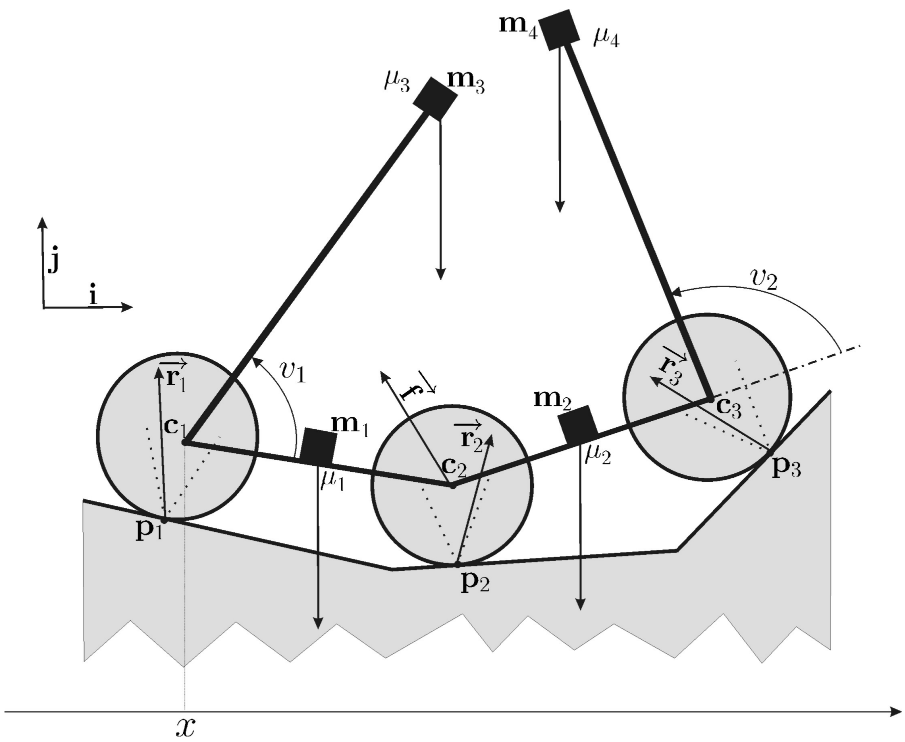
4 Control of a wheeled stair-climbing robot

(Collaboration with students and colleagues from EN-SIETA)

Consider the class of constrained dynamic systems:

$$\begin{array}{ll}\text{(i)} & \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \text{(ii)} & (\mathbf{x}(t), \mathbf{v}(t)) \in \mathbb{V},\end{array}$$

where $\mathbf{v}(t) \in \mathbb{R}^{n_v}$ is the *viable input vector* and $\mathbb{V}$ is the *viable set*.



Assume that the robot has a quasi-static motion.

1) When the robot does not move, we have

$$
\left\{
\begin{array}{rcl}
-\overrightarrow{\mathbf{p_1m_1}} \wedge \mu_1\mathbf{j} + \overrightarrow{\mathbf{p_1c_2}} \wedge \overrightarrow{\mathbf{f}} - \overrightarrow{\mathbf{p_1m_3}} \wedge \mu_3\mathbf{j} & = & 0 \\
-\overrightarrow{\mathbf{p_2m_2}} \wedge \mu_2\mathbf{j} - \overrightarrow{\mathbf{p_2c_2}} \wedge \overrightarrow{\mathbf{f}} + \overrightarrow{\mathbf{p_2p_3}} \wedge \overrightarrow{\mathbf{r}}_3 & & \\
-\overrightarrow{\mathbf{p_2m_4}} \wedge \mu_4\mathbf{j} & = & 0 \\
\overrightarrow{\mathbf{r}}_1 - (\mu_1 + \mu_3)\mathbf{j} + \overrightarrow{\mathbf{f}} & = & 0 \\
\overrightarrow{\mathbf{r}}_2 - \overrightarrow{\mathbf{f}} - (\mu_2 + \mu_4)\mathbf{j} + \overrightarrow{\mathbf{r}}_3 & = & 0,
\end{array}
\right.
$$

This system can be written into a matrix form as

$$\mathbf{A}_1(x).\mathbf{y} = \mathbf{b}_1(x),$$

where

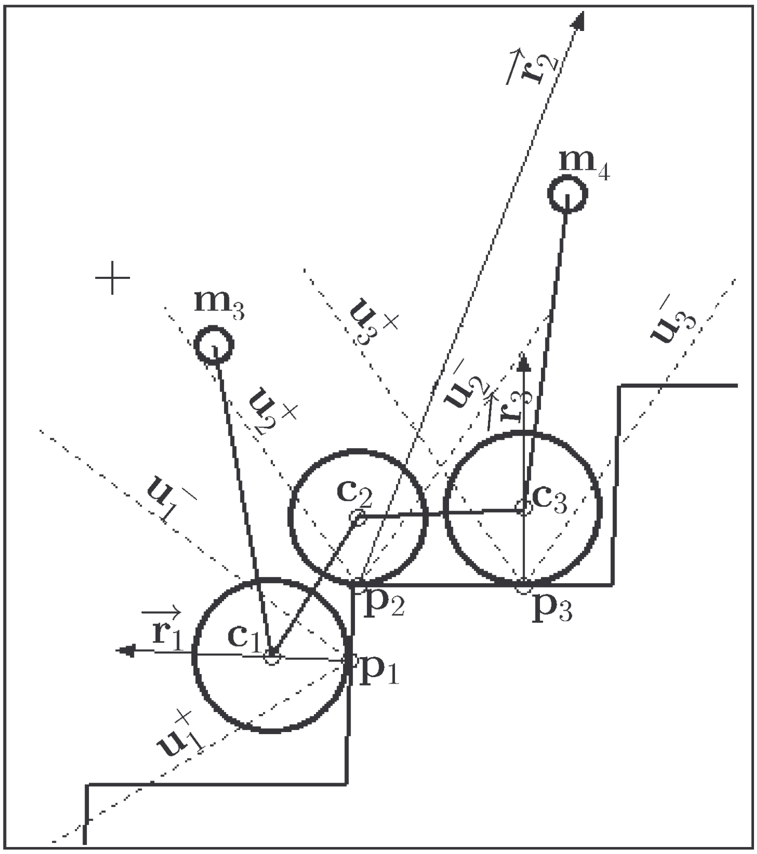
$$\mathbf{y} \;=\; \left(r_{1x}, r_{1y}, r_{2x}, r_{2y}, r_{3x}, r_{3y}, f_x, f_y, m_{3x}, m_{4x}\right)^{\mathsf{T}}.$$

2) None of the wheels will slide if all $\overrightarrow{\mathbf{r}}_i$ belong to their corresponding Coulomb cones:

$$\det(\overrightarrow{\mathbf{r}}_i, \mathbf{u}_i^-) \leq 0 \text{ and } \det(\mathbf{u}_i^+, \overrightarrow{\mathbf{r}}_i) \leq 0,$$

where $\mathbf{u}_i^-$ and $\mathbf{u}_i^+$ denote the two vectors supporting the ith Coulomb cone $\mathcal{C}_i$. These inequalities can be rewritten into

$$\mathbf{A}_2(x).\mathbf{y} \leq \mathbf{0}.$$

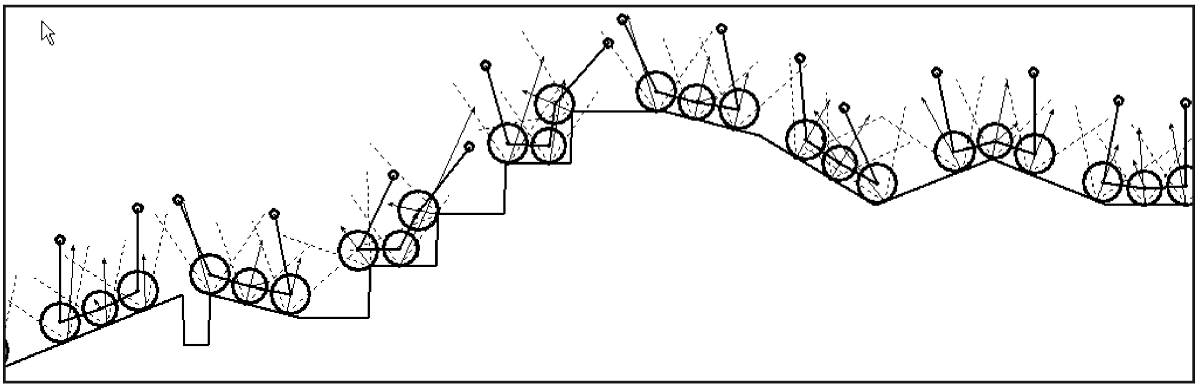


$\vec{r}_2$
m_4
m_3
u_3^+
u_3^-
u_2^-
$\vec{r}_3$
u_2^+
u_1^-
c_2
c_3
p_2
p_3
$\vec{r}_1$
c_1
p_1
u_1^+

3) There is a relation between $\mathbf{y}$ and $\mathbf{v}$ of the form $\mathbf{v} = \mathbf{c}(\mathbf{y})$.

Finally,

$$\begin{aligned} \mathbf{A}_1(x).\mathbf{y} &= \mathbf{b}_1(x) \\ \mathbf{A}_2(x).\mathbf{y} &\leq \mathbf{0}. \\ \mathbf{v} &= \mathbf{c}(\mathbf{y}) \end{aligned}$$



The figure below represents the robot built by the robotics team of the ENSIETA engineering school that has won the 2005 robot cup ETAS. The robot can be seen as a three-dimensional version of the robot treated above. It has been proven to be very competitive on irregular grounds but failed to cross over some compulsory obstacles (such as stairs).

5 Localization of an AUV

Collaboration with the GESMA (Groupe d'Etude Sous-Marine de l'Atlantique). The sensors are composed by a DGPS when the AUV is at the surface of the ocean, an accelerometer and a camera oriented toward the bottom

The state space equations are given by

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = u_1 \\ \dot{v} = u_2 \end{cases}$$

An estimation of the state vector can be obtained by integration :

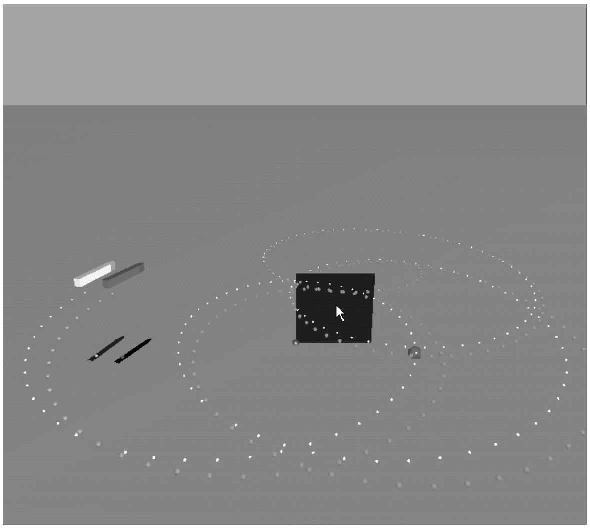
$$\frac{d}{dt}\begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{\theta} \end{pmatrix} = \begin{pmatrix} v\cos\hat{\theta} \\ v\sin\hat{\theta} \\ u_1 \end{pmatrix}.$$

This state estimation is used by our controller.

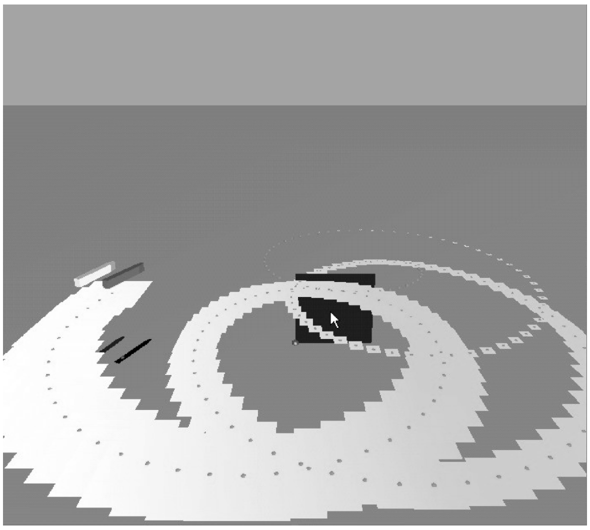
The controller to be used is given by

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} -\frac{\sin\hat{\theta}}{v} & \frac{\cos\hat{\theta}}{v} \\ \cos\hat{\theta} & \sin\hat{\theta} \end{pmatrix} \begin{pmatrix} \frac{x_d-\hat{x}}{4} + \dot{x}_d - v\cos\hat{\theta} + \ddot{x}_d \\ \frac{y_d-\hat{y}}{4} + \dot{y}_d - v\sin\hat{\theta} + \ddot{y}_d \end{pmatrix}.$$

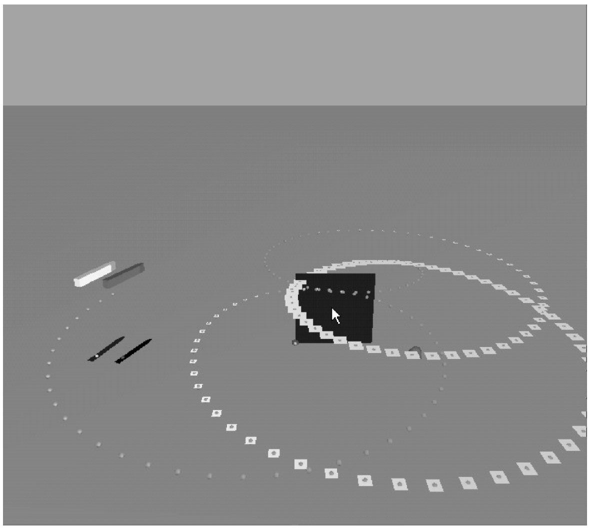
The figure below represents the actual trajectory in the situation where the desired trajectory is a cycloid. The white AUV represents the location where the AUV thinks it is located. The grey AUV represents the actual location.



An envelope containing the actual trajectory can be obtained using an interval simulation.



In the case where the output location is considered for the propagation, we get the following envelope.



If the camera is taken into account, we get :