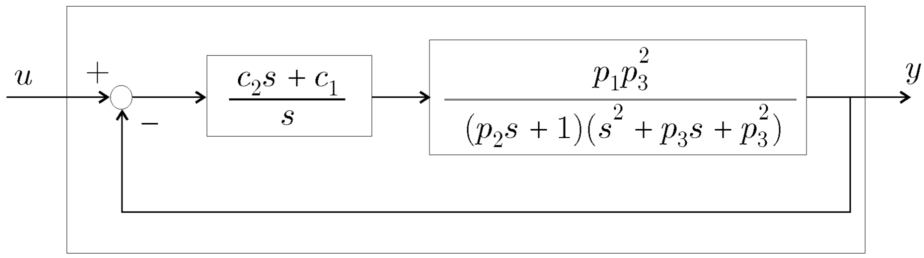
Embedded intervals

L. Jaulin
ENSIETA, Brest

Groupe de travail *calcul ensembliste*
du GDR Macs
Jeudi 19 juillet 2007

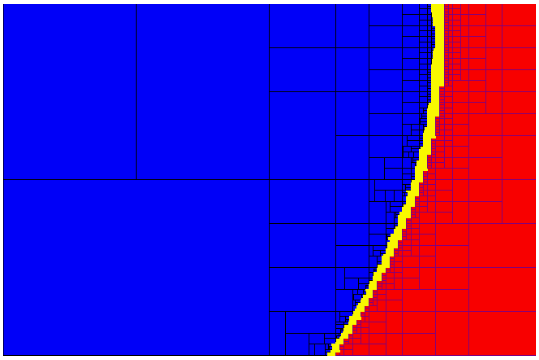
1 Intervals for robotics

1.1 Robust control



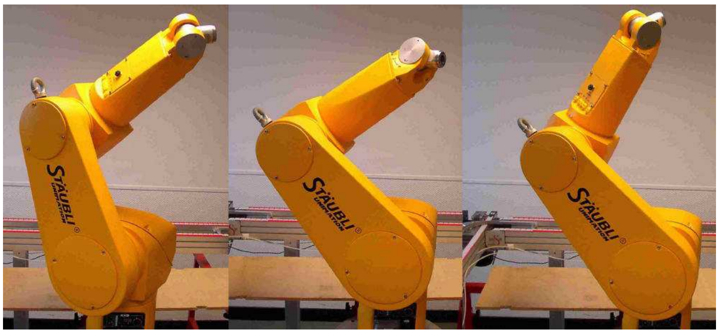
Finding one robust controller amounts to find one point inside the set

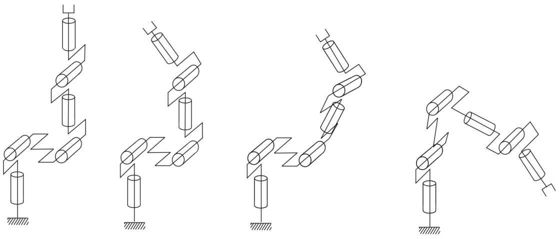
$$\mathbb{T}_\mathsf{c} = \{\mathbf{c} \in [\mathbf{c}] \mid \forall \mathbf{p} \in [\mathbf{p}], r(\mathbf{c}, \mathbf{p}) > 0\}$$



Approximation of the set $\neg\mathbb{T}_c$

1.2 Calibration

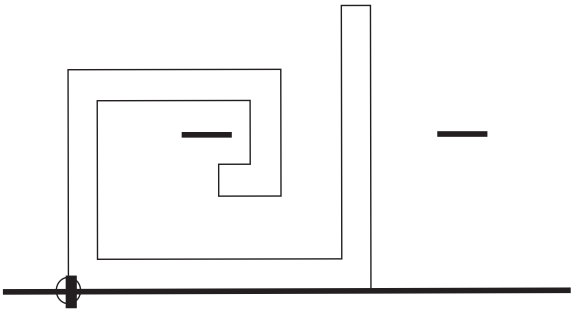




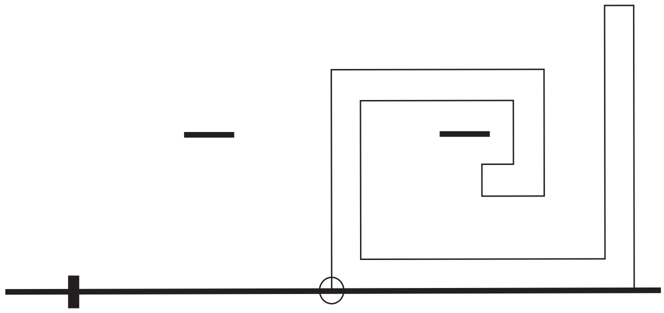
The calibration of the robot Staubli RX90 amounts to solve a parametric estimation problem where the parameter vector is

$$\begin{aligned}\mathbf{p} \;=\; & (r_0, \alpha_1, d_1, r_1, \dots, \alpha_5, d_5, r_5, \alpha_6, d_6,\\ & \theta_0, \theta_1^o, \dots, \theta_5^o,\\ & b_x^1, b_y^1, b_z^1, b_x^2, b_y^2, b_z^2, b_x^3, b_y^3, b_z^3).\end{aligned}$$

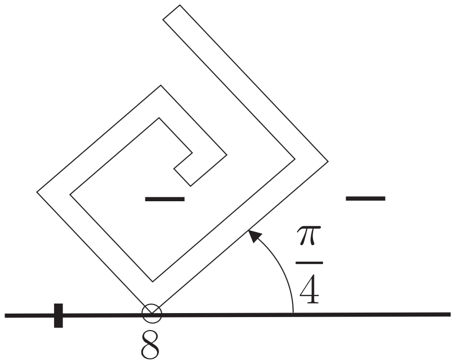
1.3 Path planning



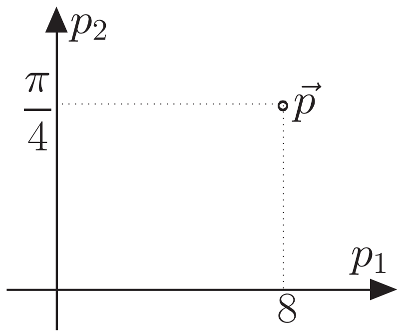
Initial configuration: $\vec{p} = (0\ 0)^{\mathrm{T}}$



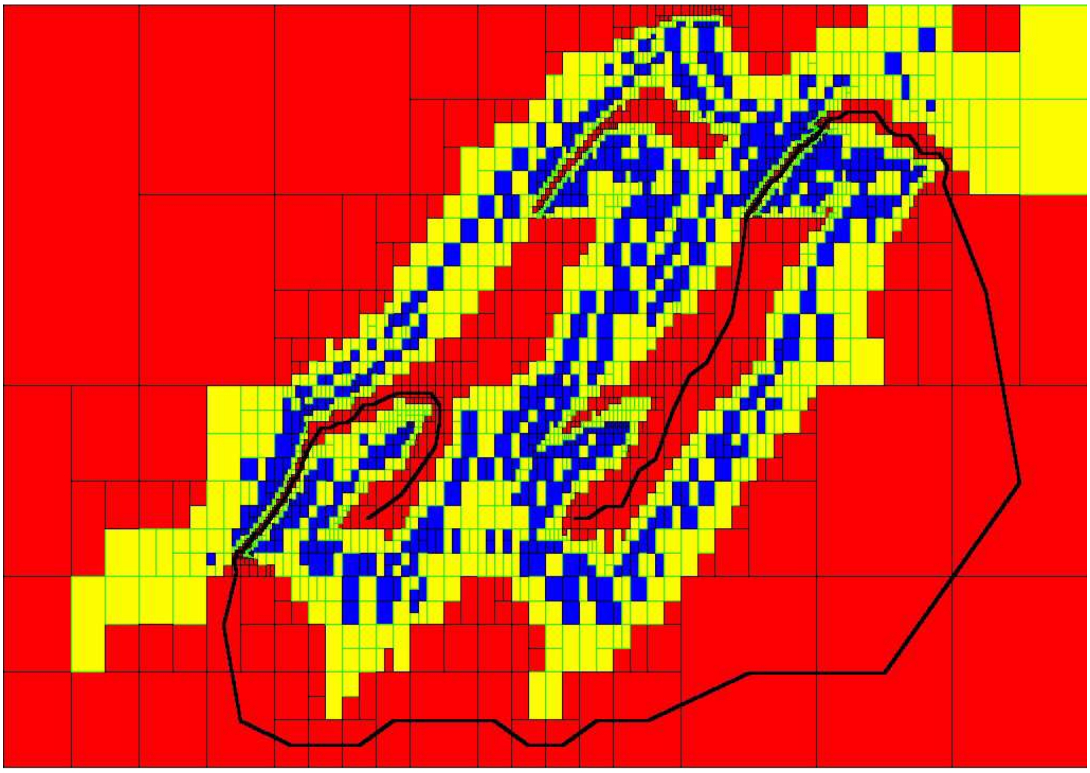
Goal configuration: $\vec{p} = (17\ 0)^{\mathrm{T}}$

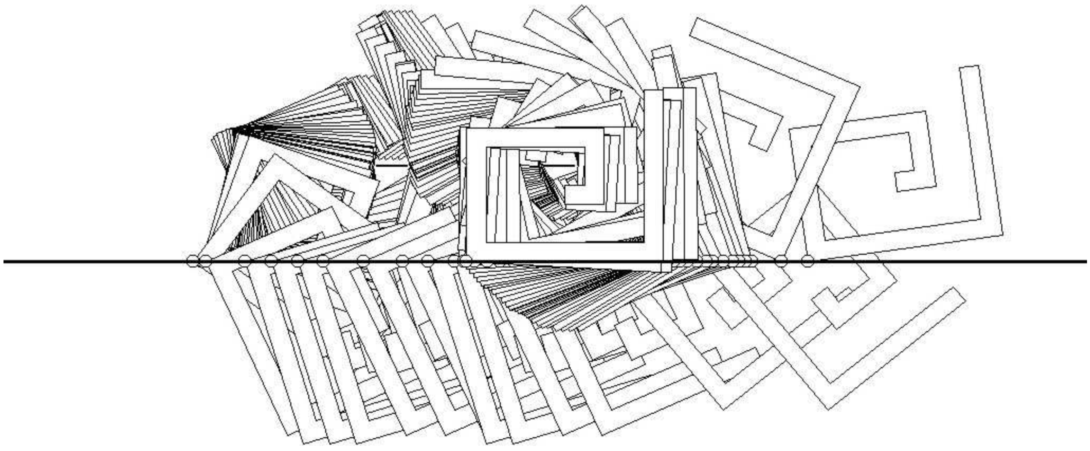


Room



Configuration space





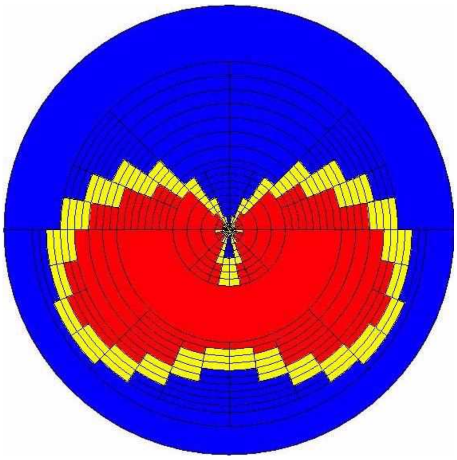
1.4 Nonlinear control

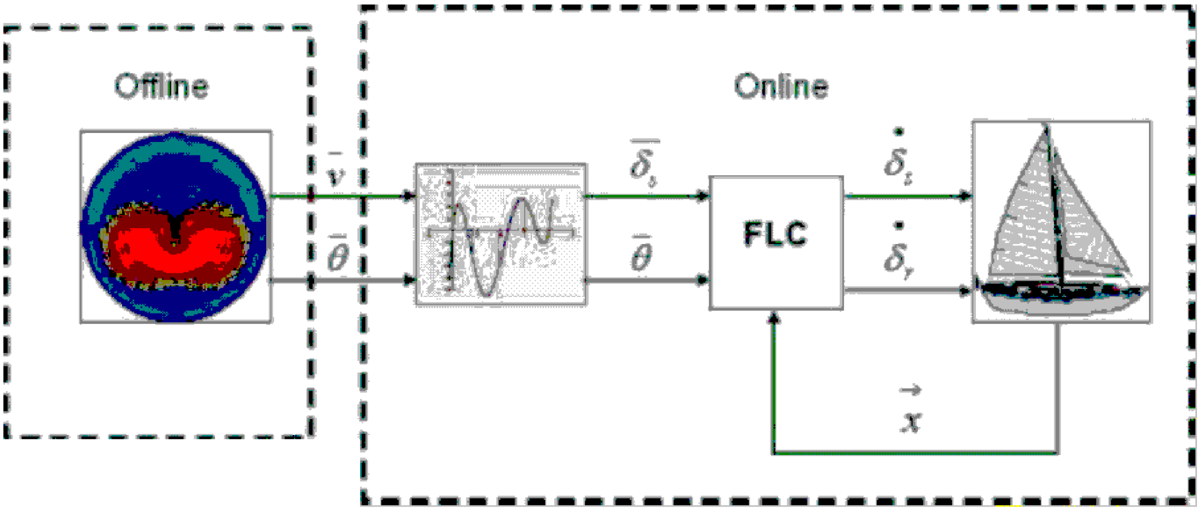
State equations of a sailboat

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

Polar speed diagram

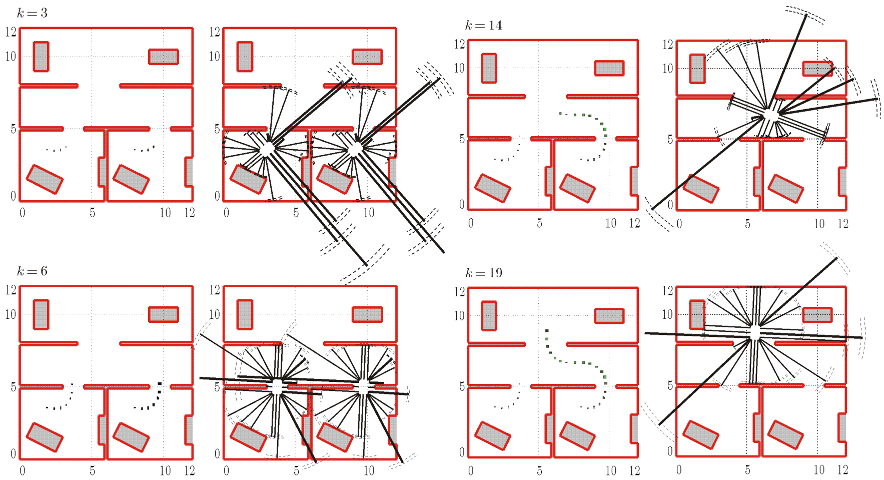
$$\mathbb{W} = \{ \quad (\theta, v) \mid \quad \begin{array}{l} \exists(\omega, u_1, u_2, f_s, f_r, \delta_r, \delta_s) \\ \omega = 0, u_1 = 0, u_2 = 0 \\ \frac{f_s \sin \delta_s - f_r \sin \delta_r - \alpha_f v}{m} = 0 \\ \frac{(\ell - r_s \cos \delta_s) f_s - r_r \cos \delta_r f_r}{J} = 0 \\ f_s = \alpha_s \left(V \cos(\theta + \delta_s) - v \sin \delta_s \right) \\ f_r = \alpha_r v \sin \delta_r \end{array} \quad \}.$$



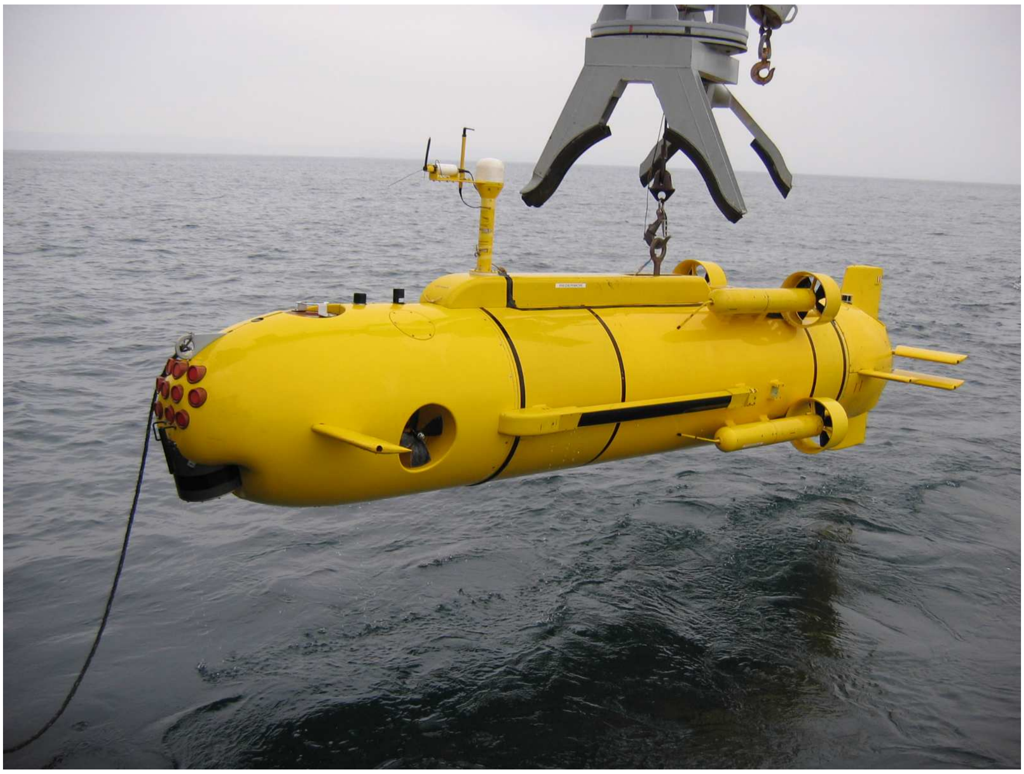


Offline
Online
$\bar{v}$
$\bar{\theta}$
$\bar{\delta}_s$
$\bar{\theta}$
FLC
$\dot{\delta}_s$
$\dot{\delta}_r$
$\vec{x}$

1.5 State estimation

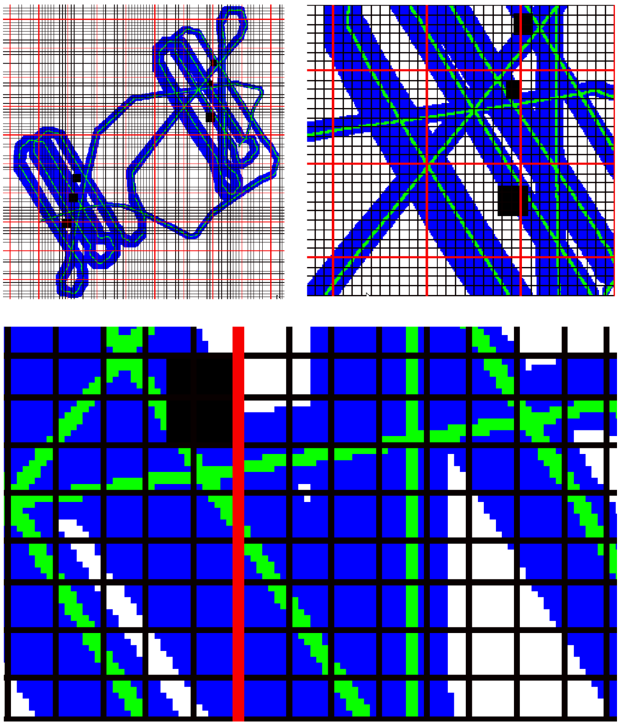


1.6 SLAM



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


A785



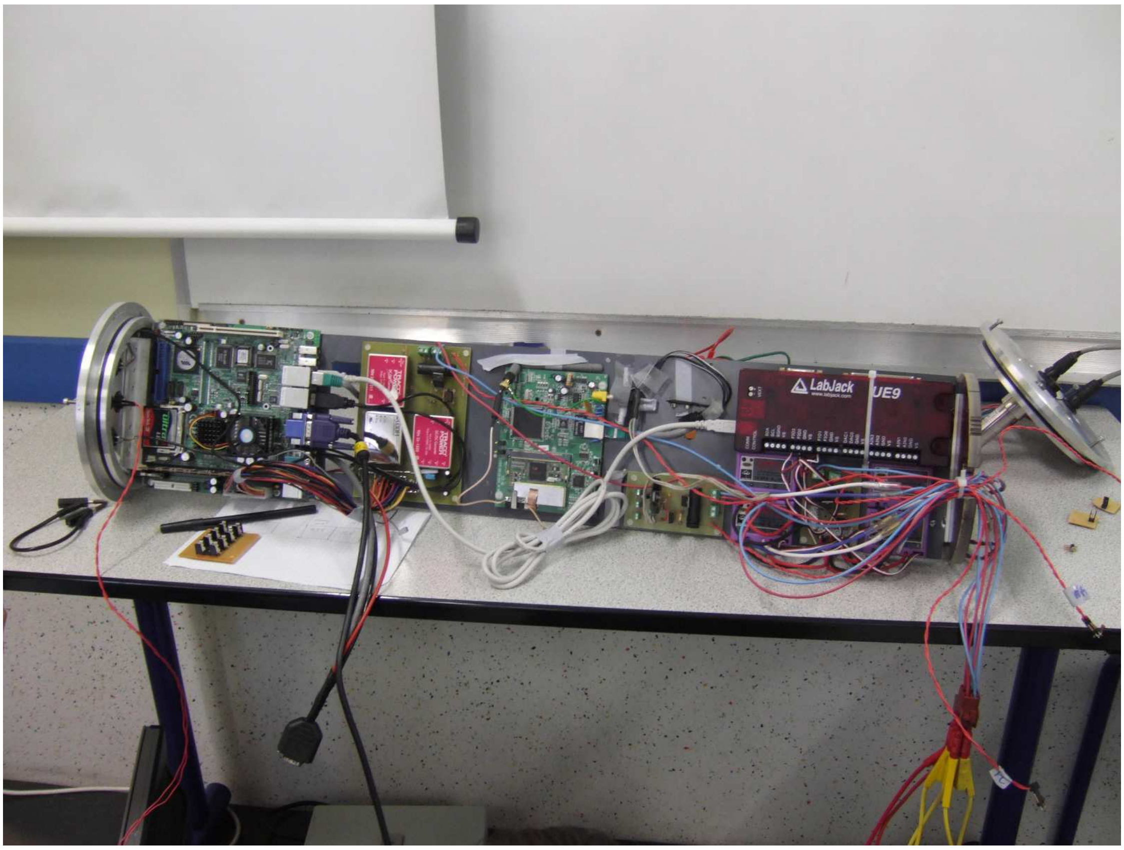
Trajectory reconstructed by GESMI

2 SAUC'E competition

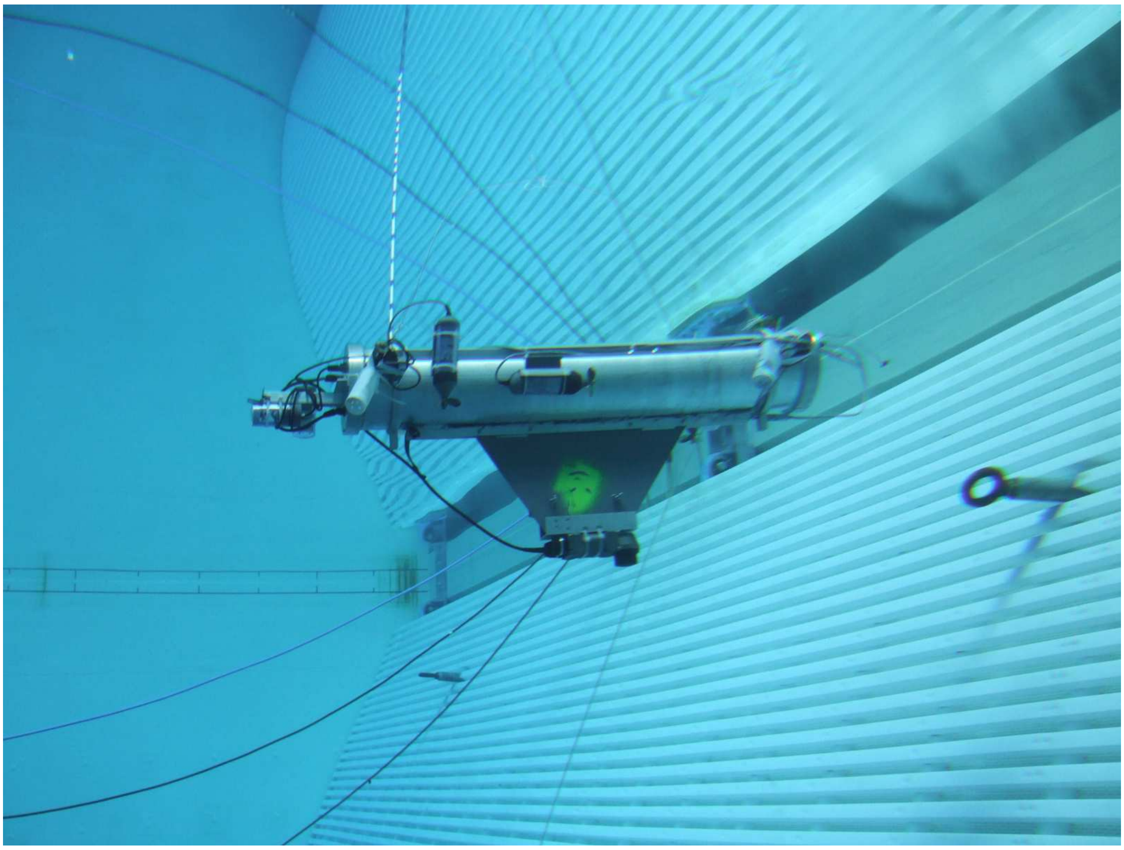
In Portsmouth, July 12-15, 2007.

3 SAUC'ISSE

(show the movie)

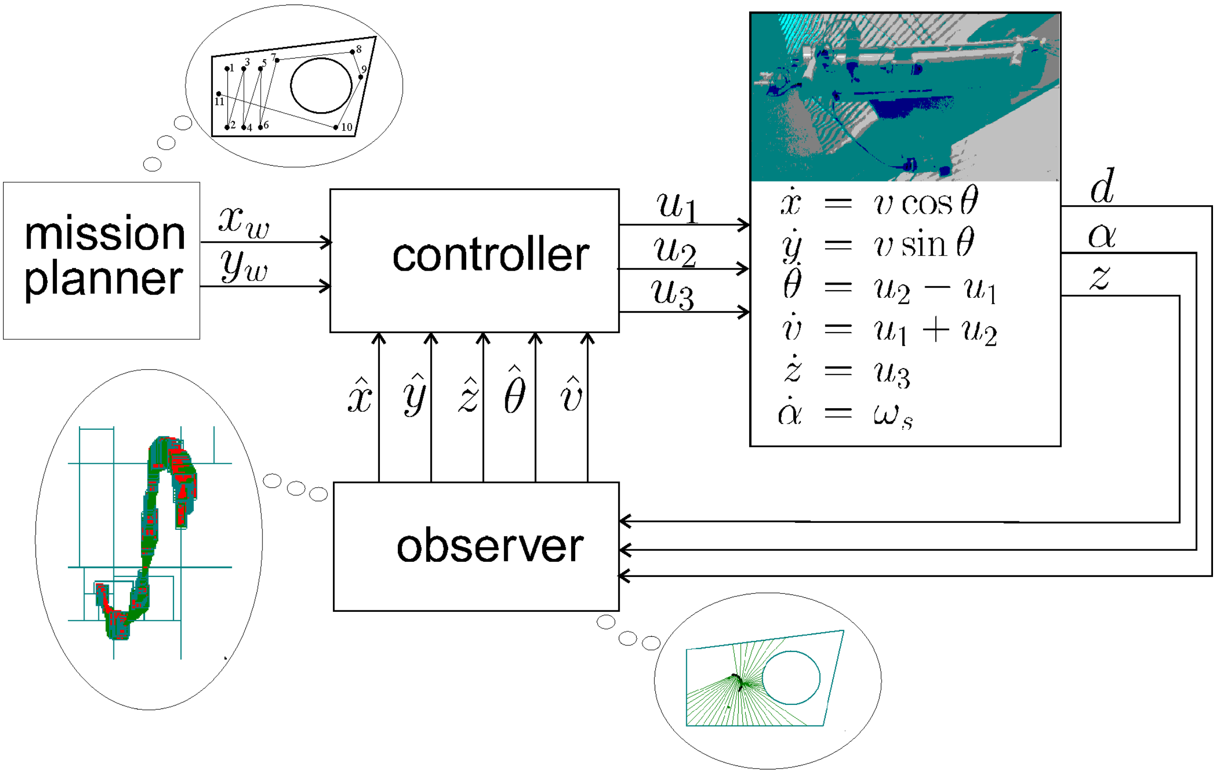

LabJack
UE9


AR HAUT



4 Localization and control

4.1 Principle



4.2 Some intervals

If

$$\mathbb{S}=\{x\in\mathbb{R},\ \gamma((x\in[a(1)])\,,\dots,(x\in[a(m)]))\}\,,$$

where γ is any Boolean function. An optimal contractor for $\mathbb{S}$ is .

Algorithm Relax(in: $\gamma,[x],[a(1)],\dots,[a(m)]$, out: $[x]$)	
1	$\mathcal{X}=\left\{x^-,x^+,a^-(1),a^+(1),\dots,a^-(m),a^+(m)\right\}$
2	$\mathcal{V}=\{v\in\mathcal{X}\cap[x],\ \gamma((v\in[a(1)])\,,\dots,(v\in[a(m)]))\}$
3	Return the smallest interval which encloses $\mathcal{V}$.

Vector case. If

$$\mathbb{S} = \{\mathbf{x} \in \mathbb{R}^n, \, \gamma((\mathbf{x} \in [\mathbf{a}(1)]) \, , \dots , (\mathbf{x} \in [\mathbf{a}(m)]))\} \, ,$$

where γ is an increasing Boolean function, then,

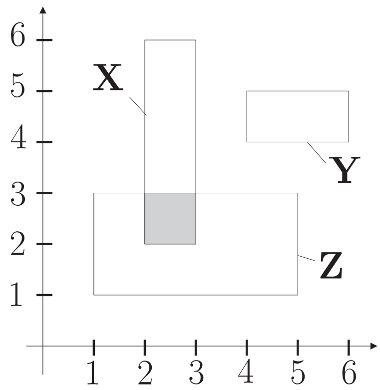
$$\begin{array}{rl} & \gamma((\mathbf{x} \in [\mathbf{a}(1)]) \, , \dots , (\mathbf{x} \in [\mathbf{a}(m)])) \\ \Rightarrow & \forall i \in \{1, \dots, n\}, \gamma((x_i \in [a_i(1)]) \, , \dots , (x_i \in [a_i(m)])) \end{array}$$

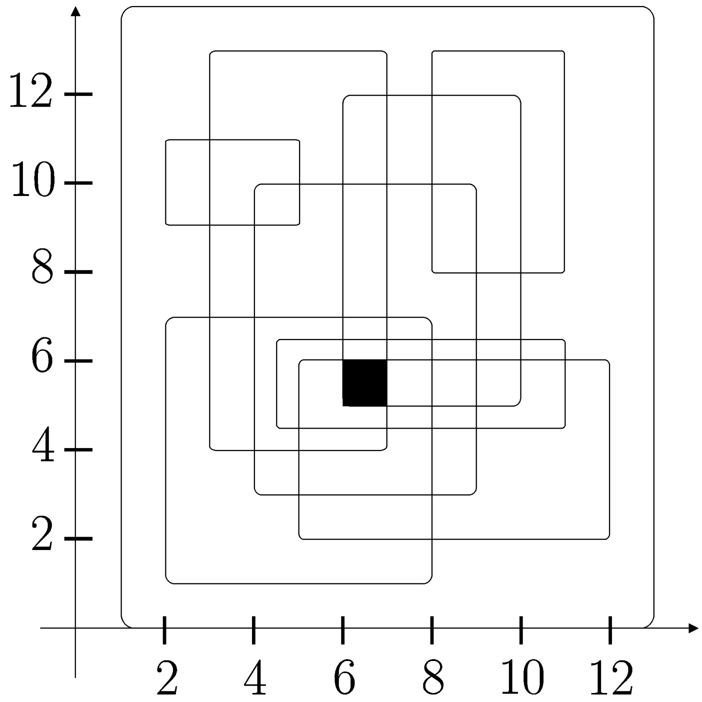
Thus, a contractor for $\mathbb{S}$ is given by

Algorithm Relax(in: $\gamma, [\mathbf{x}], [\mathbf{a}(1)], \dots, [\mathbf{a}(m)]$, out: $[\mathbf{x}]$)	
1	for $i = 1$ to n
2	Relax($\gamma, [x_i], [a_i(1)], \dots, [a_i(m)]$)

Example: If

$$
\begin{aligned}
& \gamma((\mathbf{x} \in \mathbf{X}), (\mathbf{x} \in \mathbf{Y}), (\mathbf{x} \in \mathbf{Z})) \\
\Leftrightarrow\ & \mathbf{x} \text{ belongs to at least 2 of the 3 boxes } \mathbf{X}, \mathbf{Y}, \mathbf{Z}. \\
\Leftrightarrow\ & (\mathbf{x} \in \mathbf{X} \ \wedge \mathbf{x} \in \mathbf{Y}) \vee (\mathbf{x} \in \mathbf{X} \ \wedge \mathbf{x} \in \mathbf{Z}) \\
& \vee (\mathbf{x} \in \mathbf{Y} \ \wedge \mathbf{x} \in \mathbf{Z})
\end{aligned}
$$





The black box is the 2-intersection of 9 boxes

4.3 Observer

Consider the discrete time dynamic system

$$\begin{array}{rcl} \mathbf{x}(k+1) & = & \mathbf{f}_k(\mathbf{x}(k)) \\ \mathbf{y}(k) & = & \mathbf{g}(\mathbf{x}(k)). \end{array}$$

In a bounded-error context, we generally assume that

$$\mathbf{y}(k) \in \mathbb{Y}(k).$$

To robustify against outliers, we make the following assumption:

Outliers may exist for the outputs but within any time window of length ℓ we never have more than q outliers.

We can thus define recursively the feasible set $\mathbb{X}(k)$ for $\mathbf{x}(k)$:

(i) $\mathbb{X}(k) = \mathbb{R}^n$, if $k \leq 0$.

(ii) and $\mathbb{X}(k+1)$ is the set of all $\mathbf{x}(k+1) \in \mathbb{R}^n$ such that

$$\begin{cases} \exists \mathbf{x}(k-\ell) \in \mathbb{X}(k-\ell), \ldots, \exists \mathbf{x}(k) \in \mathbb{X}(k), \\ \exists \mathbf{y}(k-\ell), \ldots, \exists \mathbf{y}(k), \\ \mathbf{x}(k-\ell+1) = \mathbf{f}_{k-\ell}(\mathbf{x}(k-\ell)), \ldots, \mathbf{x}(k+1) = \mathbf{f}_k(\mathbf{x}(k)) \\ \mathbf{y}(k-\ell) = \mathbf{g}(\mathbf{x}(k-\ell)), \ldots, \mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k)) \\ \bigwedge_q ((\mathbf{y}(k-\ell) \in \mathbb{Y}(k-\ell)) \ldots, (\mathbf{y}(k) \in \mathbb{Y}(k))) \,. \end{cases}$$

Assume that $\mathbf{f}_k$ is invertible, *i.e.*,

$$\mathbf{x}(k+1) = \mathbf{f}_k(\mathbf{x}(k)) \Leftrightarrow \mathbf{x}(k) = \overline{\mathbf{f}}_k(\mathbf{x}(k+1)).$$

Define the functions

$$\overline{\mathbf{f}}_k^{\ell}(\mathbf{x}(k+1)) = \overline{\mathbf{f}}_{k-\ell} \circ \cdots \circ \overline{\mathbf{f}}_{k-1} \circ \overline{\mathbf{f}}_k\left(\mathbf{x}(k+1)\right).$$

Then, we have

$$\begin{pmatrix} \mathbf{x}(k-\ell) \\ \vdots \\ \mathbf{x}(k) \end{pmatrix} = \begin{pmatrix} \overline{\mathbf{f}}_k^{\ell+1}(\mathbf{x}(k+1)) \\ \vdots \\ \overline{\mathbf{f}}_k^{1}(\mathbf{x}(k+1)) \end{pmatrix}.$$

and

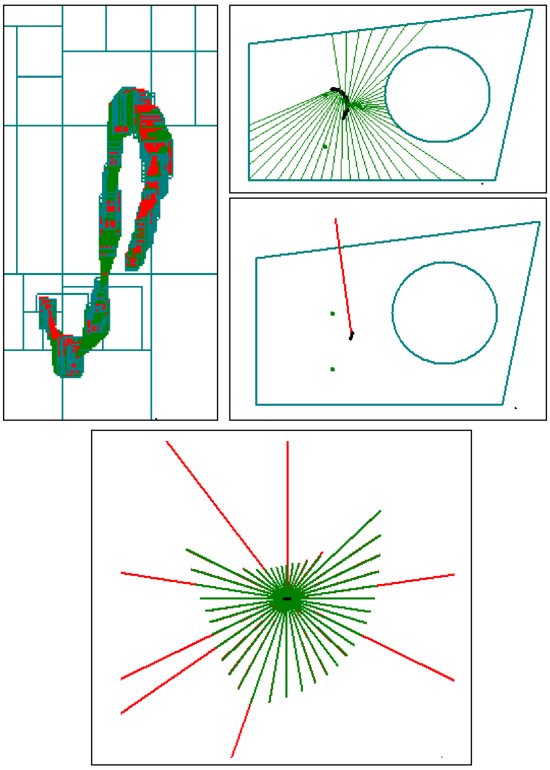
$$\begin{pmatrix} \mathbf{y}(k-\ell) \\ \vdots \\ \mathbf{y}(k) \end{pmatrix} = \begin{pmatrix} \mathbf{g} \circ \overline{\mathbf{f}}_k^{\ell+1}(\mathbf{x}(k+1)) \\ \vdots \\ \mathbf{g} \circ \overline{\mathbf{f}}_k^{1}(\mathbf{x}(k+1)) \end{pmatrix}.$$

Thus $\mathbb{X}(k+1)$ is given by the set of all $\mathbf{x}(k+1) \in \mathbb{R}^n$ such that

$$\left\{ \begin{array}{l} \bar{\mathbf{f}}_k^{\ell+1}(\mathbf{x}(k+1)) \in \mathbb{X}(k-\ell) \wedge \ldots \\ \cdots \wedge \bar{\mathbf{f}}_k^{1}(\mathbf{x}(k+1)) \in \mathbb{X}(k), \\ \bigwedge_q \left(\begin{array}{c} \left(\mathbf{g} \circ \bar{\mathbf{f}}_k^{\ell+1}(\mathbf{x}(k+1)) \in \mathbb{Y}(k-\ell)\right), \ldots, \\ \left(\mathbf{g} \circ \bar{\mathbf{f}}_k^{1}(\mathbf{x}(k+1)) \in \mathbb{Y}(k)\right) \end{array} \right). \end{array} \right. .$$

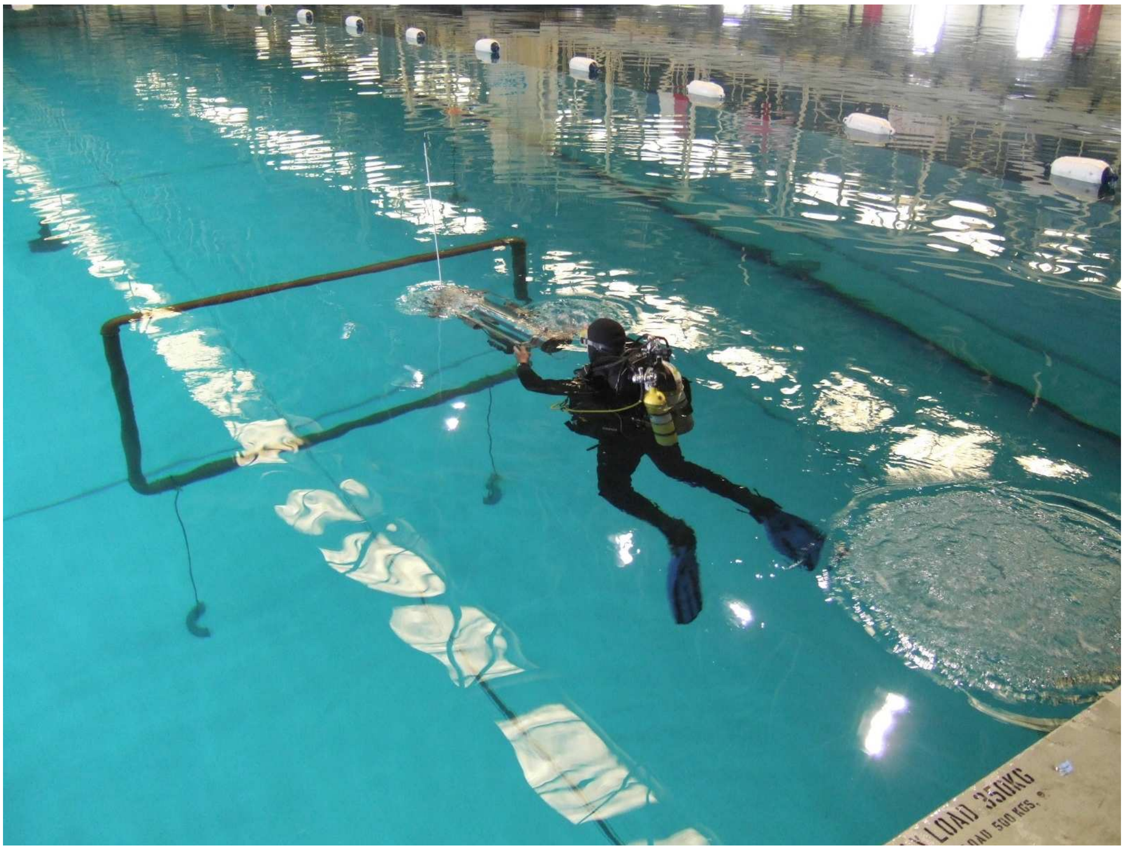
or equivalently

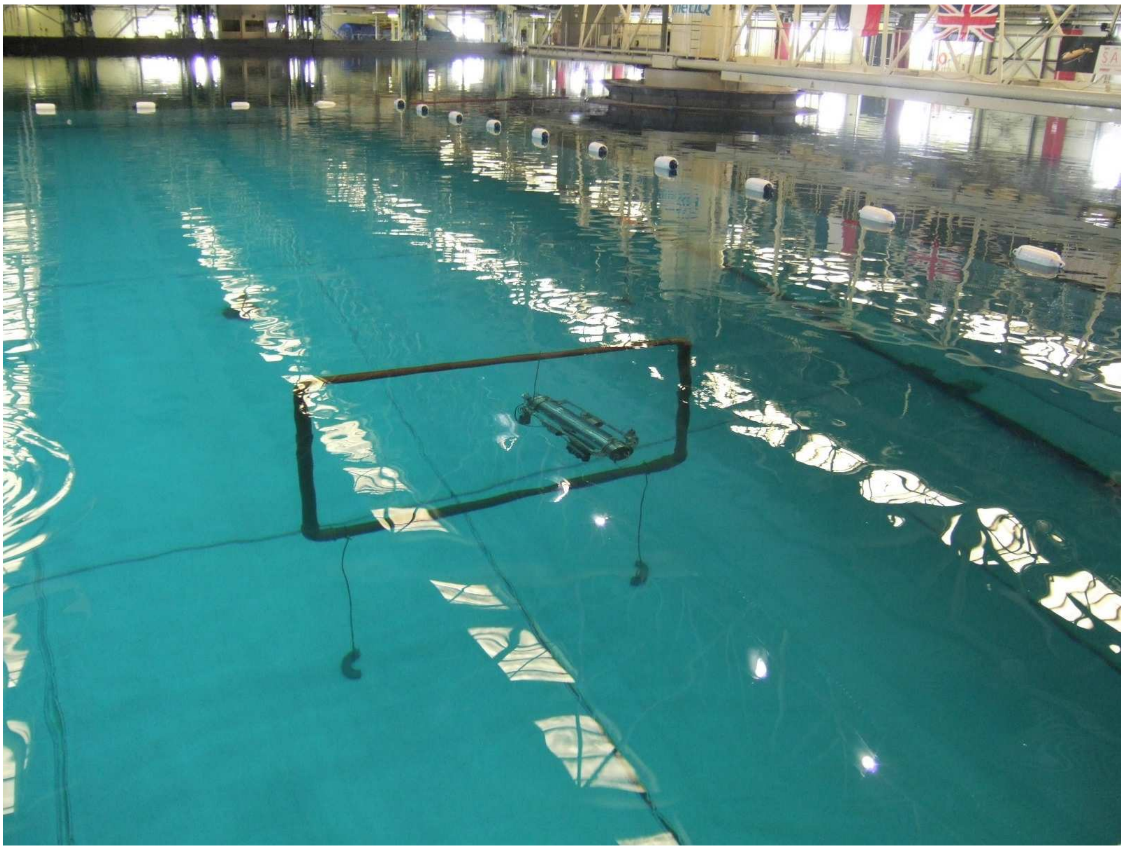
$$\begin{aligned}\mathbb{X}(k+1) &= \left(\bigcap_{i\in\{1,\ldots,\ell+1\}} \left(\bar{\mathbf{f}}_k^i\right)^{-1}(\mathbb{X}(k-i-1))\right) \\ &\quad \bigcap \left(\bigcap_{i\in\{1,\ldots,\ell+1\}}^{q} \left(\mathbf{g}\circ\bar{\mathbf{f}}_k^i\right)^{-1}(\mathbb{Y}(k-i-1))\right).\end{aligned}$$



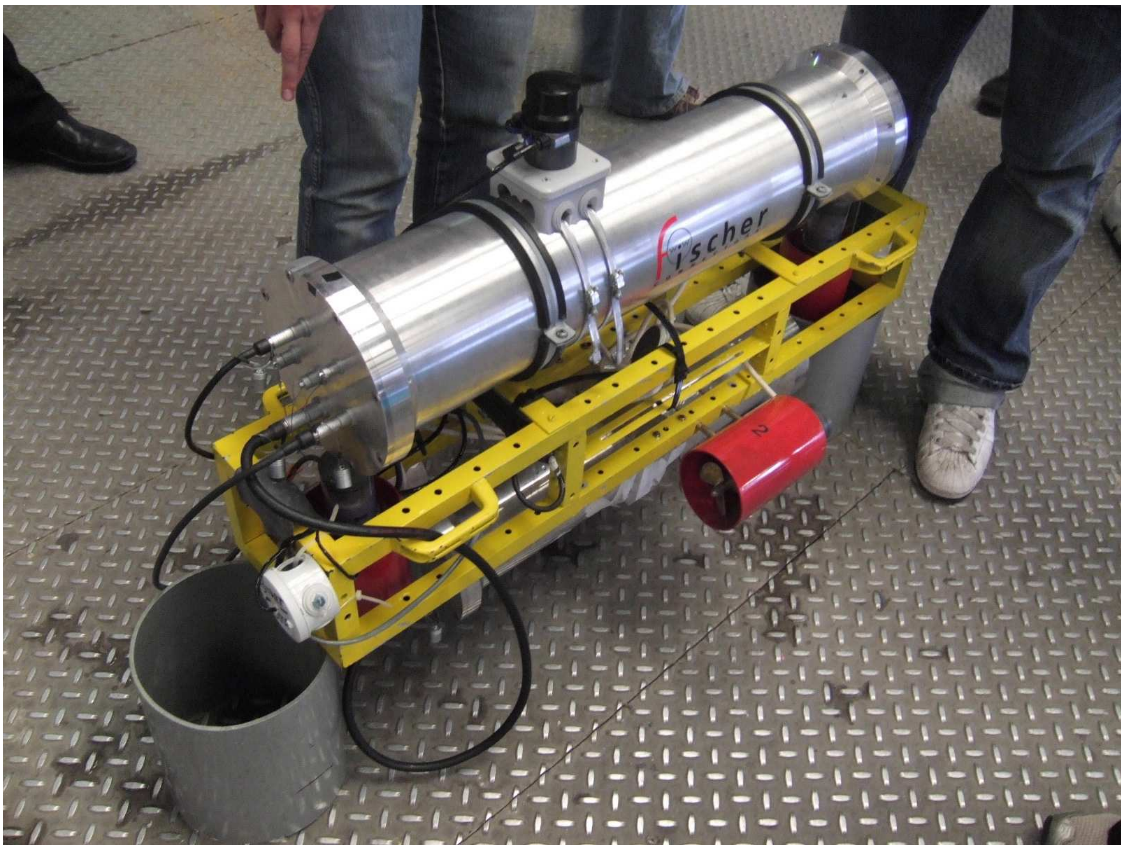
5 Competition

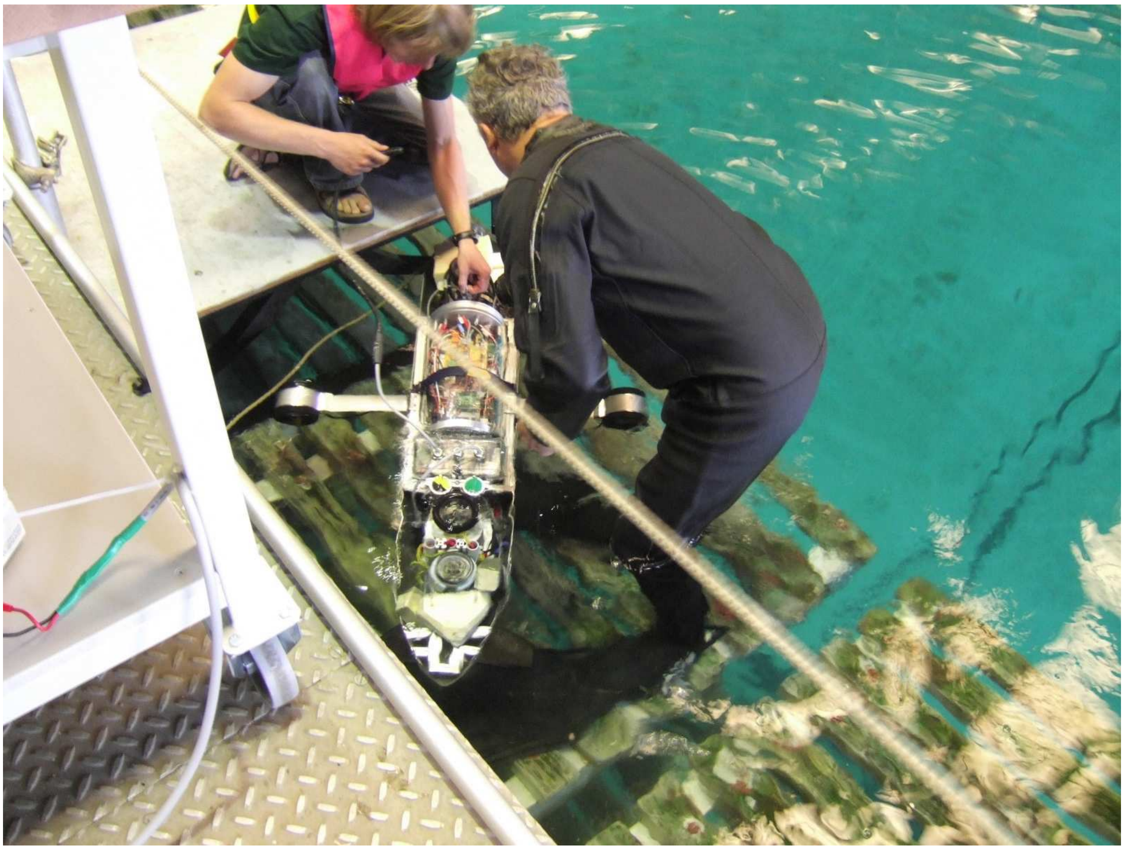

SAUC'ISSE
Ensieta
PILOT
PILOT
PILOT

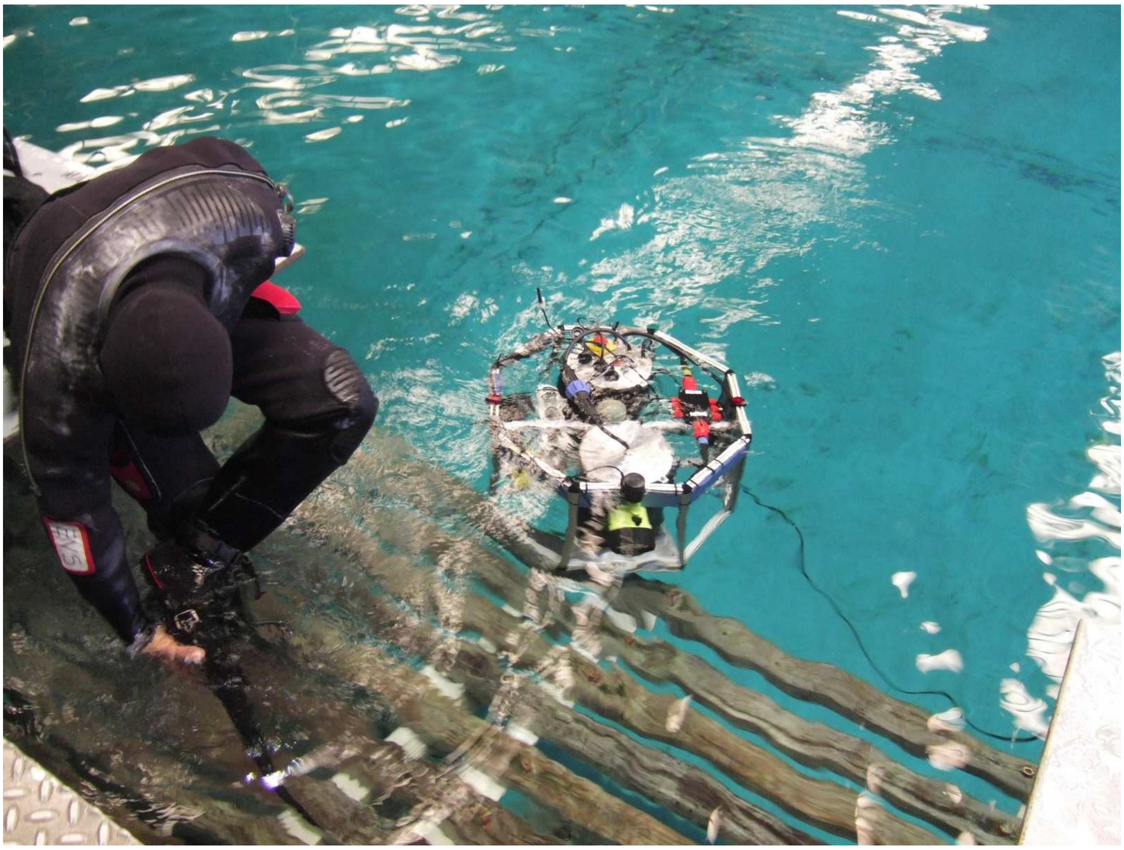

LOAD 350KG
500 KGS.




Team Nessie
bp


fischer






REAP