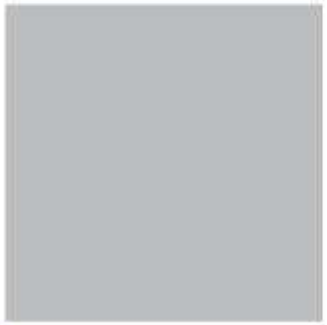
ENSTA Bretagne



SAUC'ISSE et SARDINE, 2 robots sous-marins autonomes

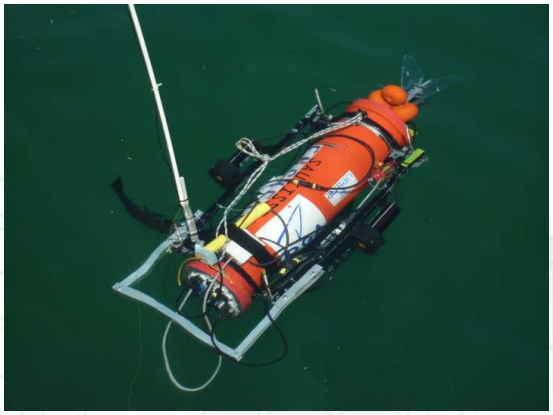
Fabrice LE BARS

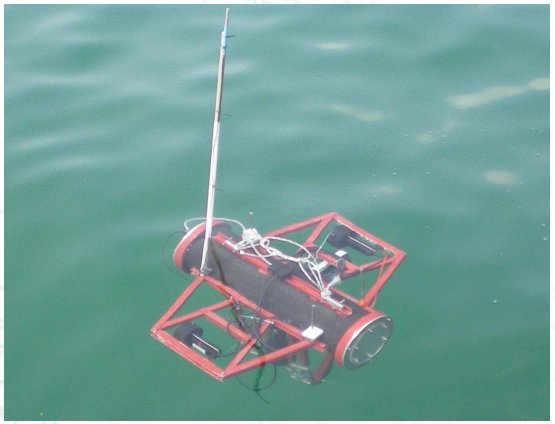
Sommaire

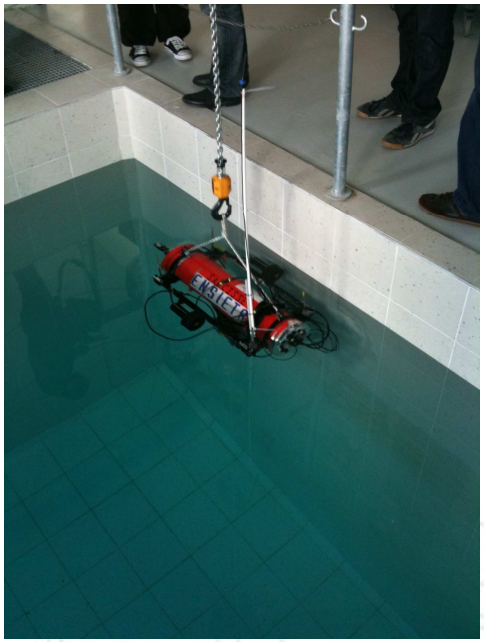
- Introduction
- SAUC-E, SAUC'ISSE et SARDINE
- Problème du SLAM
- Calcul par intervalles
- Autres robots

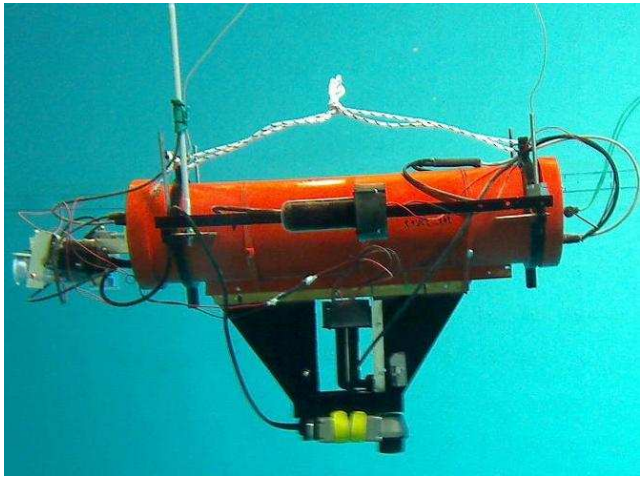
Introduction

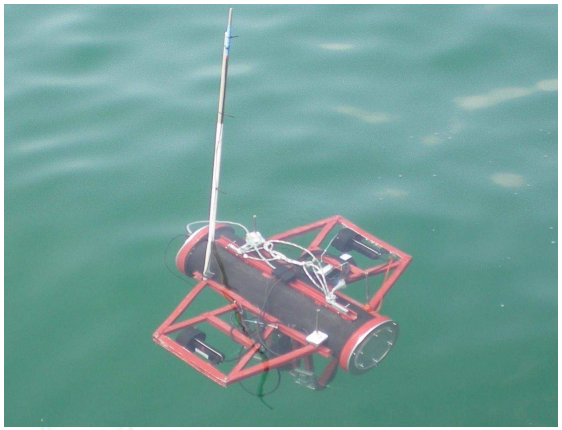
- Participation de l'ENSTA Bretagne (ex ENSIETA) au concours SAUC-E :
  - 2 robots sous-marins : SAUC'ISSE (construit en 2007) et SARDINE (construit en 2010)

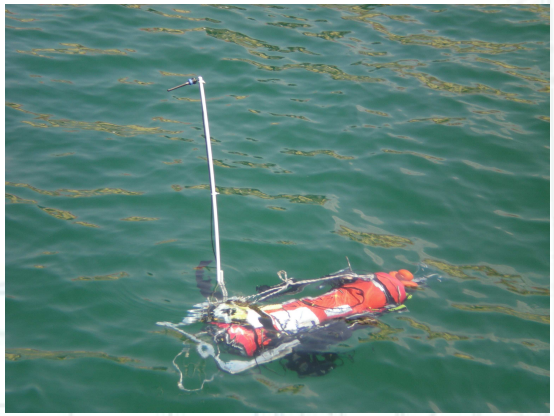








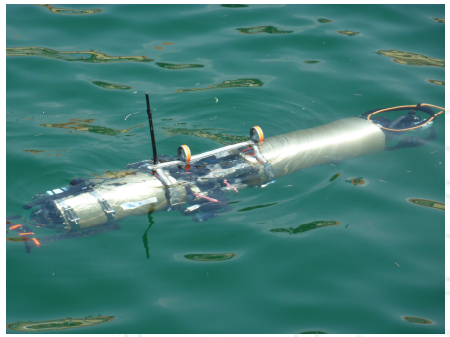
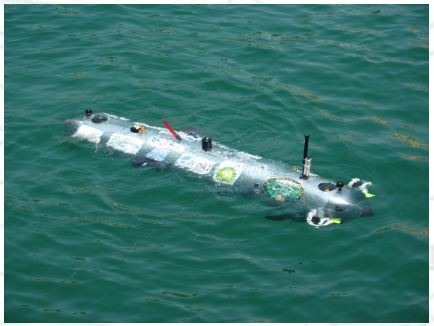


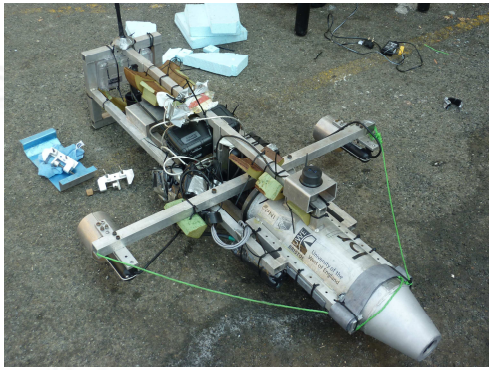


SAUC-E, SAUC'ISSE et SARDINE

Concours SAUC-E

- SAUC-E (Student Autonomous Underwater Challenge - Europe)
  - Concours de robots sous-marins autonomes dans une piscine ou un port
  - Missions : passer à travers des cadres, détecter et localiser différents objets par sonar ou caméra…

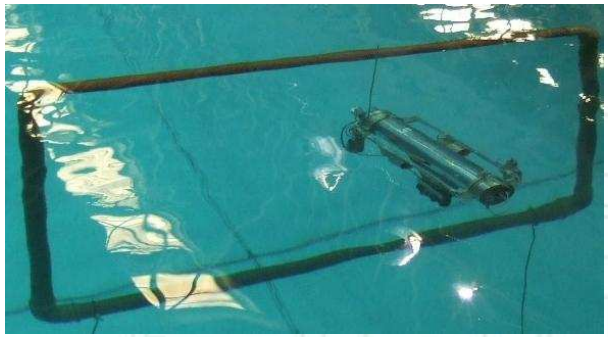



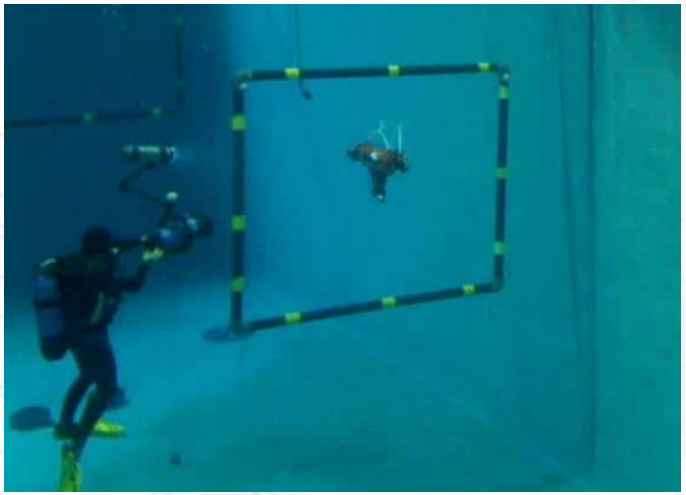


Missions du concours

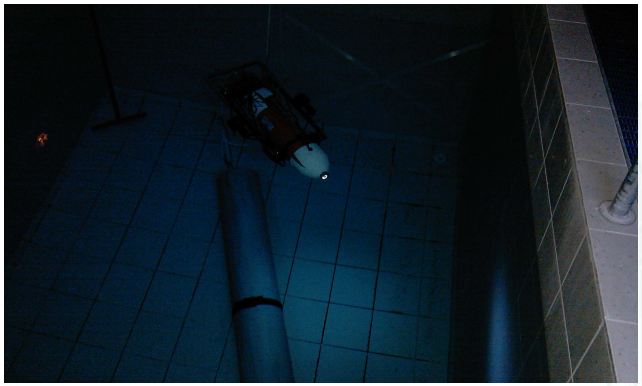
Participations

- SAUC-E (Student Autonomous Underwater Challenge - Europe)
  - 4 participations au concours : en 2007 (3ème sur 6), 2008 (2ème sur 6), 2009 (2ème sur 8) et 2010 (3ème sur 9)





SAUC'ISSE

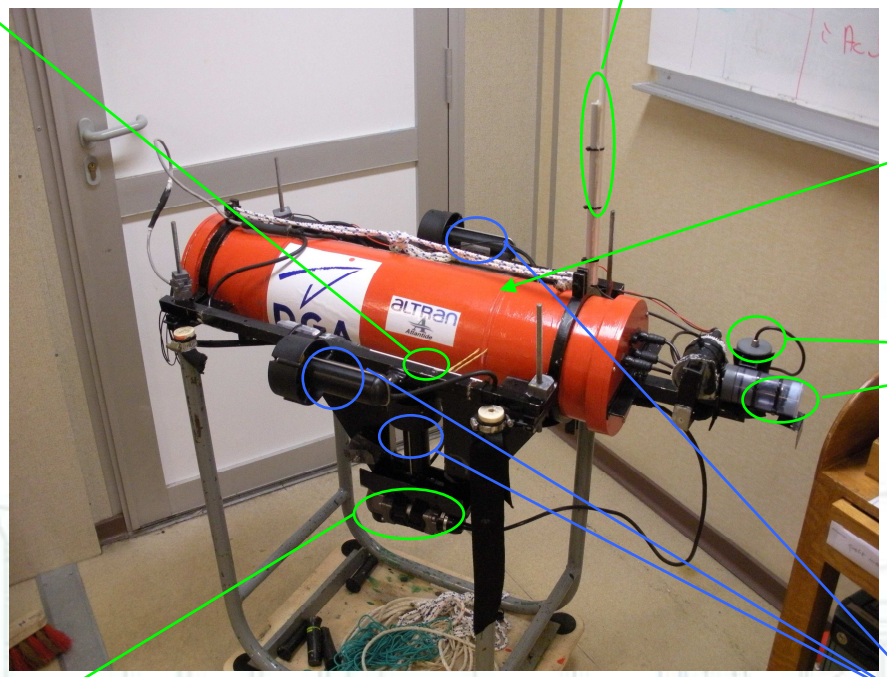


SAUC'ISSE (extérieur)

Switch :
It is a home-made watertight switch.

WIFI antenna :
It allows the communication with an external computer (up to a depth of 1 m).

Aluminium tube :
The submarine is based on an aluminium tube of 80 cm with a diameter of 20 cm. It contains the embedded electronics and the batteries. It is closed by 2 aluminium covers with IP68 connectors to connect the external peripherals with the internal devices.

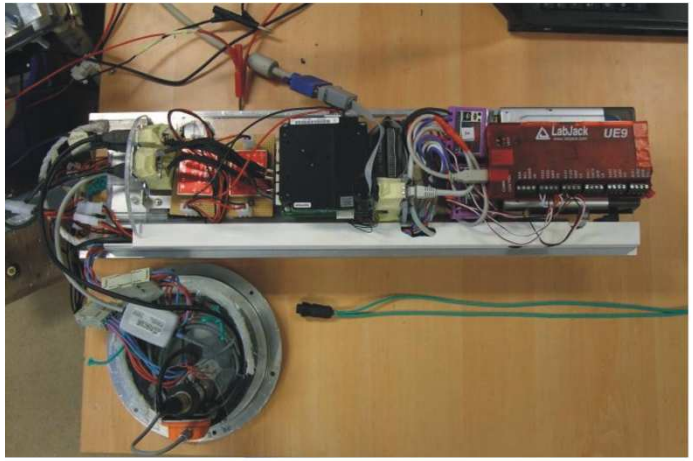


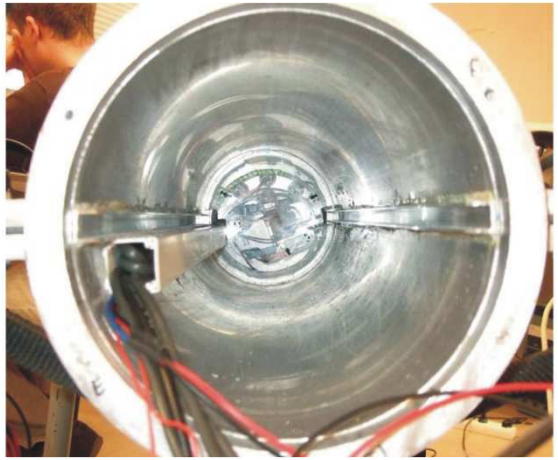
Cameras :
We use 2 watertight cameras to locate different objects in the water.

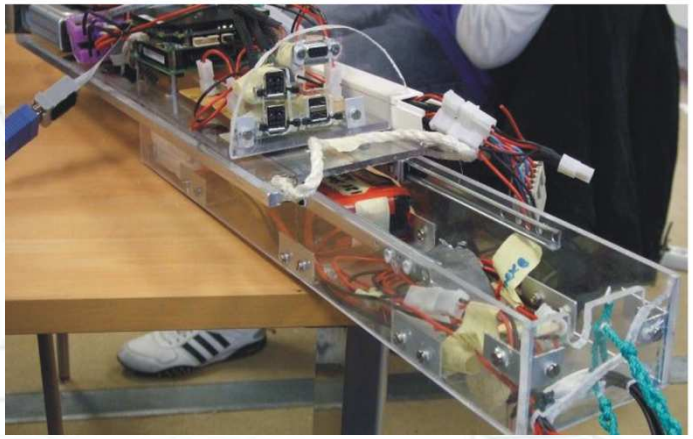
Thrusters :
2 horizontal thrusters handle the speed and the direction of the robot.
1 vertical thruster controls the depth.
The submarine is stable thanks to a heavy keel which is also used to hold the sonar and the vertical thruster.

Sonar :
The sonar is used to localize the submarine by trying to detect the borders of the water area. The sonar makes a continuous scan of 360°.

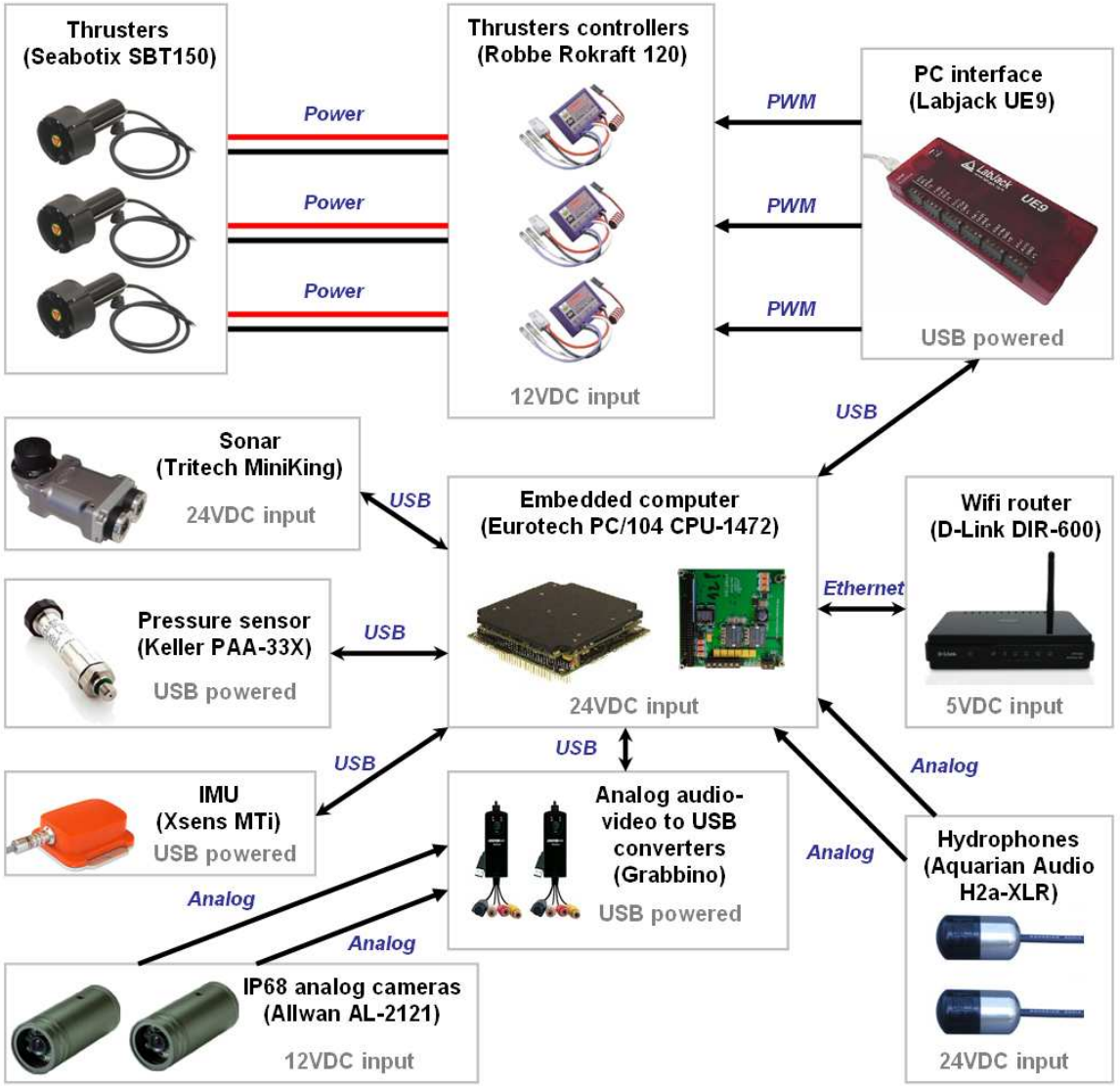
SAUC'ISSE (intérieur)



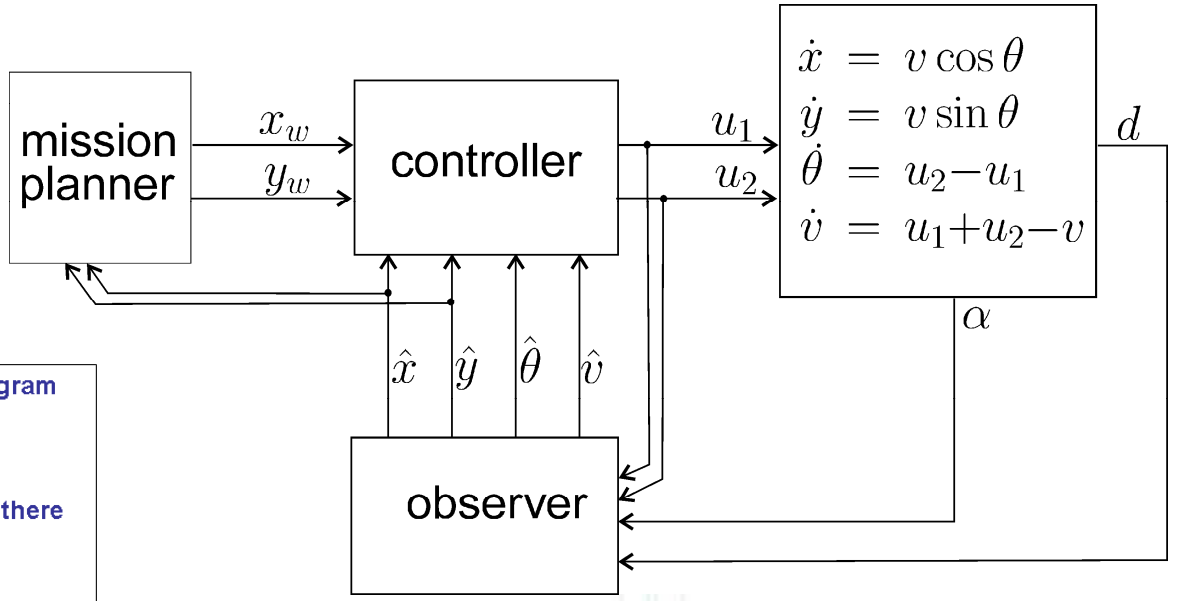




SAUC'ISSE (électronique)

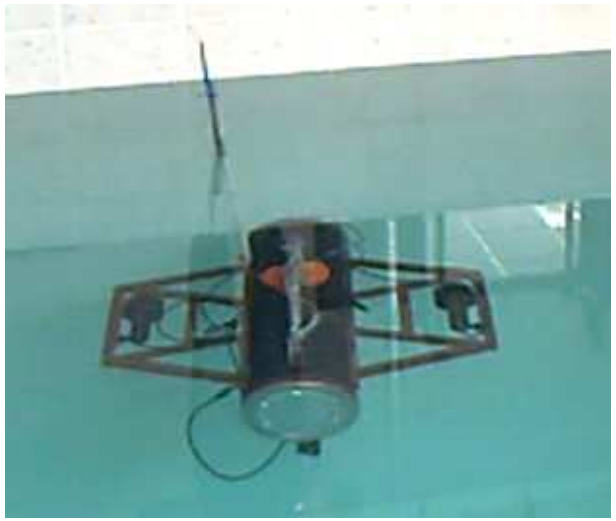


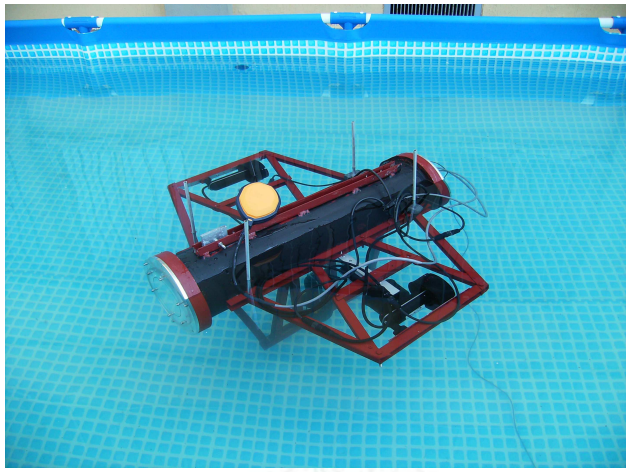
SAUC'ISSE (informatique)

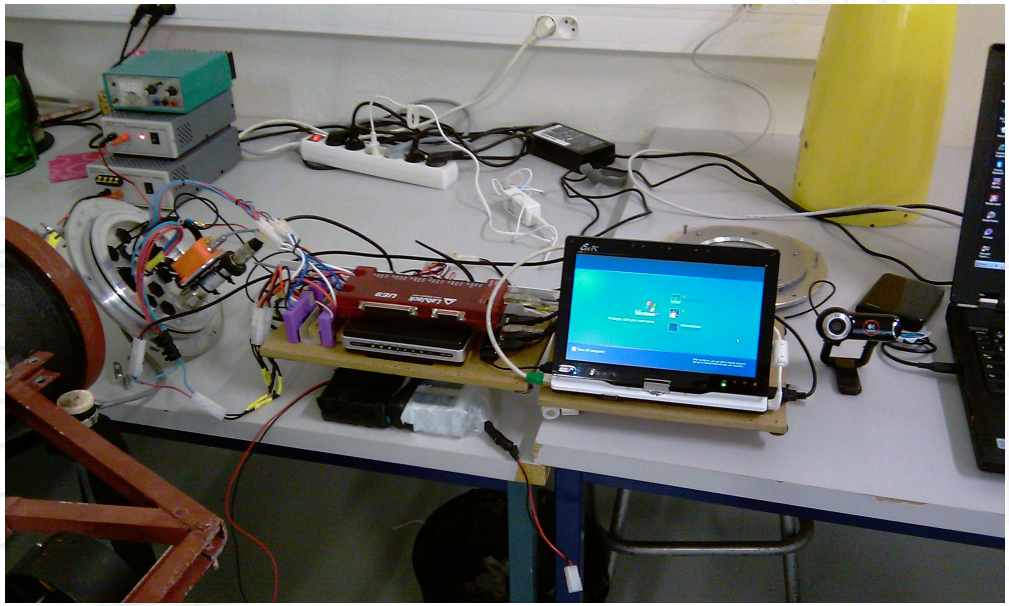



```
% this is a commentary. It is ignored by the program
% Example 1: go to depth -3m
%% step1: start depth regulation
depthreg -3
%% step2: wait 15s for the robot to actually get there
wait 15
% Example 2: some random movements at -3m
heading 1
wait 5
thrust 1
wait 10
stop
heading 1.57
wait 10
% Example 3: high level action
configureWallFollowing 12.5 1.2 -1.57 0 0.3 0.2 0.5
startWallFollowing
wait 20
stopWallFollowing
generalstop
depthreg 0
```

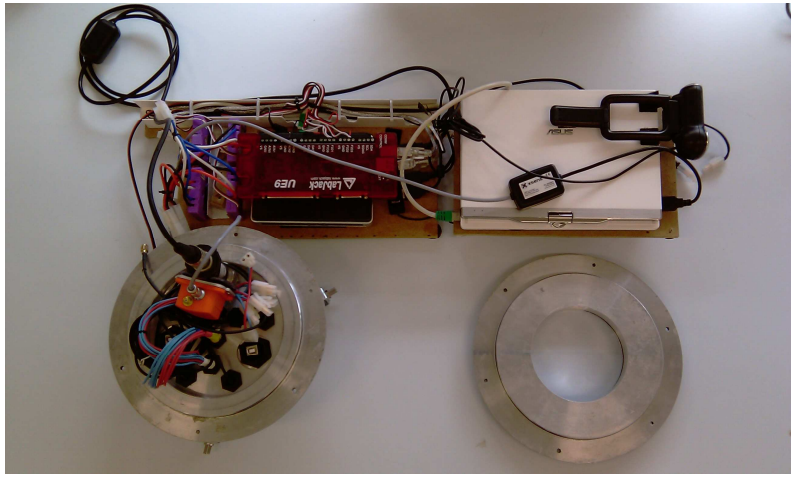
SARDINE

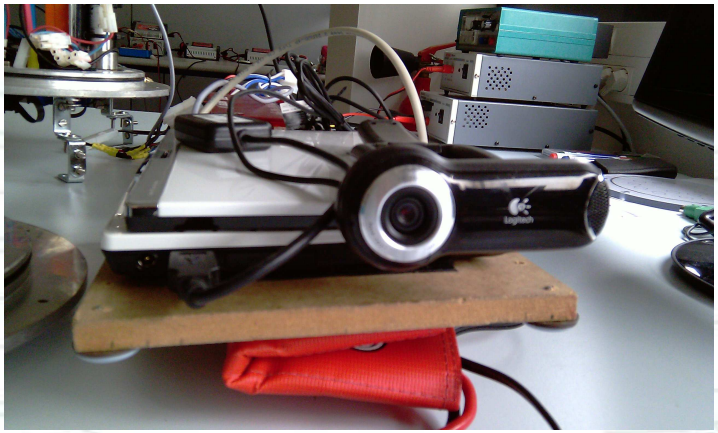


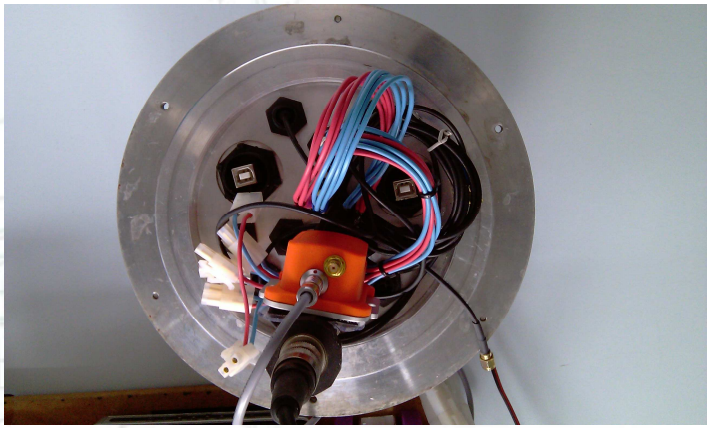




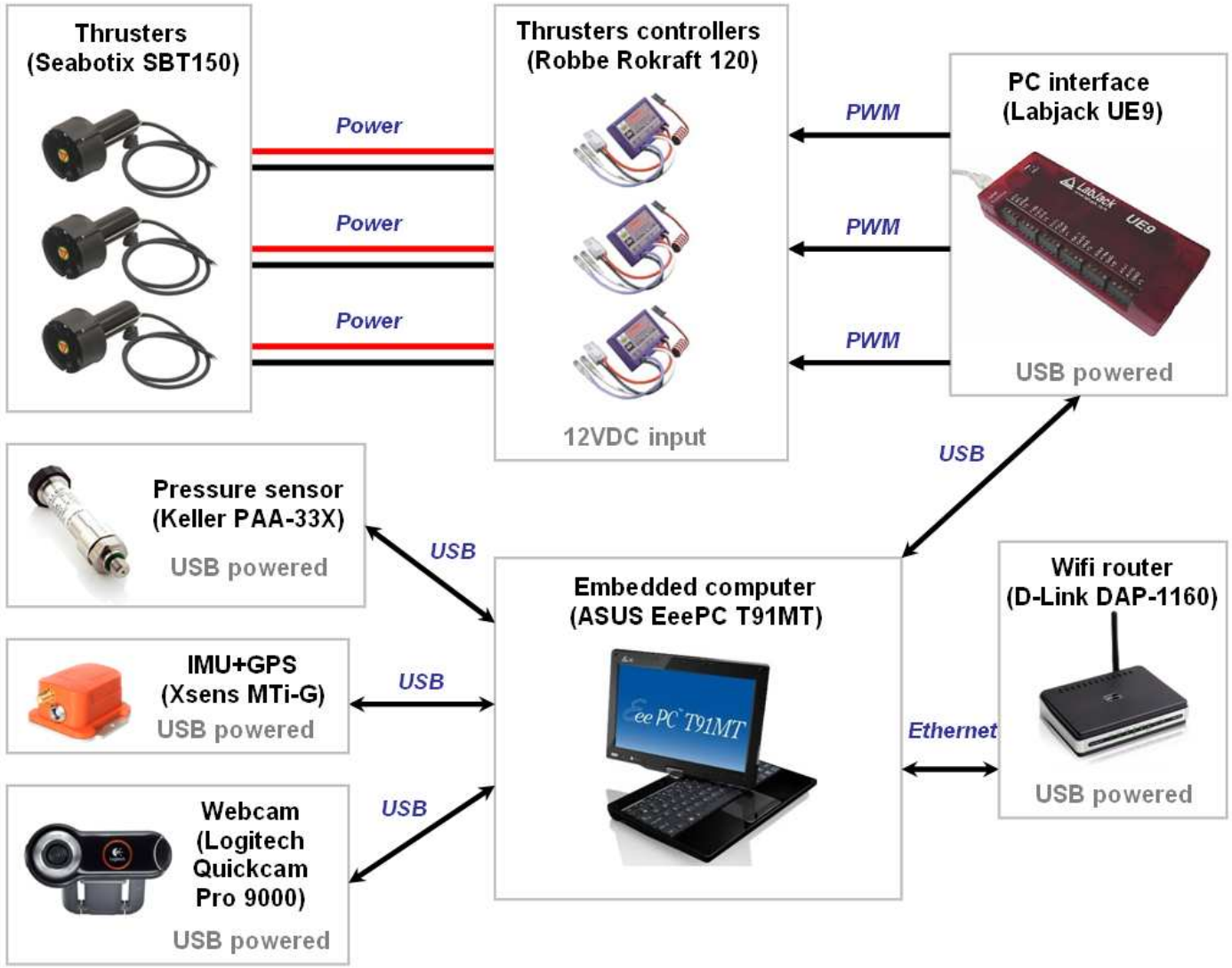
SARDINE



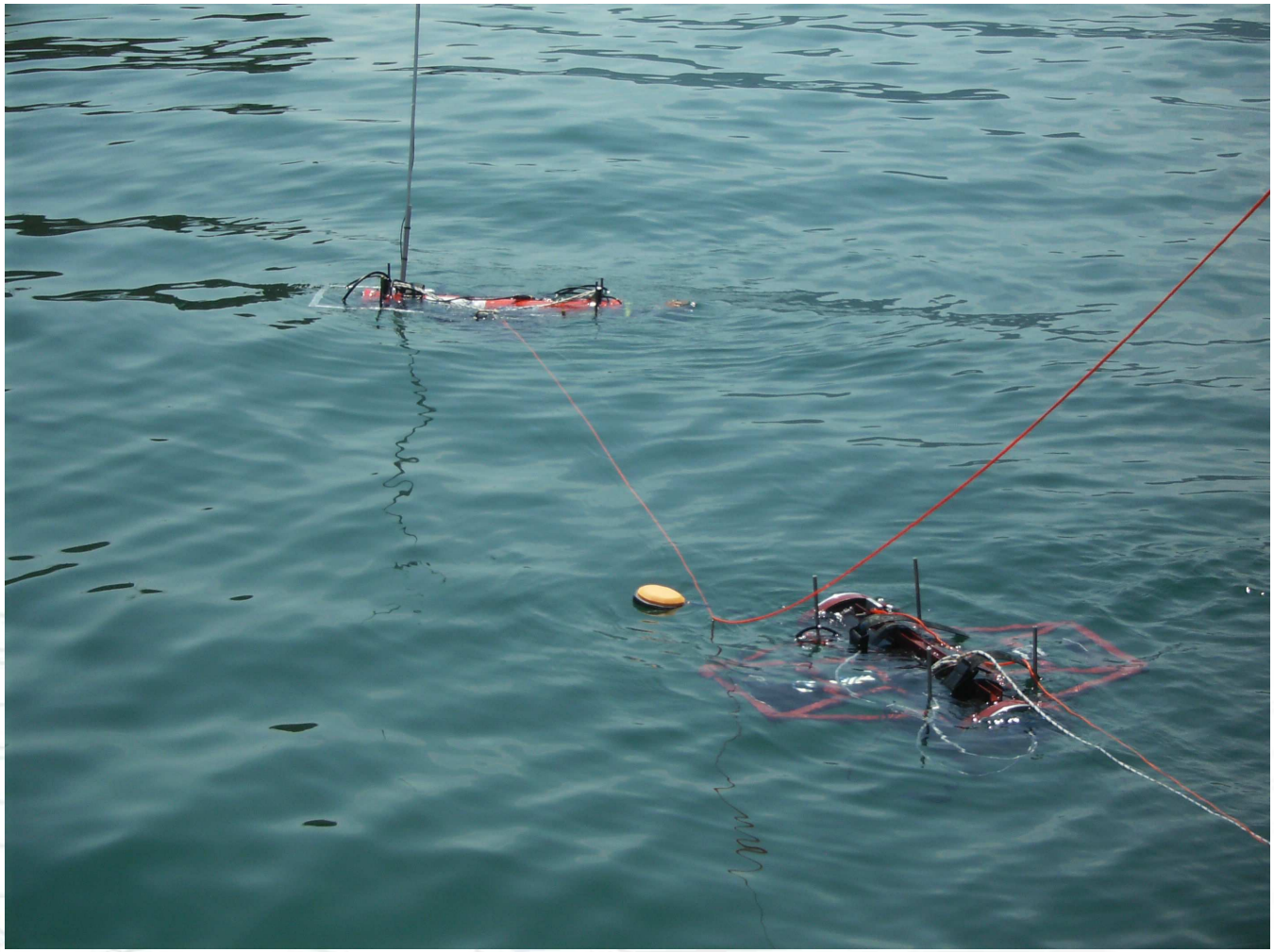


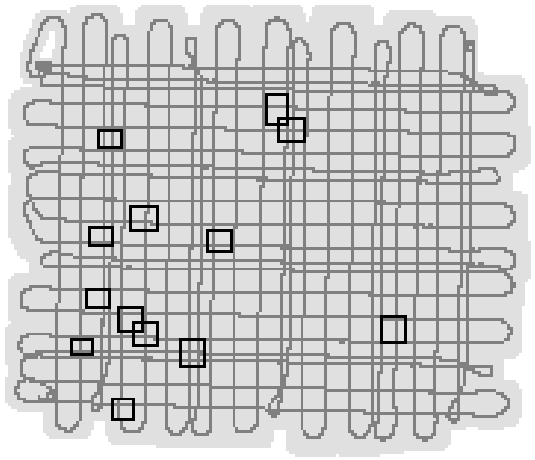


SARDINE (électronique)



Vers une meute de robots…





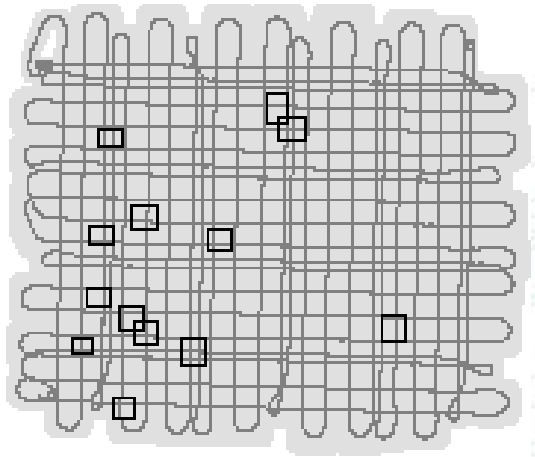
Problème du SLAM

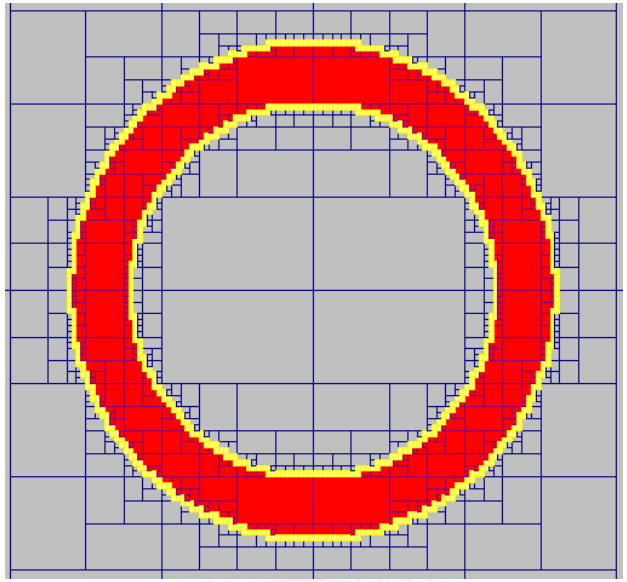
Problème du SLAM

- **SLAM** = Simultaneous Localization And Mapping : **se localiser** dans un **environnement inconnu** en prenant des **points de repères**
- Un robot sous-marin connait :
  - Sa position initiale (GPS en surface)
  - Son modèle de déplacement (équations)
  - Des données de navigation (profondeur, orientation, vitesse)
  - Des données sur ce qui l'entoure (images sonar, caméra)
- Plus il avance, plus ses erreurs d'estimation de position s'accumulent : le robot se perd

Problème du SLAM

- **Idée** du SLAM :
  - **Relever la position d'objets quand on se localise bien** (cartographie à partir de la localisation)
  - **Utiliser** la position de ces **points de repères** quand on repasse devant et qu'on est perdu pour retrouver la trajectoire qu'on a faite (localisation grâce à la cartographie)





Calcul par intervalles

Calcul par intervalles

- Différentes **approches** existent pour traiter le problème du SLAM :
  - **Probabilistes**

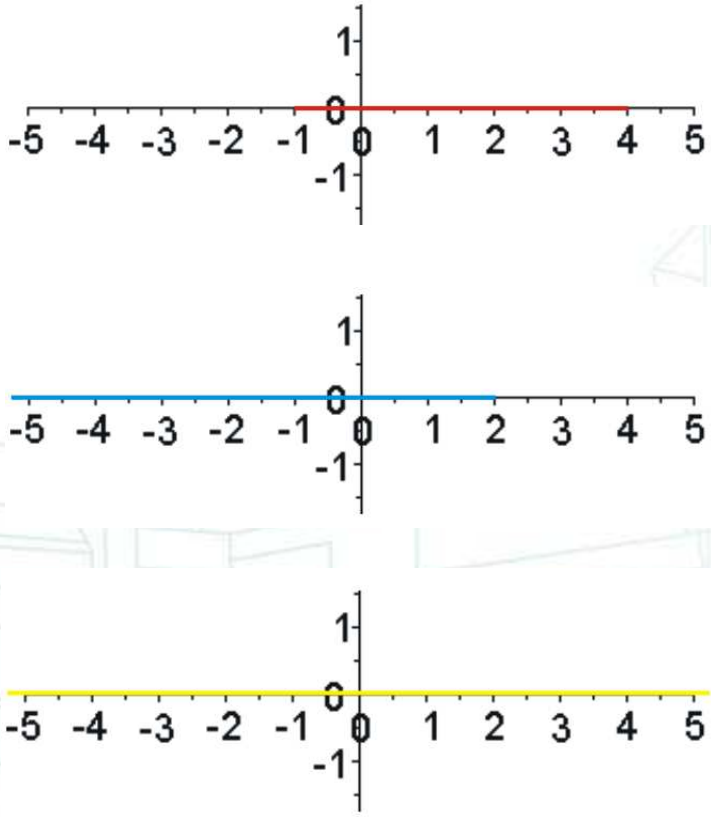
    => On obtient les solutions les plus probables
  - **Ensemblistes**

    Intervalles : [x-err,x+err]

    => On obtient des intervalles dans lesquels on est sûr qu'il y a les solutions

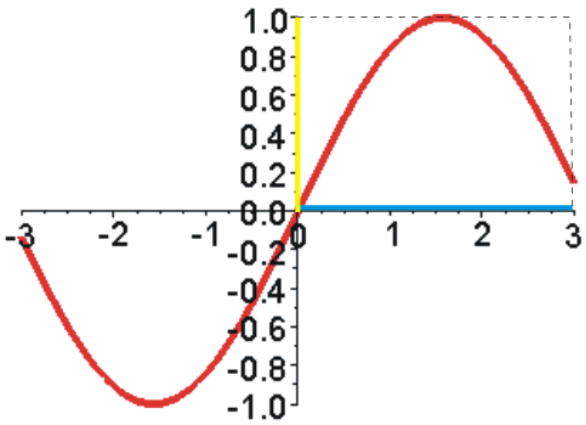
Calcul par intervalles

- [-1,4], [-oo,2],[-oo,oo] sont des exemples d'intervalles



Calcul par intervalles

- Operations {+,-,*,/} :
  - [x-,x+] op [y-,y+] = plus petit intervalle contenant l'ensemble des valeurs possibles
  - [-1,4]+[2,3]=[1,7]
  - [-1,4]*[2,3]=[-3,12]
  - [-1,4]/[2,3]=[-1/2,2]
- Multiplication par un nombre, intersection, union
  - 2[-1,4]=[-2,8]
  - [-1,3] inter [2,4]=[2,3]
  - [-1,2] union [3,4]=[-1,4]
- Image par une fonction
  - sin([0,pi])=[0,1]



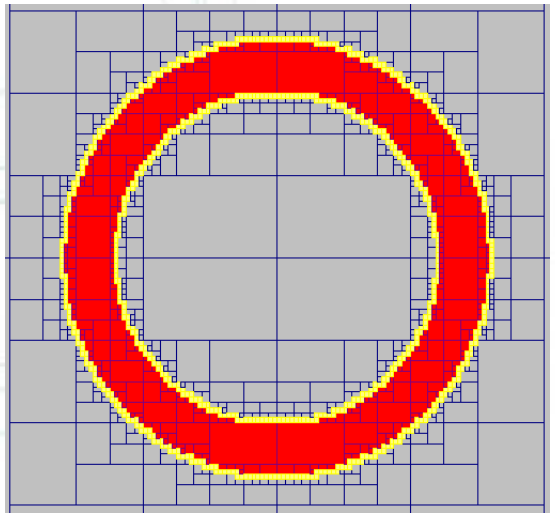
Calcul par intervalles

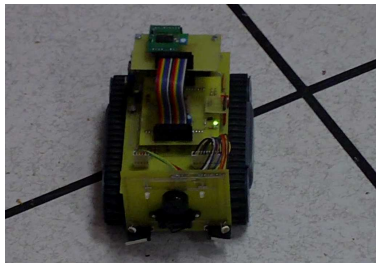
- Contraction et propagation :
  - Si on sait que z=x+y et z=[-1,1], x=[0,1], y=[0,1], alors on a

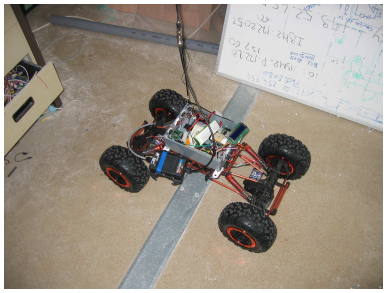
    z=x+y=[0,1]+[0,1]=[0,2]

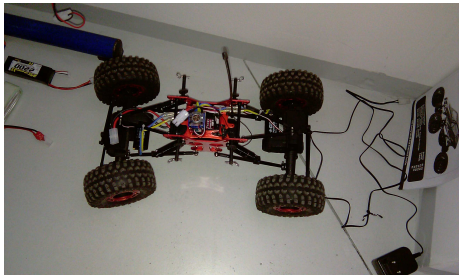
    Or z=[-1,1] au départ donc z=([-1,1] inter [0,2])=[0,1]

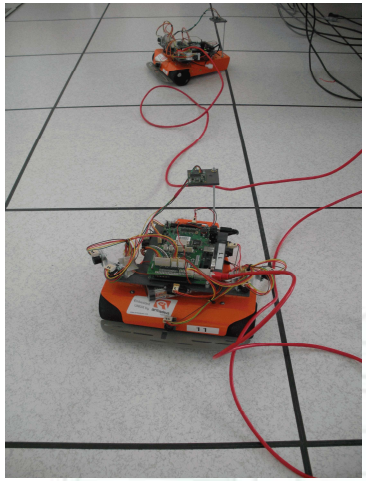
    y=z-x…
  - On peut donc résoudre des équations de cette façon…
- D'autres techniques sont disponibles en plus (bissections…)

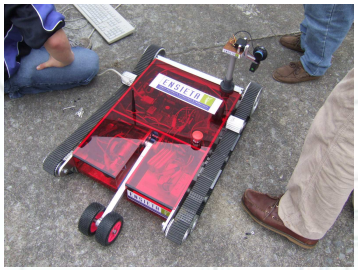


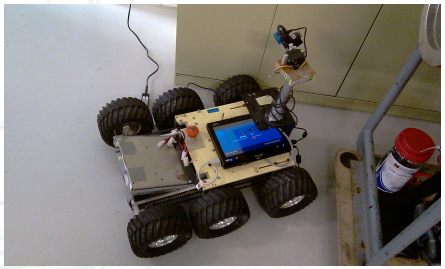




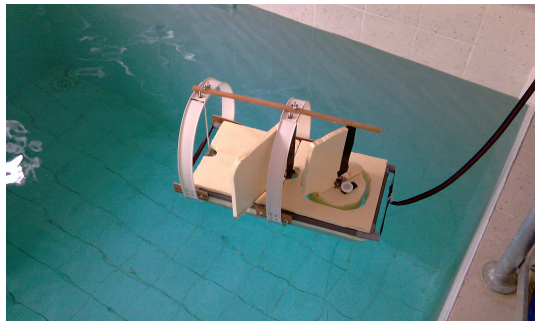








Les autres robots de l'ENSTA Bretagne



Les autres robots de l'ENSTA Bretagne

Questions?

- Liens utiles pour plus d'informations
  - http://www.ensta-bretagne.fr/lebars/
  - http://media.ensta-bretagne.fr/robotics/
  - http://www.youtube.com/user/ensietarobotics
  - http://www.facebook.com/pages/SAUCISSE/142805275731790?ref=sgm
- Contacts
  - fabrice.le_bars@ensta-bretagne.fr
  - robotique@ensta-bretagne.fr