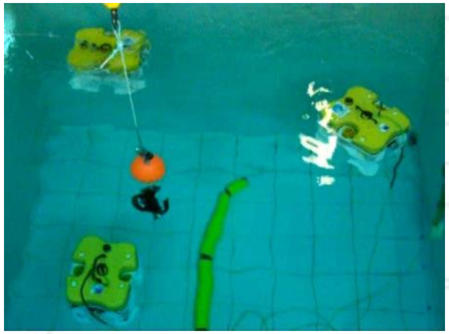
ENSTA Bretagne

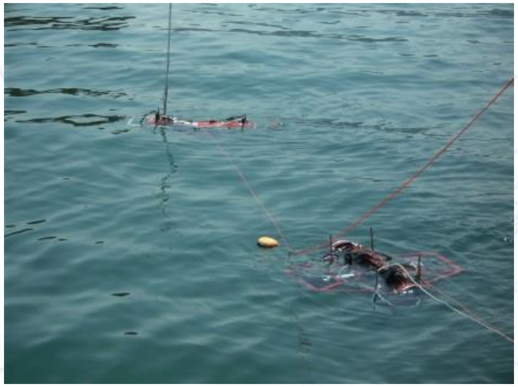
Localization problems for an AUV

Introduction

Localization problems in mobile robotics

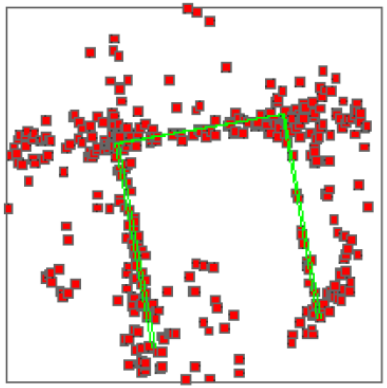
- To be able to know what to do, an autonomous mobile robot first needs to know its current state, especially its position

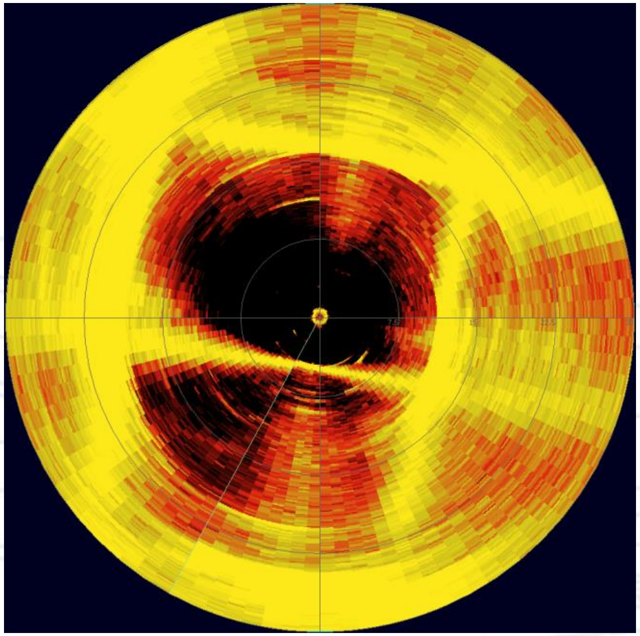




Localization problems in mobile robotics

- Problems
  - Non-linear dynamic model of the robots
  - Deal with outliers in measurements
  - Fusing different types of data
  - Representation of uncertainties





Sonar localization for an AUV

Localization problems in mobile robotics

- Different localization scenarios for an AUV
  - Dead-reckoning using compass and thrusters inputs or using DVL
  - Static sonar localization when inside a known basin
  - Dynamic sonar localization when inside a known basin

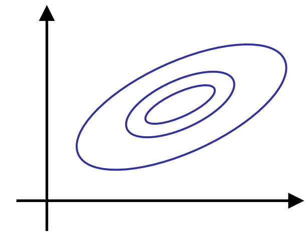
Interval analysis

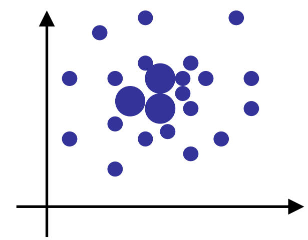
Uncertainties

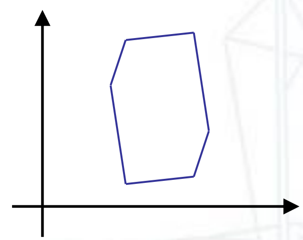
- Representations of uncertainties
  - Probabilistic methods
    - Gaussian
    - Particles

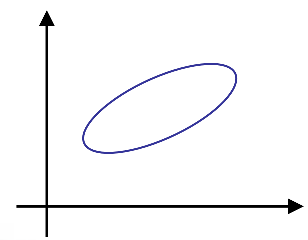
    => Try to get most probable solutions
  - Set-membership methods
    - Zonotopes
    - Ellipsoids
    - Intervals

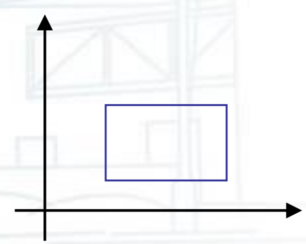
    => Try to enclose all possible solutions

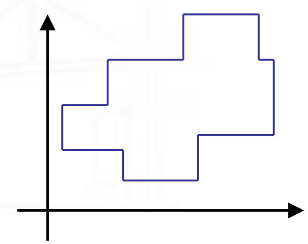






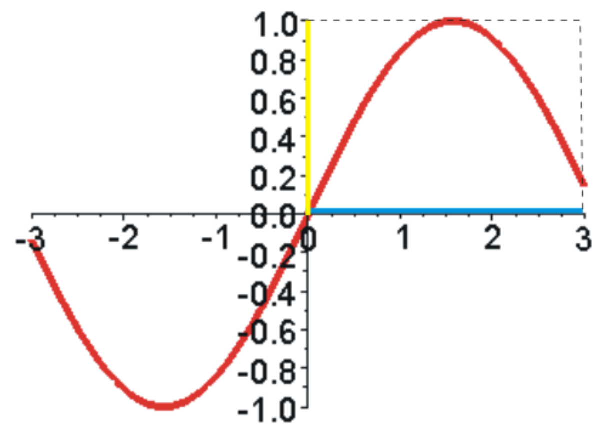






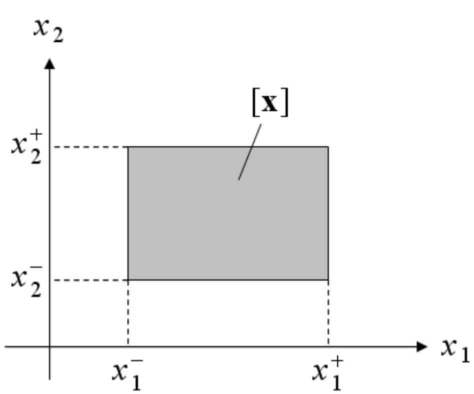
Interval arithmetic

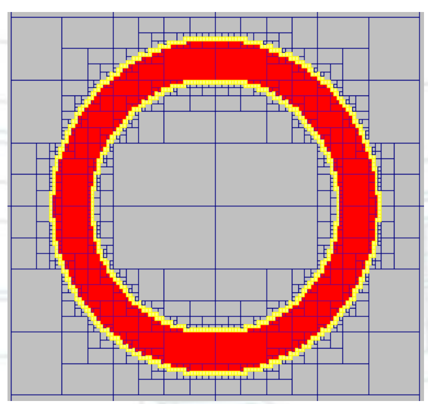
- $[-\infty, 2]$, $[-1, 4]$, $[-\infty, \infty]$ are examples of intervals
- Operations $\diamond \in \{+, -, *, /\}$
  - $[x^-, x^+] \diamond [y^-, y^+] =$ smallest interval containing the set of all possible values for $x \diamond y$
  - $[-1, 4] + [2, 3] = [1, 7]$
  - $[-1, 4] * [2, 3] = [-3, 12]$
  - $[-1, 4]/[2, 3] = [-1/2, 2]$
- Multiplication by a number, intersection, union
  - $2[-1, 4] = [-2, 8]$
  - $[-1, 3] \cap [2, 4] = [2, 3]$
  - $[-1, 2] \sqcup [3, 4] = [-1, 4]$
- Image by a function
  - $\sin([0, \pi]) = [0, 1]$

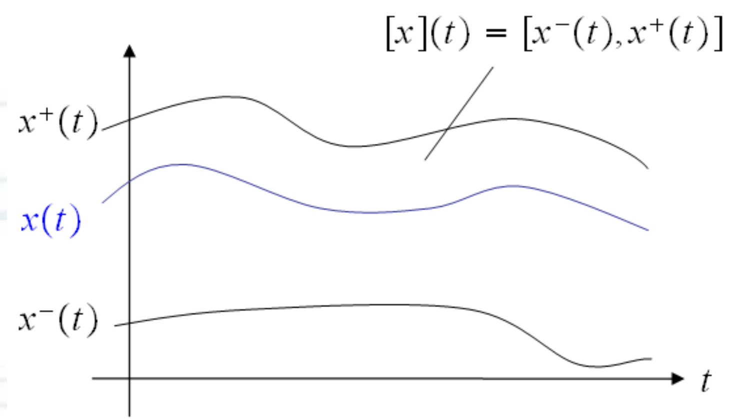


Interval analysis

- Real intervals can be generalized
  - Vectors intervals (boxes)
  - Sets intervals
  - Functions intervals (tubes)
  - Any set with a lattice structure





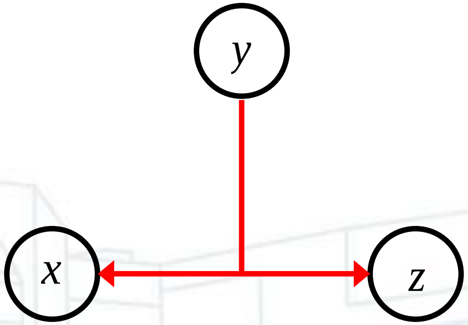


Interval analysis

- CSP (Constraint Satisfaction Problem)
  - A CSP is made of
    - A set of variables
    - A set of domains enclosing the variables
    - A set of constraints between the variables

Interval analysis

- Contraction
  - If $z^2 = \exp(x) + y$ and $x \in [1,4]$, $y \in [3.1, 3.2]$, $z \in [4,7]$, then
    - $x = \ln(z^2 - y) \Rightarrow x \in [x] \cap \ln\left([z]^2 - [y]\right) = [2.5, 3.9]$

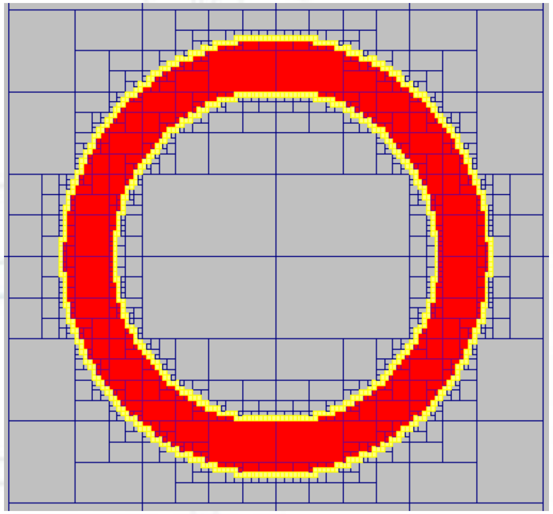


Interval analysis

- Contraction and propagation
  - We call *contractor* an operator that reduce the domain of variables
  - A *propagation* is a repeated call to contractors
  - We can repeat contractions until a fix point

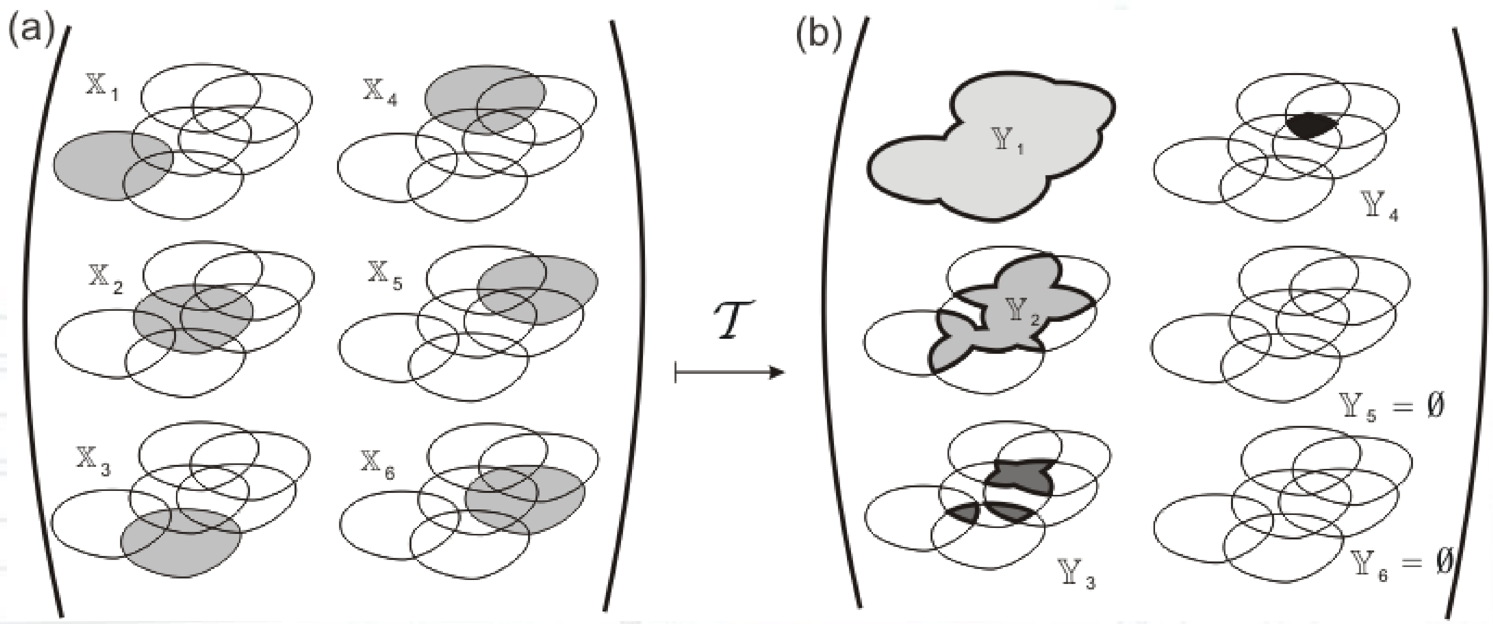
Interval analysis

- Other techniques such as bisections can also be done



Interval analysis

- Handling outliers : relaxed intersection (q-intersection)
  - Example of a CSP with 6 constraints : C_1,C_2,C_3,C_4,C_5,C_6 where 2 constraints are inconsistent



Sets that satisfy only one constraint

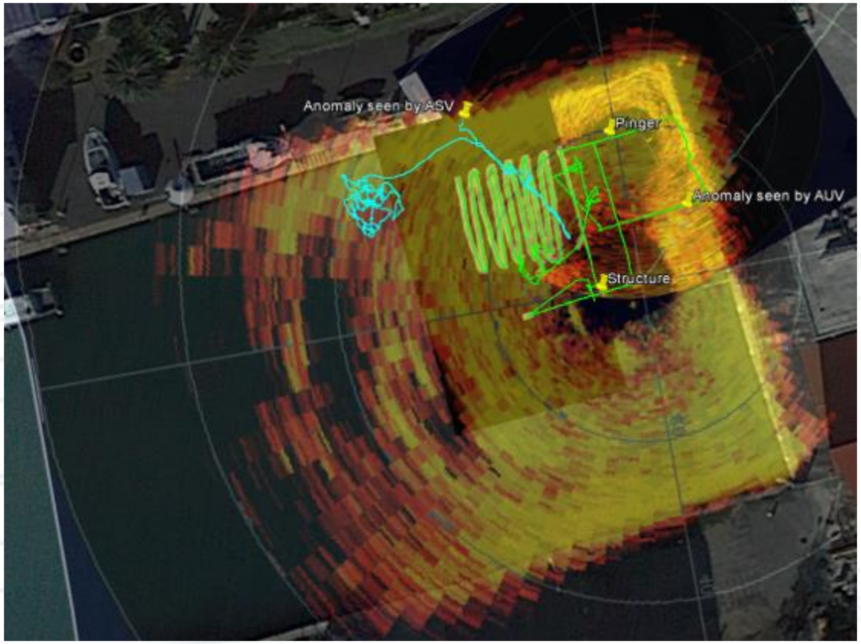
Sets that satisfy only a defined number of constraints

Interval analysis

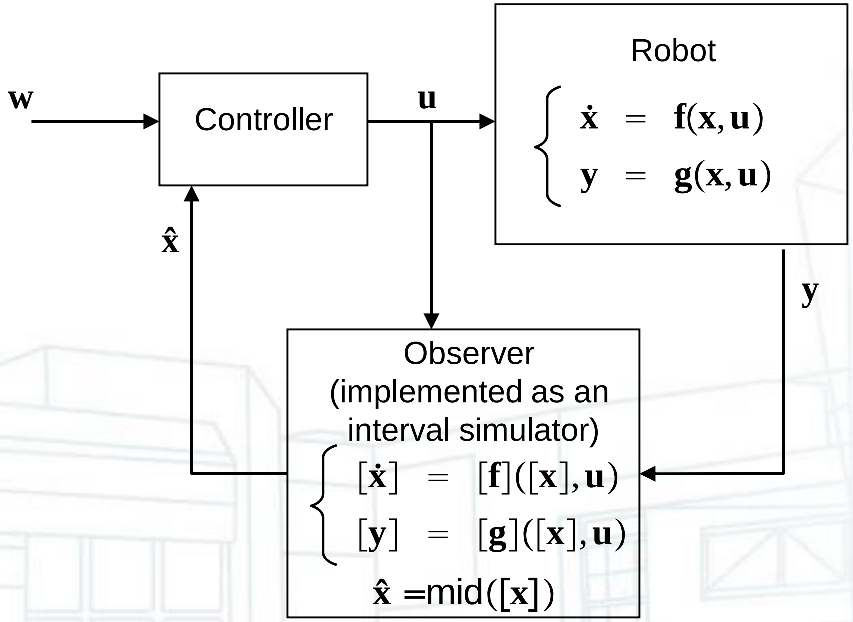
- To learn
  - IAMOOC : https://www.ensta-bretagne.fr/jaulin/iamooc.html
  - pyIbex : http://www.ensta-bretagne.fr/desrochers/pyibex
  - EASIBEX-MATLAB : https://github.com/ENSTABretagneRobotics/EASIBEX-MATLAB

Localization for an AUV

- Examples of different localization scenarios for an AUV
  - Dead-reckoning using compass and thrusters inputs or using DVL
  - Static sonar localization when inside a basin
  - Dynamic sonar localization



Dead-reckoning



Dead-reckoning

- Simplified state equations of the AUV using compass and thrusters inputs
  - Need some known position, e.g. using GPS when on surface
  - Need to set coefficients related to the speed of the robot depending on inputs, damping, etc.
  - Possible forward and backward propagation in time using interval analysis

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= u_2 - u_1 \\ \dot{v} &= u_1 + u_2 - v \end{aligned}$$

Dead-reckoning

- Example

```
% Known initial state.
x_hat=[[-0.1,0.1];[-0.1,0.1];[-0.1,0.1];[10-0.1,10+0.1]];

while (bExit == 0)
    % Simulator.
    x = x+f(x,u)*dt; % Simulated evolution equation of the robot.
    y = g(x); % Simulated observation equation.

    % Interval observer.
    x_hat = i_Add(x_hat,dt*f_hat(x_hat,u)); % Estimated state of the robot using evolution equation.
    x_hat(3,:) = i_Inter({x_hat(3,:);[y(3)-0.1,y(3)+0.1]}); % Contract estimated state using observation equation (compass)

    draw_hat(x_hat); % Draw estimated position.
    draw(x); % Simulated robot.

    % Wait a little bit.
    pause(dt);
    t = t+dt;
end
```

Dead-reckoning

- Problems
  - This simple model assumes no lateral movement, and there is no sensor to estimate it
  - Estimation errors accumulate quickly with time since there are no position or speed measurements

Dead-reckoning

- If we add a DVL
  - Measure speed, so errors accumulate more slowly
  - We have more general equations, since any lateral movement is measured

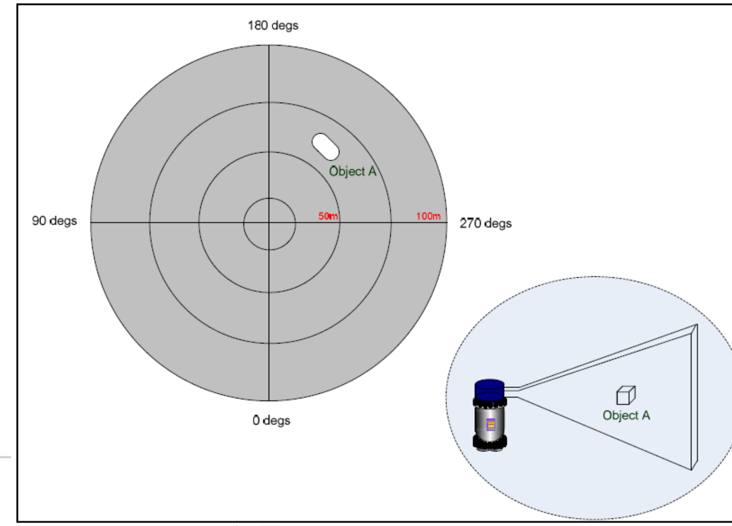
$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = R(\phi, \theta, \psi) \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}$$

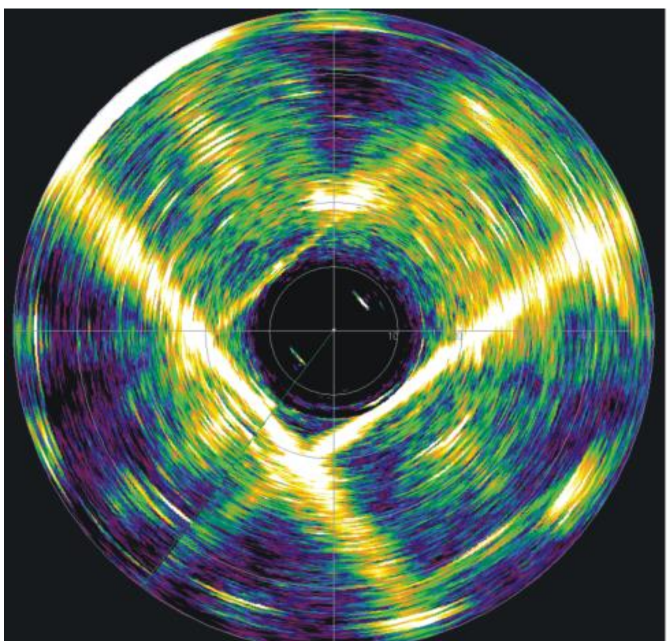
Nortek DVL 1 MHz

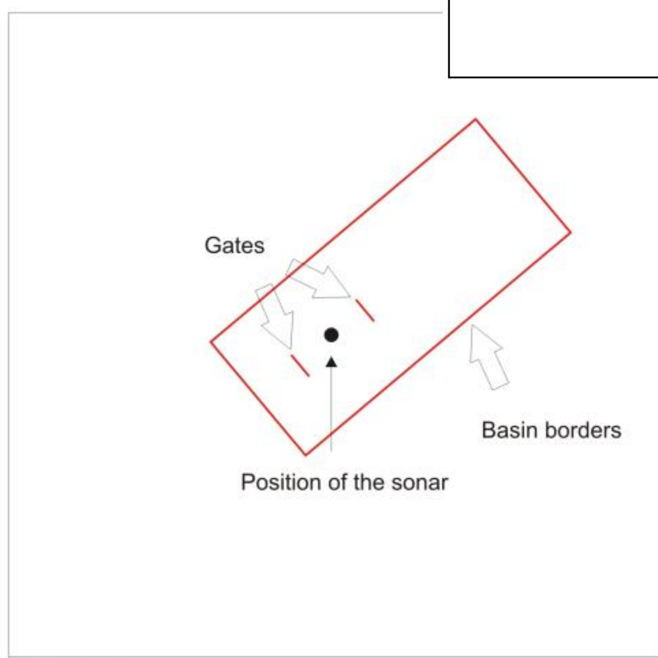
TRDI Pathfinder

Static sonar localization

- Using a sonar in a known environment, we can stop the accumulation of estimation errors
  - Mechanical rotating sonar

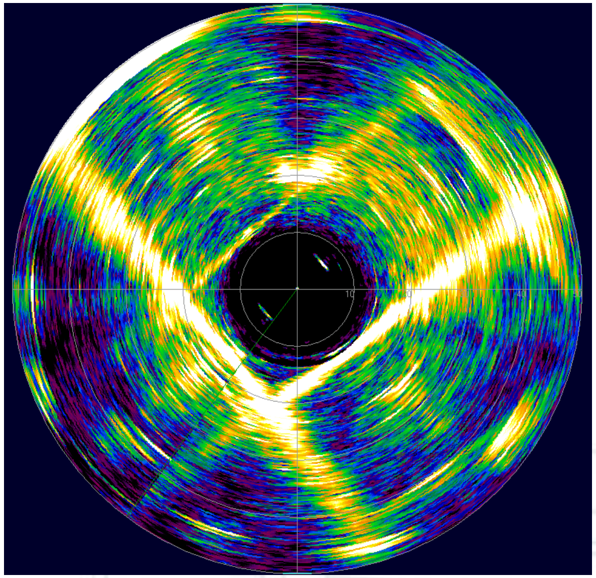


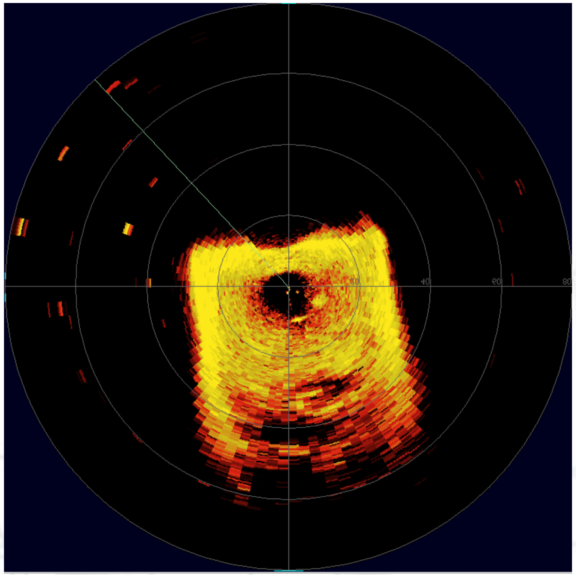


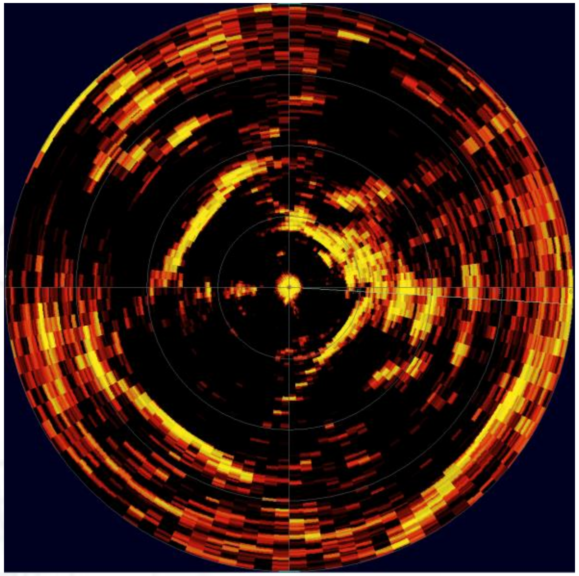


Static sonar localization

- Sonar data when the robot is static

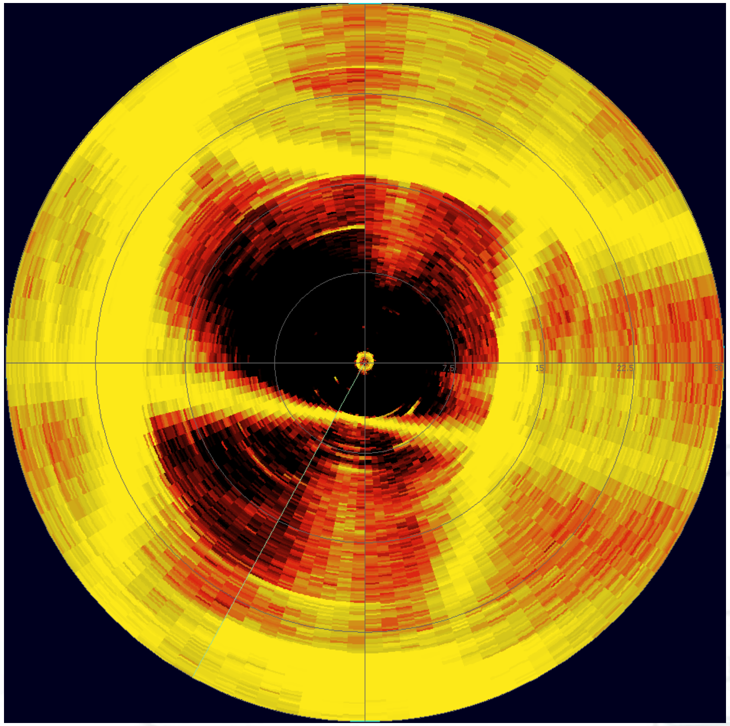


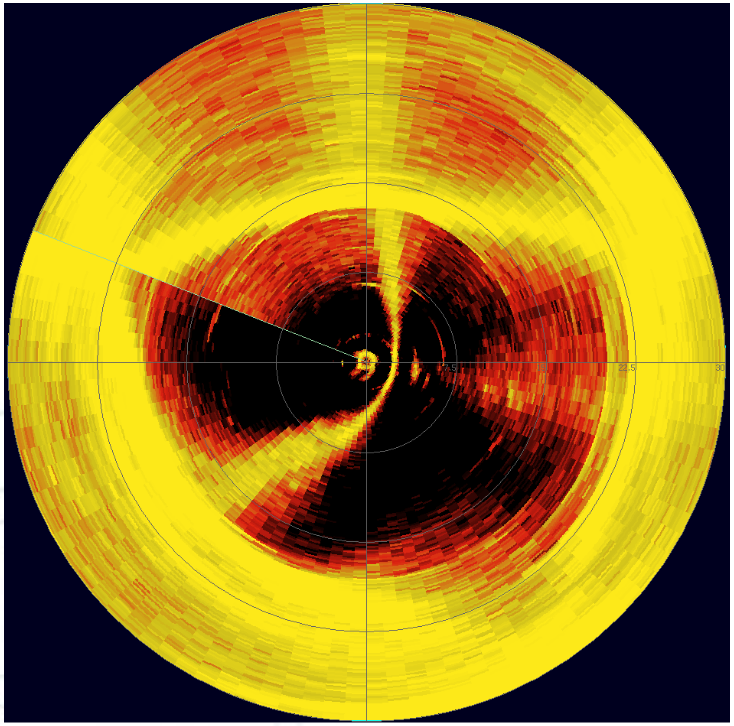




Static sonar localization

- Sonar data when the robot is moving

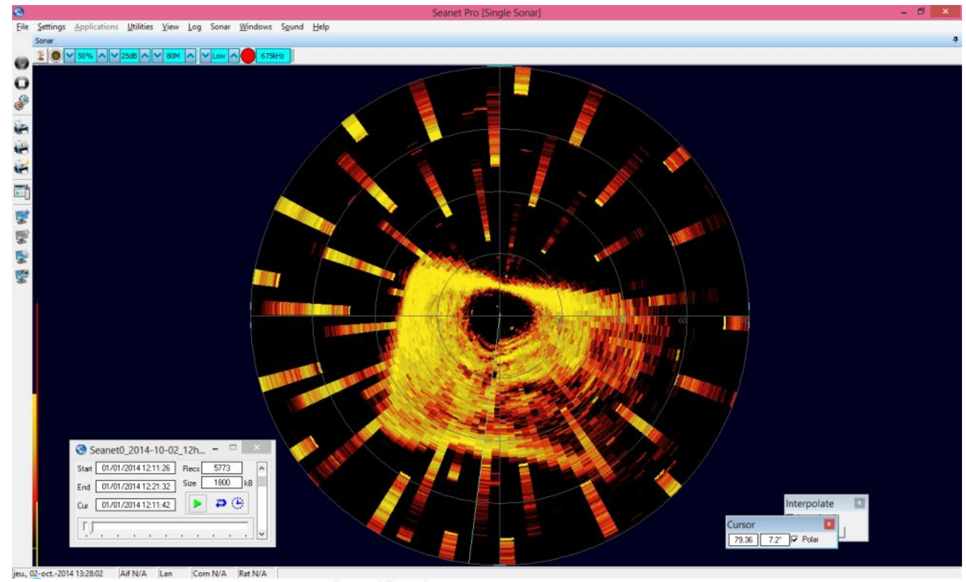


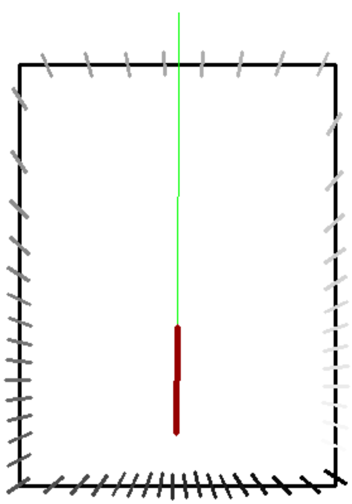


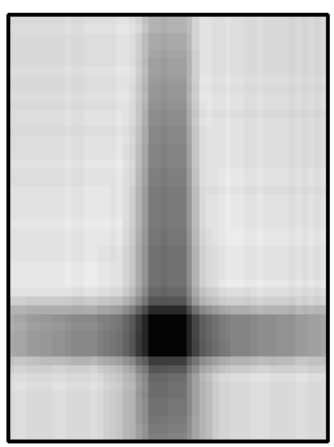
Static sonar localization

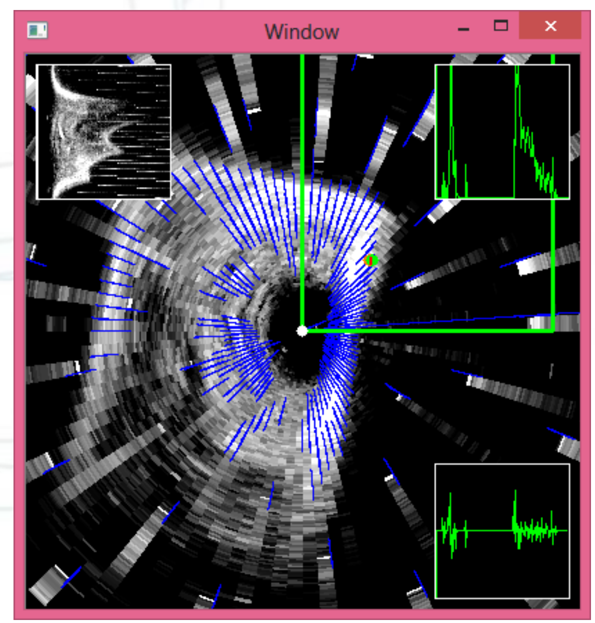
ENSTA Bretagne

- Context : map of the environment known but outliers expected, good compass and a sonar available
- First problem : where are the walls on the sonar image?
- Second problem : where is the robot w.r.t the walls?

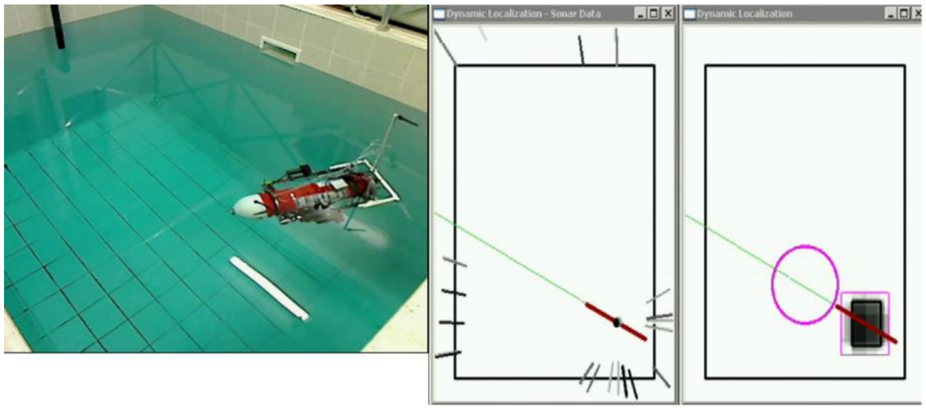


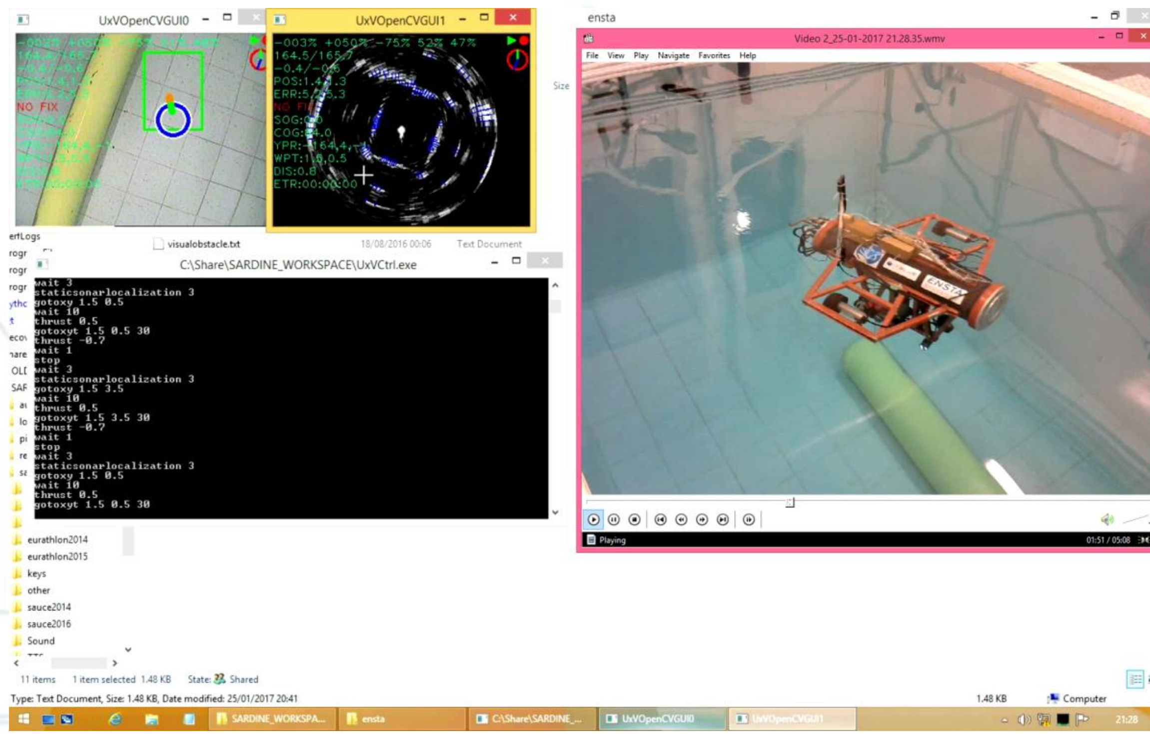






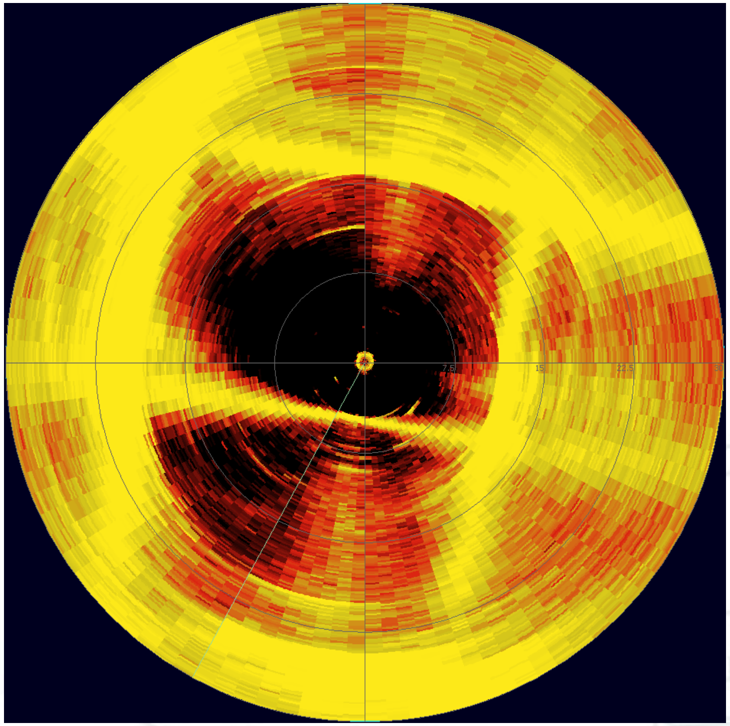
Dead-reckoning+static sonar localization

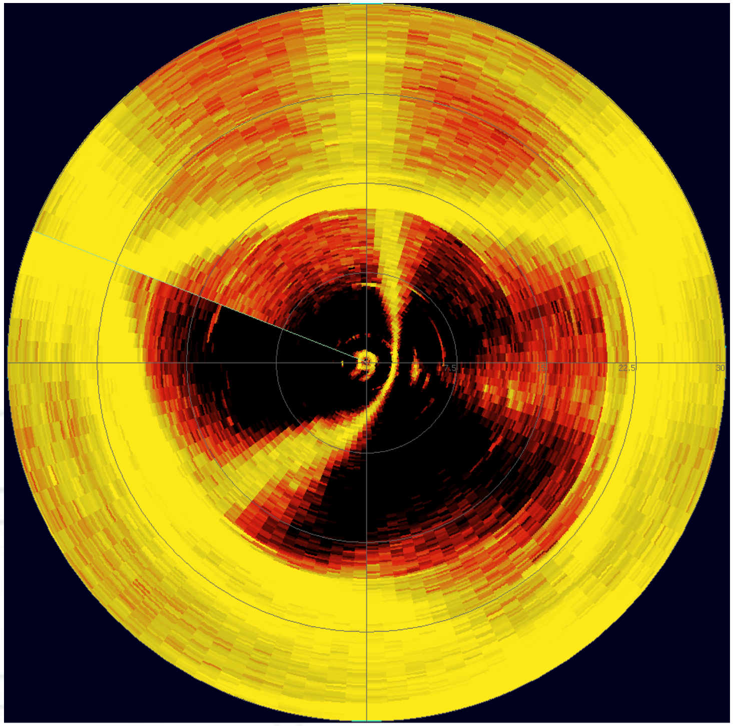




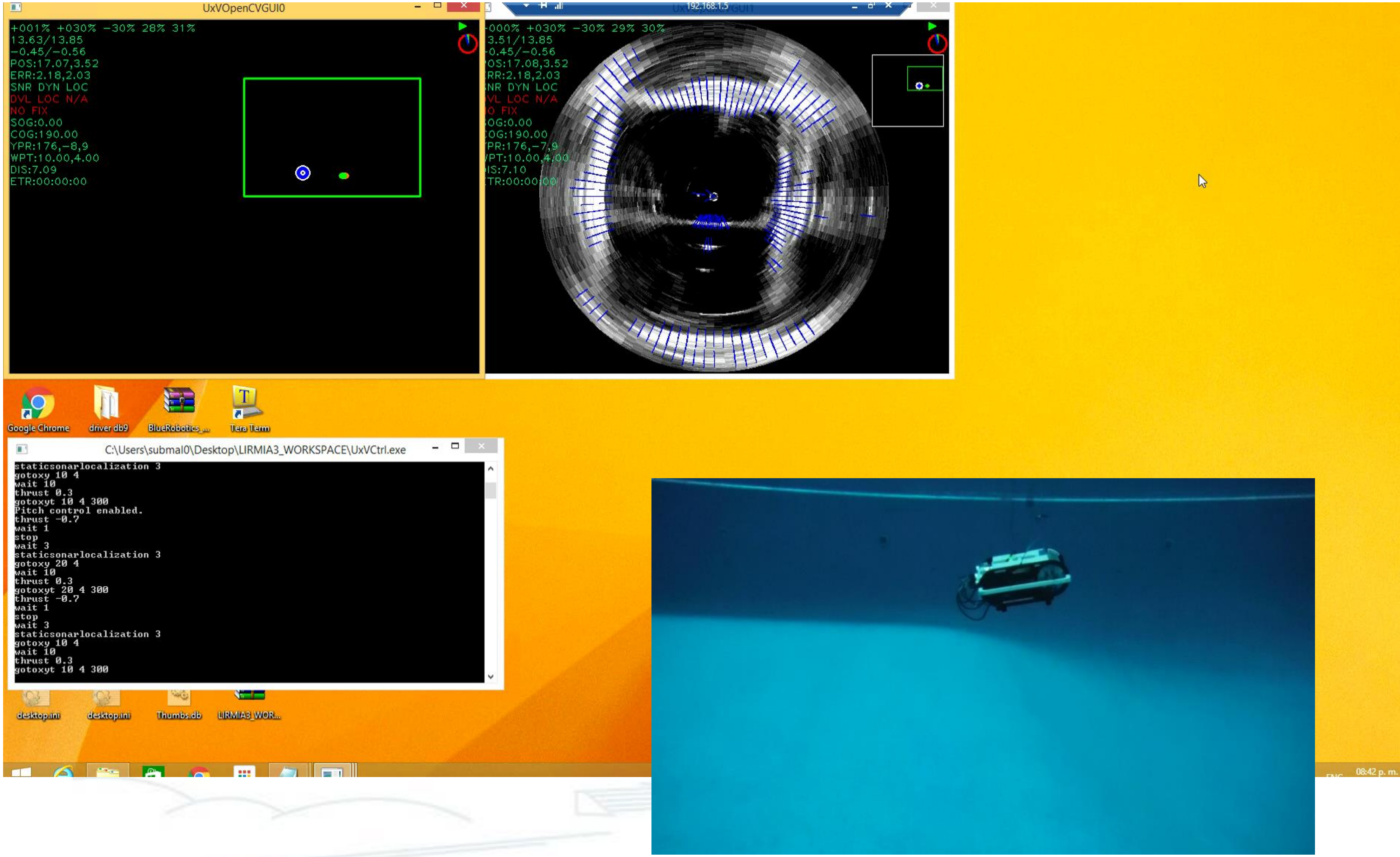
Dynamic sonar localization

- Sonar data when the robot is moving
  - We can use the state equations used for the dead-reckoning to correct the sonar data





Dynamic sonar localization



Future work

Future work

- Dead-reckoning
  - Loops detection
  - Process and fuse with raw IMU data using interval analysis
- Sonar localization
  - Estimate the heading using sonar data
  - Better classify the different types of outliers
  - Use the 3D of the sonar data