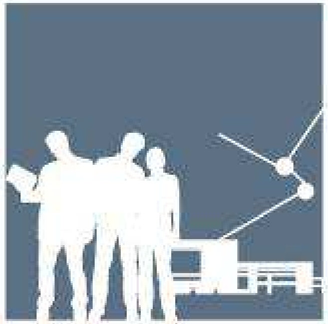


An experimental validation of a robust controller on the VAIMOS autonomous sailboat

Fabrice LE BARS

Outline

- Introduction
- VAIMOS, an autonomous sailboat for oceanography
- Autonomy / control
- Theoretical validation of the controller
- HIL simulation
- Real tests

Introduction

Introduction

- Validation process of a sailboat controller
  - Theoretical validation using interval analysis and Lyapunov methods
  - HIL (Hardware In the Loop) simulator
  - Real experiments in Brest harbor and between Brest and Douarnenez (Brittany, France)

VAIMOS, an autonomous sailboat for oceanography

VAIMOS, an autonomous sailboat for oceanography

- VAIMOS = Voilier Autonome Instrumenté pour Mesures Océanographiques de Surface
  - Collaboration between Ifremer (mechanics and electronics) / ENSTA Bretagne (automatics and embedded computer science)

VAIMOS, an autonomous sailboat for oceanography

- Purpose
  - Oceanographic measurements of various parameters near the water surface and at a depth of about one meter (temperature, salinity, chlorophyll, turbidity…)
  - Assist and / or replace oceanographic boats, fixed or floating buoys currently used

VAIMOS, an autonomous sailboat for oceanography

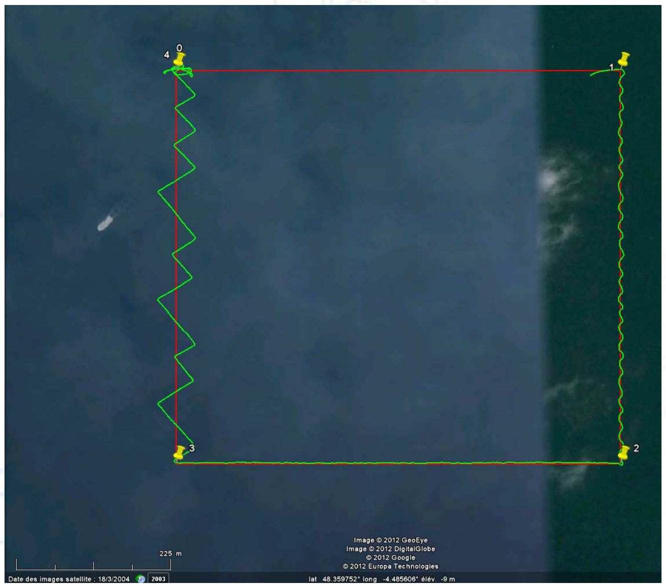
- Advantages
  - Accuracy (vs floating buoys) and easiness of setup (vs fixed buoys)
  - Low power consumption, can be made energetically autonomous for several months
  - Big payload
  - Cheap (about 20000€, probe excluded)

VAIMOS, an autonomous sailboat for oceanography

Autonomy / control

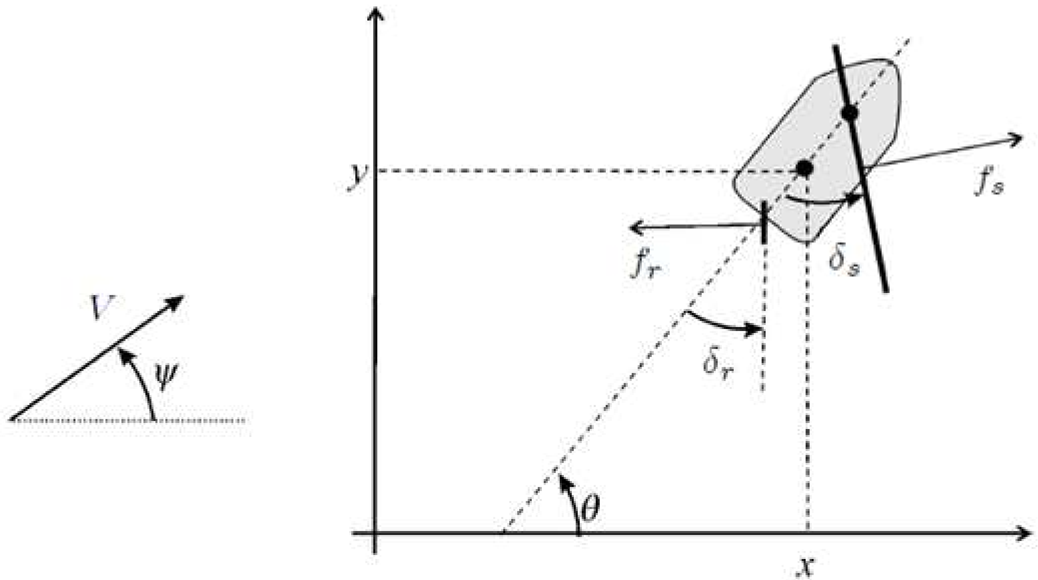
Autonomy / control

- Purpose
  - Cover autonomously an area as accurately as possible
  - Guarantee that the robot always stays in a predefined row of 25 m width, despite maneuvers inherent to course changes, tacks…
  - In this way, the sailboat becomes as accurate as a motorboat



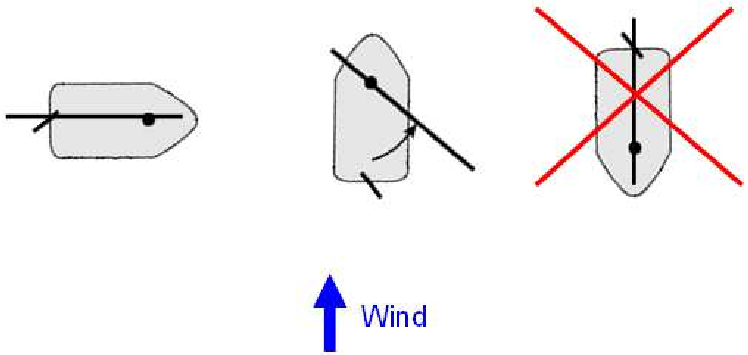
Autonomy / control

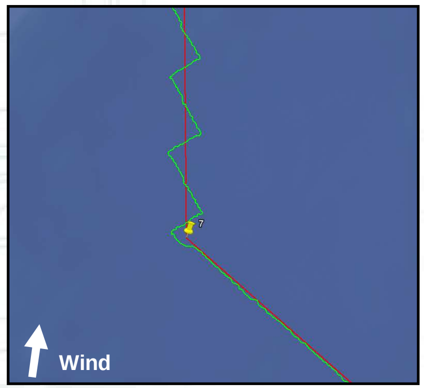
- Particularities



Autonomy / control

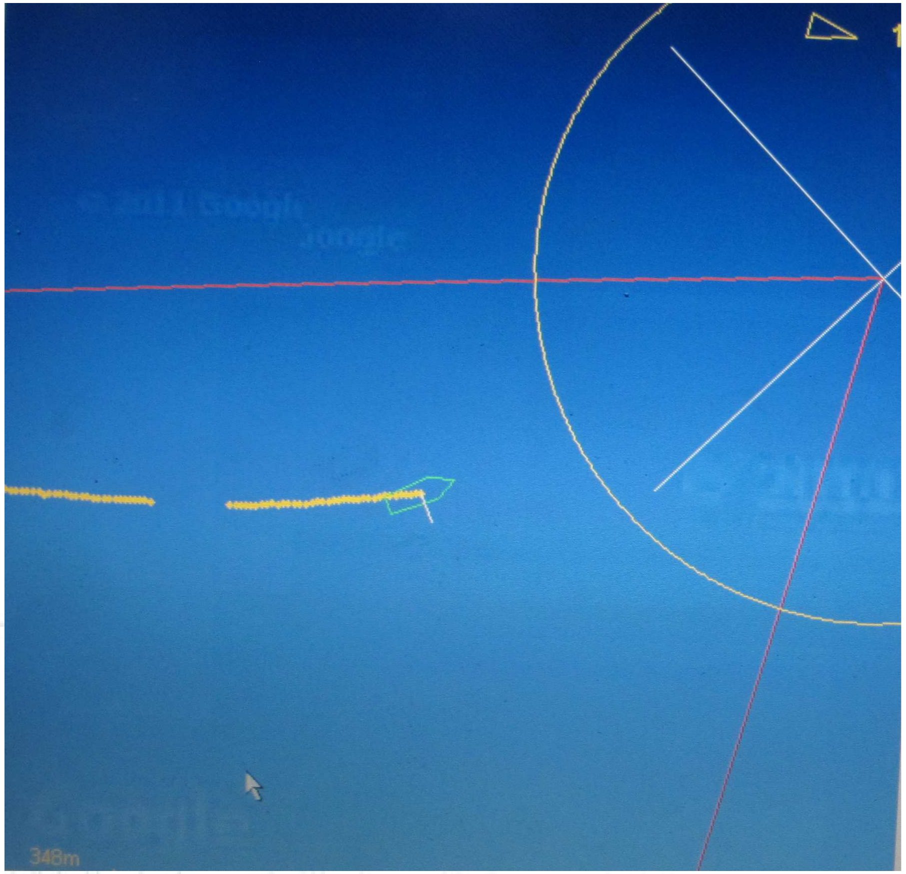
- Particularities
  - Inputs : sail max angle and rudder angle
  - Existence of headings difficult to follow depending on wind orientation
  - Need of 2 types of different strategies : nominal route or tack
  - Tack : regulation around the wind angle +or- 45 deg (clause hauled angle)





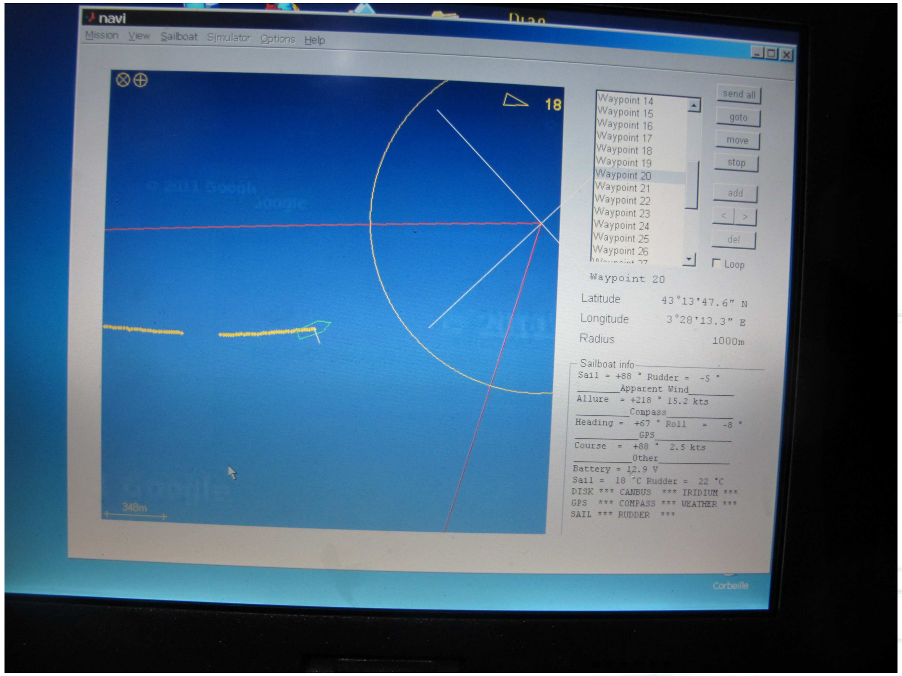
Autonomy / control

- From waypoints following to line following
  - Primitive heading control loop
  - Existing approaches : basic waypoint following
    - The robot follows a heading in direction of its waypoint
    - Waypoint reached when in a predefined radius
    - Problem : nothing prevent the drift between waypoints (because of currents...)



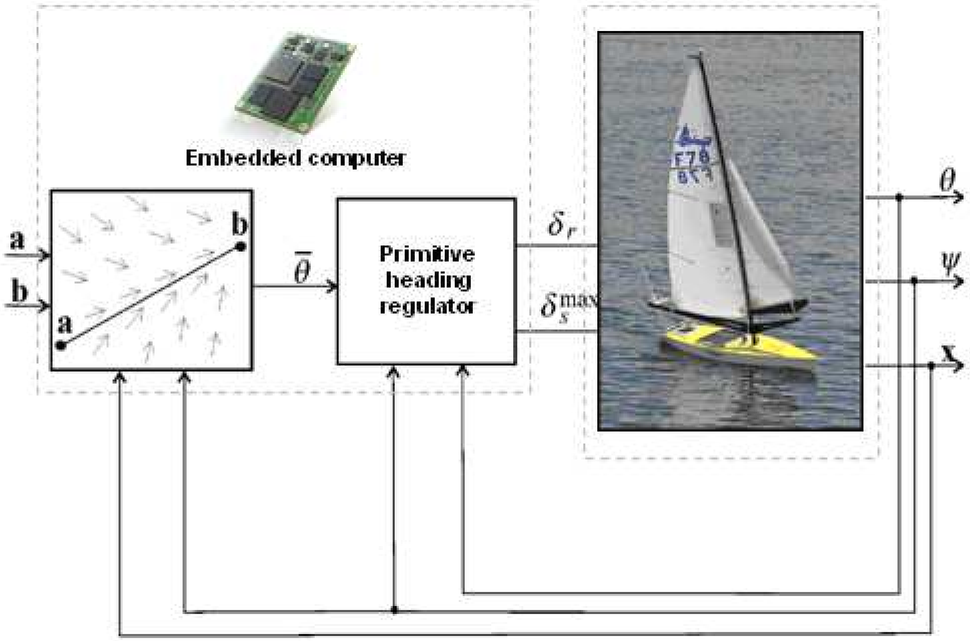
Autonomy / control

- EX : IBOAT of ISAE (Toulouse), 1st autonomous sailboat robot to cover 100 km (June 2011)



Autonomy / control

- Line following



Autonomy / control

- Line following
  - Primitive controller stage for heading control
    - Rudder control

$$\delta_r = \begin{cases} \delta_r^{\max}.\sin\left(\theta - \bar{\theta}\right) & \text{if } \cos\left(\theta - \bar{\theta}\right) \geq 0 \\ \delta_r^{\max}.\text{sign}\left(\sin\left(\theta - \bar{\theta}\right)\right) & \text{otherwise} \end{cases}$$

    - Sail control

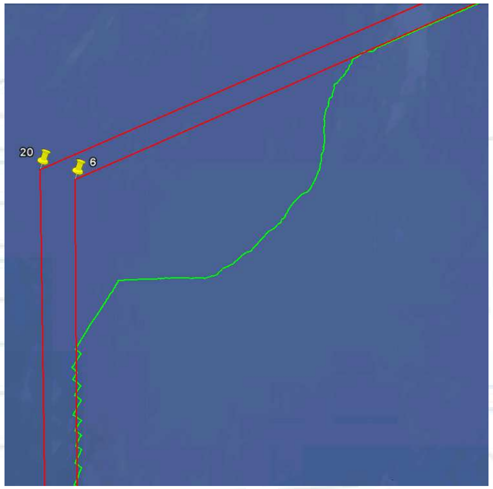
$$\delta_s^{\max} = \frac{\pi}{2}.\left(\frac{\cos(\psi - \bar{\theta}) + 1}{2}\right)$$

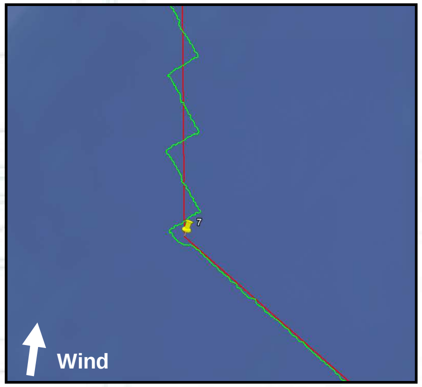
  - Supervisor decides between 2 modes : nominal route or tack => always send feasible headings to primitive controller
  - Navigation manager sends lines to supervisor and validates lines
    - Validation condition

$$\langle \mathbf{b}_j - \mathbf{a}_j, \mathbf{m} - \mathbf{b}_j \rangle \geq 0$$

Autonomy / control

- Line following
  - In nominal route mode : heading is the line made by the 2 current waypoints with an attractiveness angle to the line depending on the distance to the line
  - In tack mode : heading is around the wind orientation +or- 45° (clause hauled angle) => oscillations around the wind angle, oscillations amplitude being the row width





Theoretical validation of the controller

Theoretical validation of the controller

- Representation by differential inclusions and application of Lyapunov analysis methods to transform the stability problem in a set inversion problem
- Demonstration that for all possible perturbations :
  - There exist a subset of the state space where the system cannot escape when it enters in it
  - If the system is outside this subset, it will not stay outside forever

Theoretical validation of the controller

- Even if we can validate theoretically the robustness of the controller (i.e. the robot will stay in a row around its target line), additional methods must be used to check hypothesis (state equations, bounds on sensors errors, coefficients…)

HIL simulation

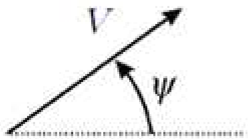
HIL simulation

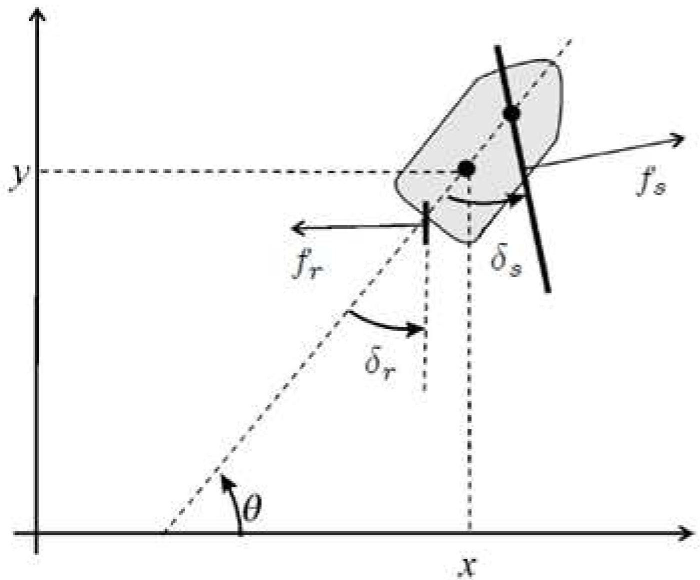
- Existing simulators
  - Use polar speed diagram of the sailboat to determine its movement
  - Use several predefined scenarios and therefore can difficultly be used to emphasize possible singularities

  => Use of state equations for our controller validation purpose

HIL simulation

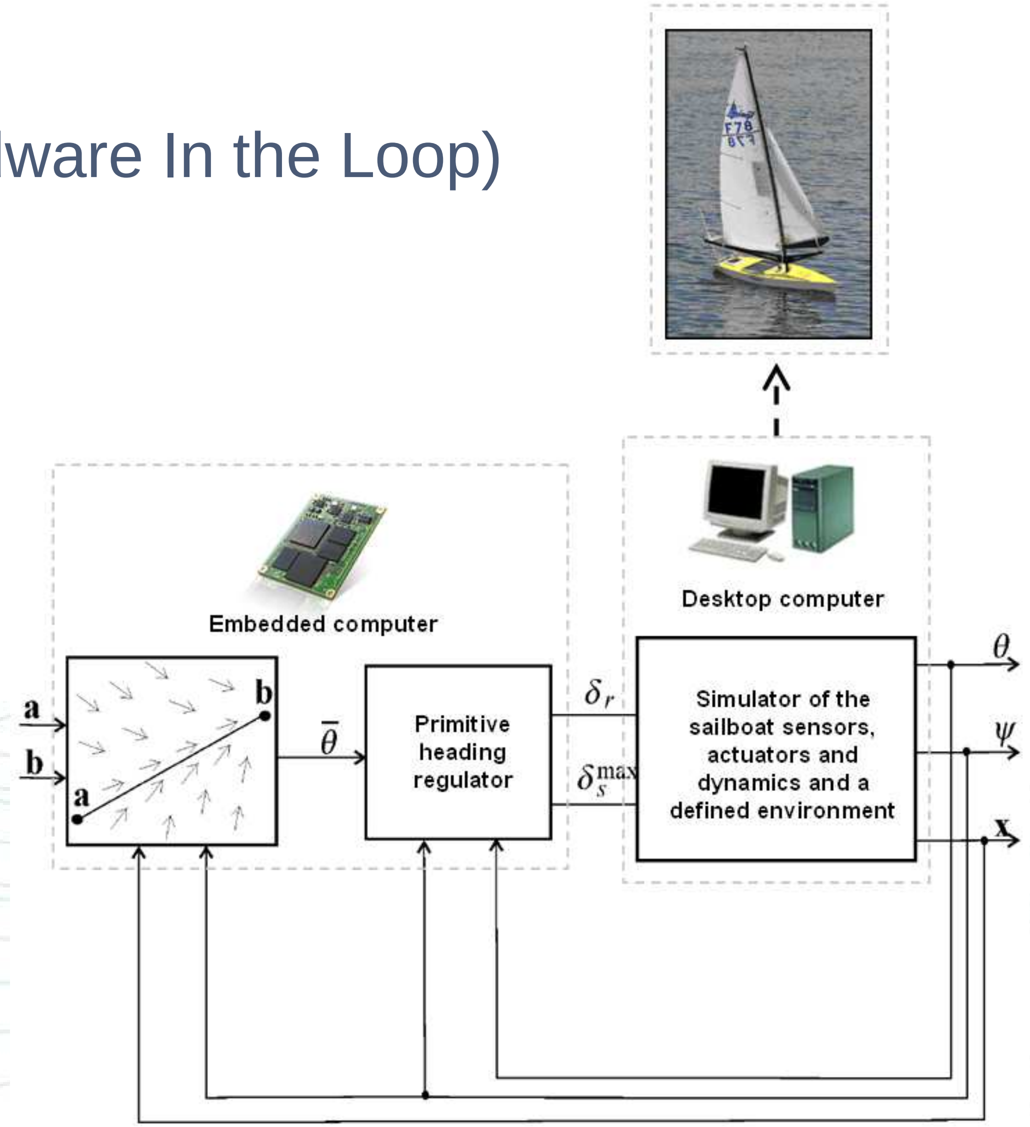
$$\left\{\begin{array}{lll}
\sigma & = & \cos(\theta-\psi)+\cos(\delta_{s\,\max}) \\
\delta_s & = & \left\{\begin{array}{ll} \pi-\theta+\psi & \text{if } \sigma<0 \\ \delta_{s\,\max}\text{sign}(\sin(\theta-\psi)) & \text{otherwise} \end{array}\right. \\
f_r & = & \alpha_r v \sin(\delta_r) \\
f_s & = & \alpha_s V \sin(\theta+\delta_s-\psi) \\
\dot{x} & = & v\cos(\theta)+\beta V\cos(\psi)+V_c\cos(\psi_c) \\
\dot{y} & = & v\sin(\theta)+\beta V\sin(\psi)+V_c\sin(\psi_c) \\
\dot{\theta} & = & \omega \\
\dot{\omega} & = & \frac{(l-r_s\cos(\delta_s))f_s-r_r\cos(\delta_r)f_r-\alpha_\theta\omega+\alpha_w h_w}{J_z} \\
\dot{v} & = & \frac{\sin(\delta_s)f_s-\sin(\delta_r)f_r-\alpha_f v^2}{m} \\
\ddot{\varphi} & = & \frac{-\alpha_\varphi\dot{\varphi}+f_s h_s\cos(\delta_v)\cos(\varphi)-m_{eq}l_{eq}g\sin(\varphi)}{J_x} \\
\dot{\varphi} & = & \dot{\varphi}
\end{array}\right.$$



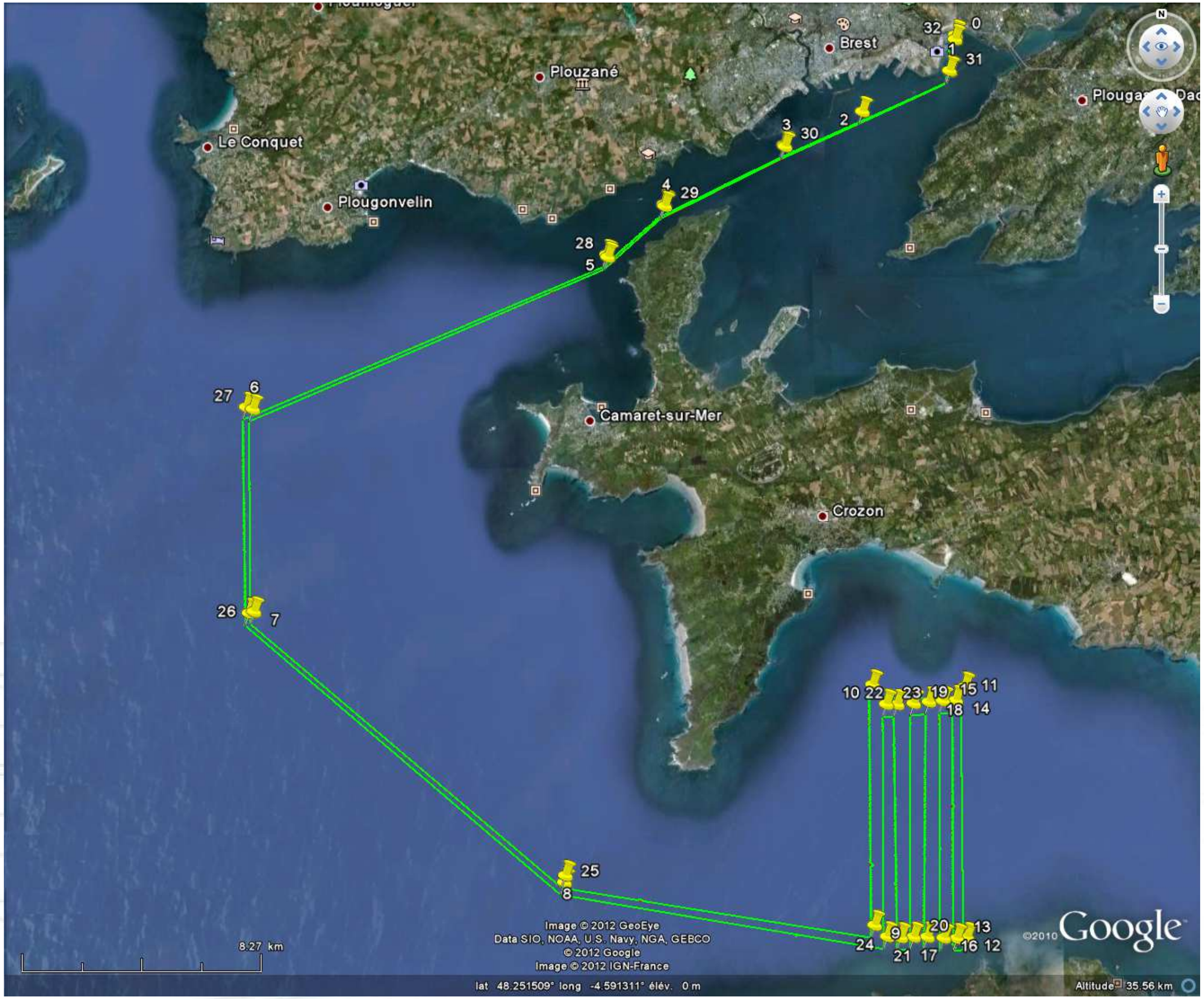


HIL simulation

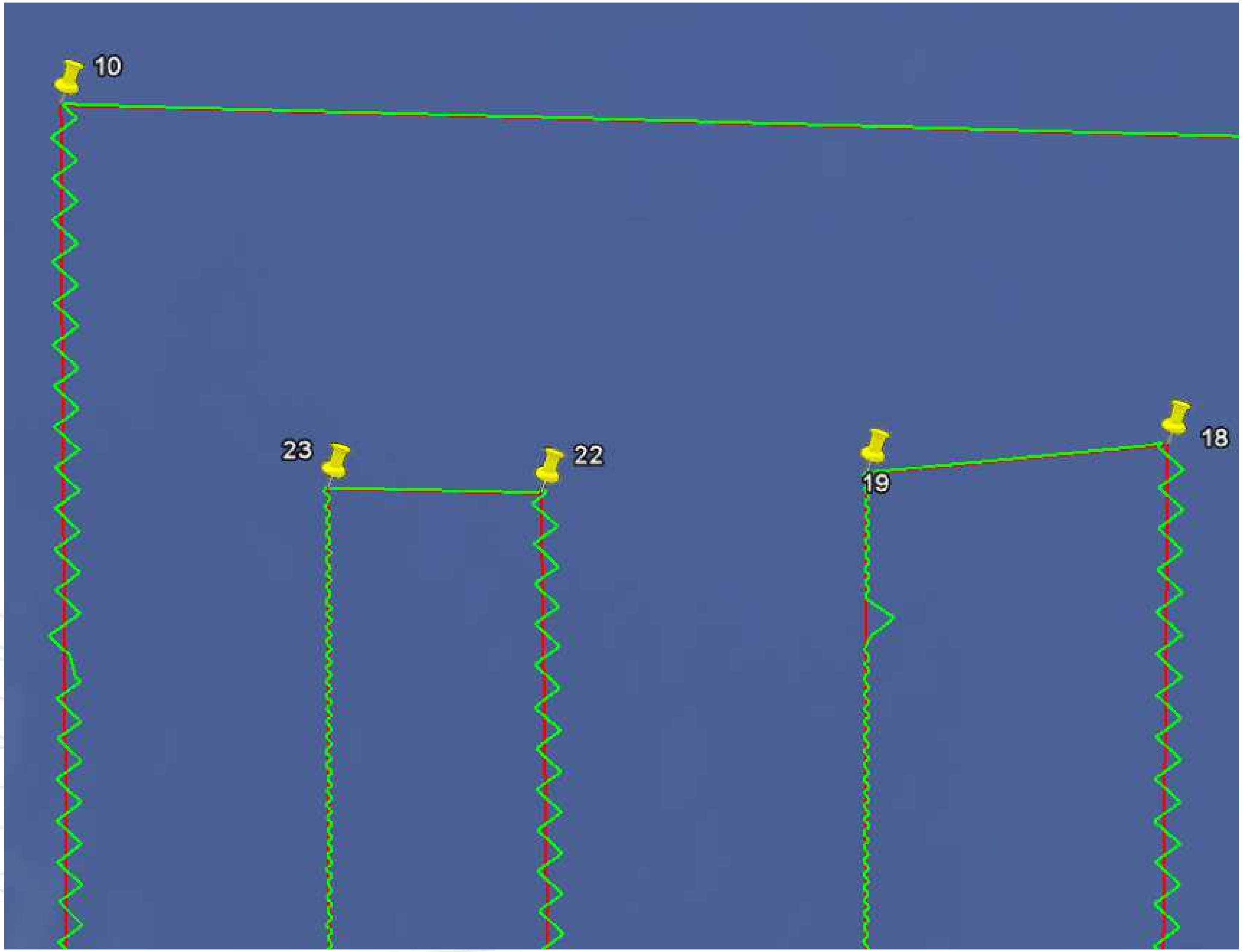
- HIL (Hardware In the Loop) simulator



HIL simulation



HIL simulation



HIL simulation

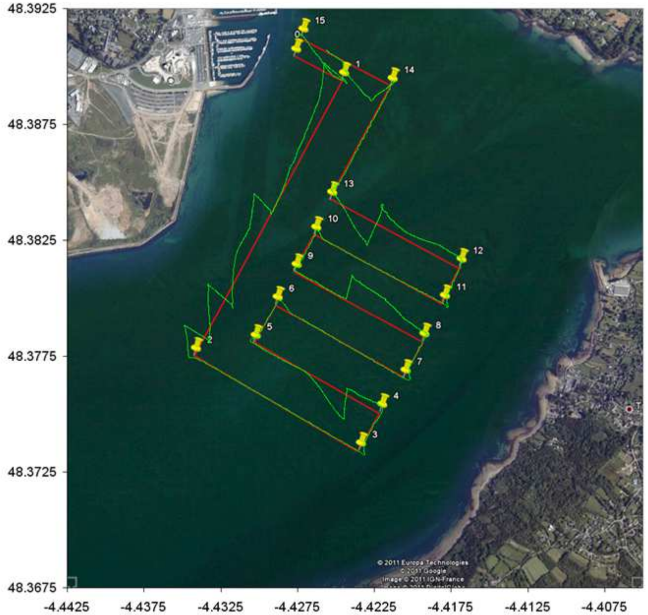

Copyright Ifremer


F78
Copyright Ifremer

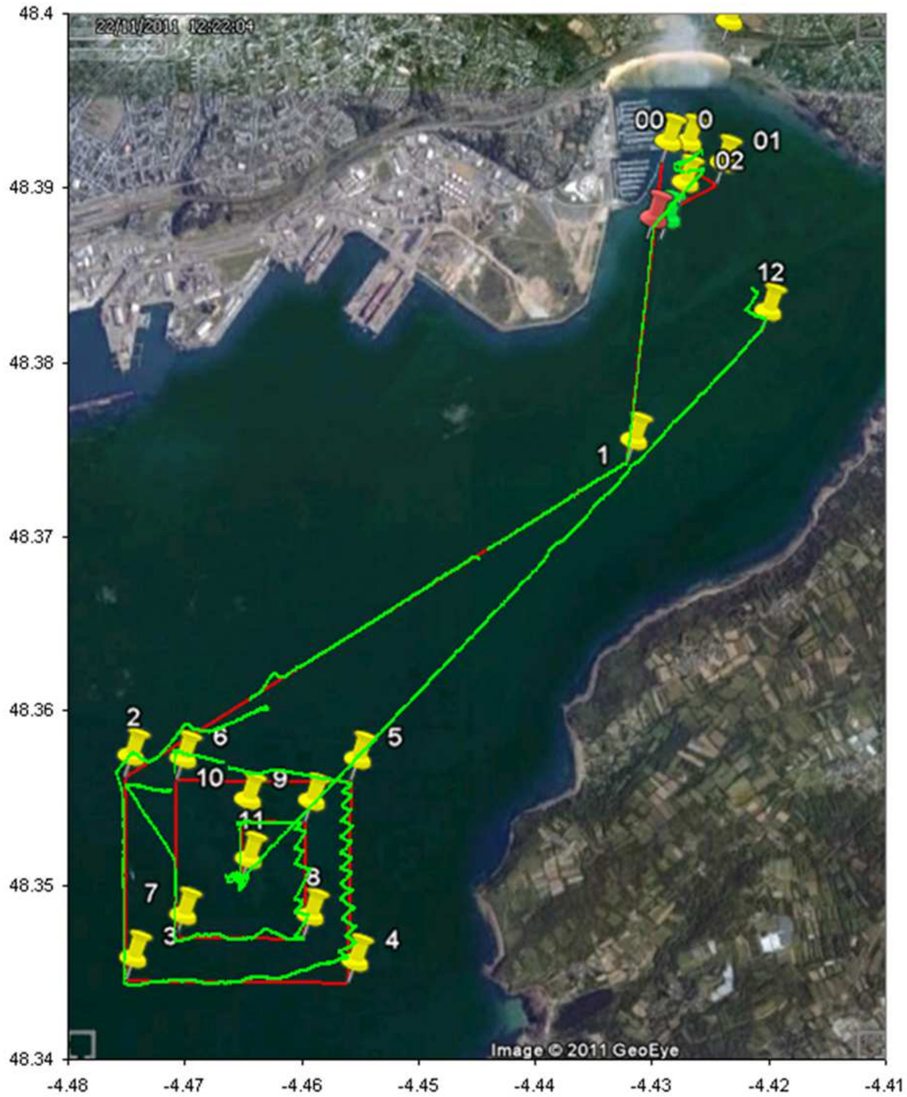

15
0
1
14
13
10
12
9
11
6
8
5
2
7
4
3
Copyright Ifremer

Real tests

Real tests

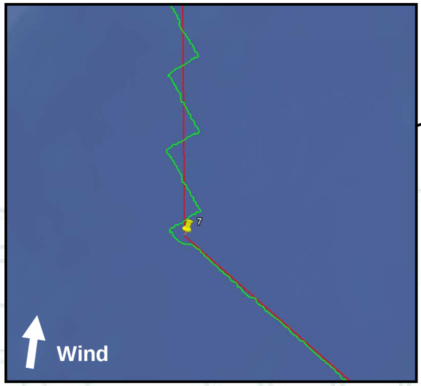


48.3925
48.3875
48.3825
48.3775
48.3725
48.3675
-4.4425
-4.4375
-4.4325
-4.4275
-4.4225
-4.4175
-4.4125
-4.4075



22/11/2011 12:22:04
00
0
01
02
12
1
2
6
5
10
9
11
7
3
8
4
Image © 2011 GeoEye
48.4
48.39
48.38
48.37
48.36
48.35
48.34
-4.48
-4.47
-4.46
-4.45
-4.44
-4.43
-4.42
-4.41

Real tests

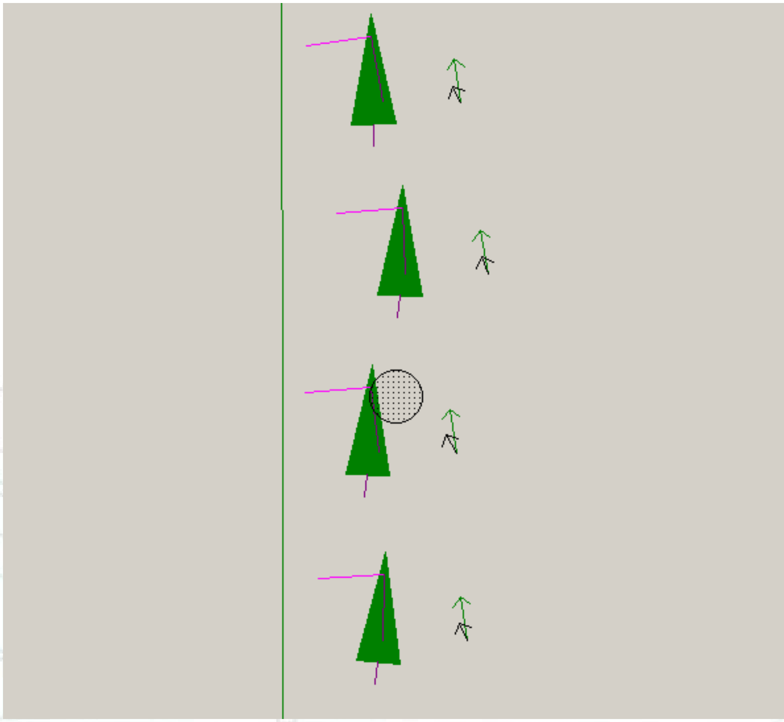


Details of a change between tack and nominal route, decided by the robot

Desired Brest-Douarnenez trajectory (red lines made by yellow waypoints) and effective trajectory (green)

Real tests

- Analysis of data from the experiments using a dashdoard (during the tests and after)



ENSTA Bretagne

Conclusion

Conclusion

- Interval analysis can be used to theoretically validate robot control algorithms
- However, in robotics we must use other validation methods such as HIL simulation and real tests to check and correct hypothesis made

Questions?

Towards swarm of marine and submarine robots