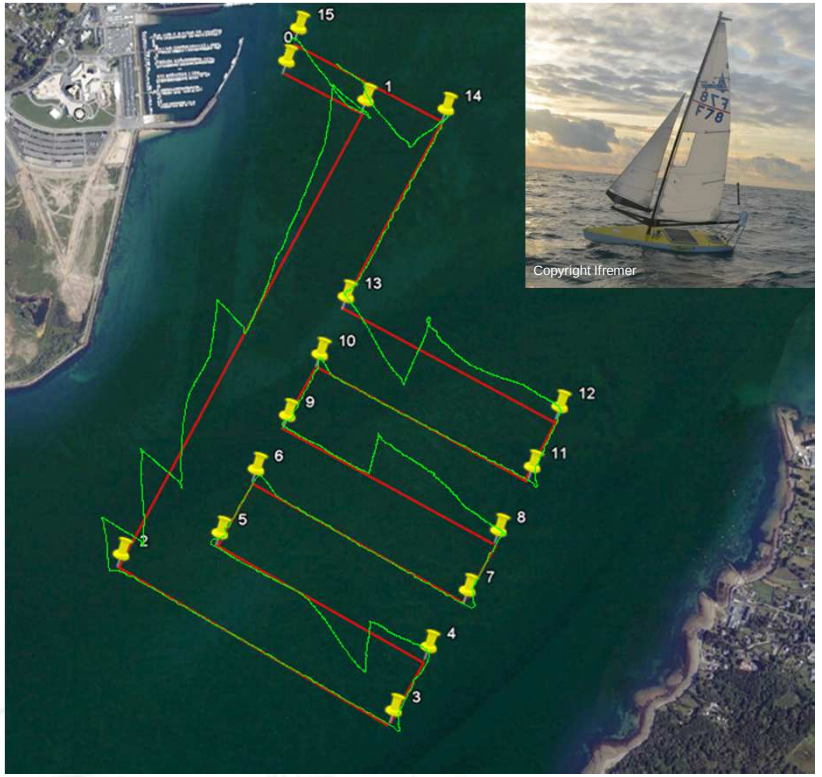
ENSTA Bretagne

Control of an autonomous sailboat : application to the VAIMOS robot

Fabrice LE BARS

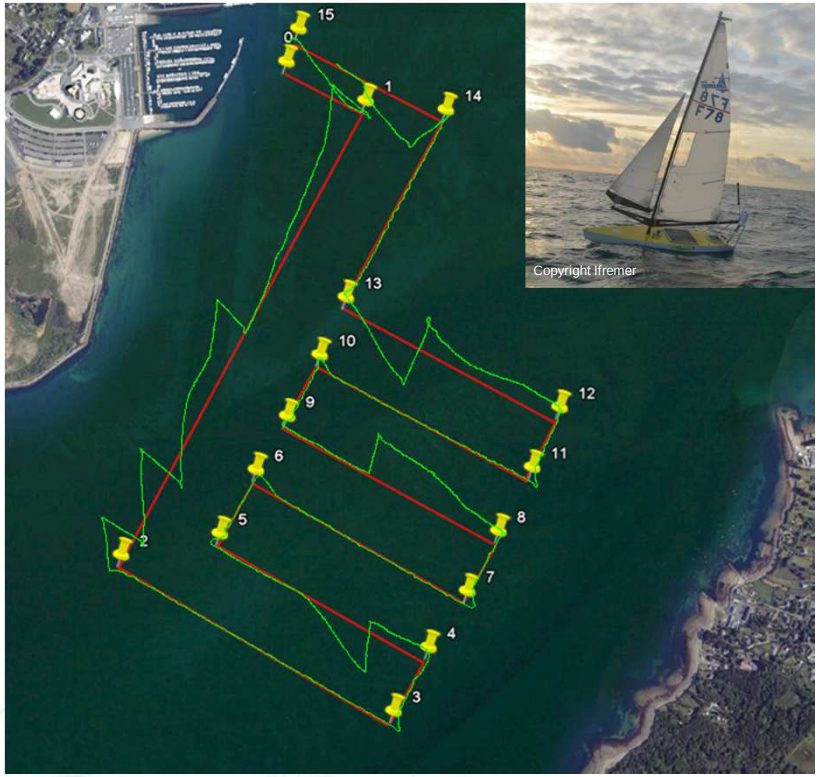


VAIMOS, an autonomous sailboat for oceanography

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)
  - Designed for oceanographic measurements at the sea surface

VAIMOS, an autonomous sailboat for oceanography



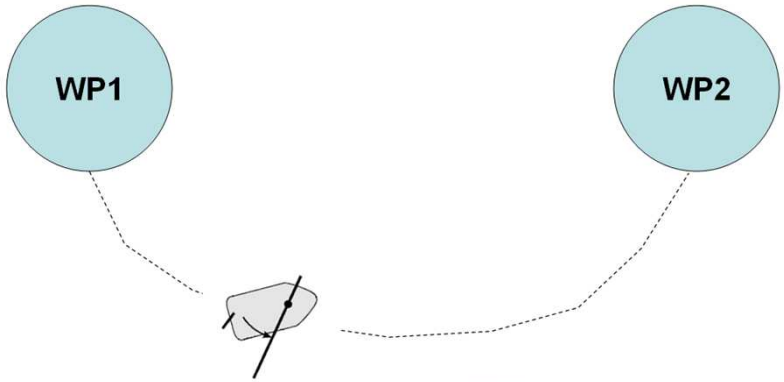
Autonomy / control

Autonomy / control

- Purpose
  - Cover autonomously an area as accurately as possible while the accompanying oceanographic ship has other activities

Autonomy / control

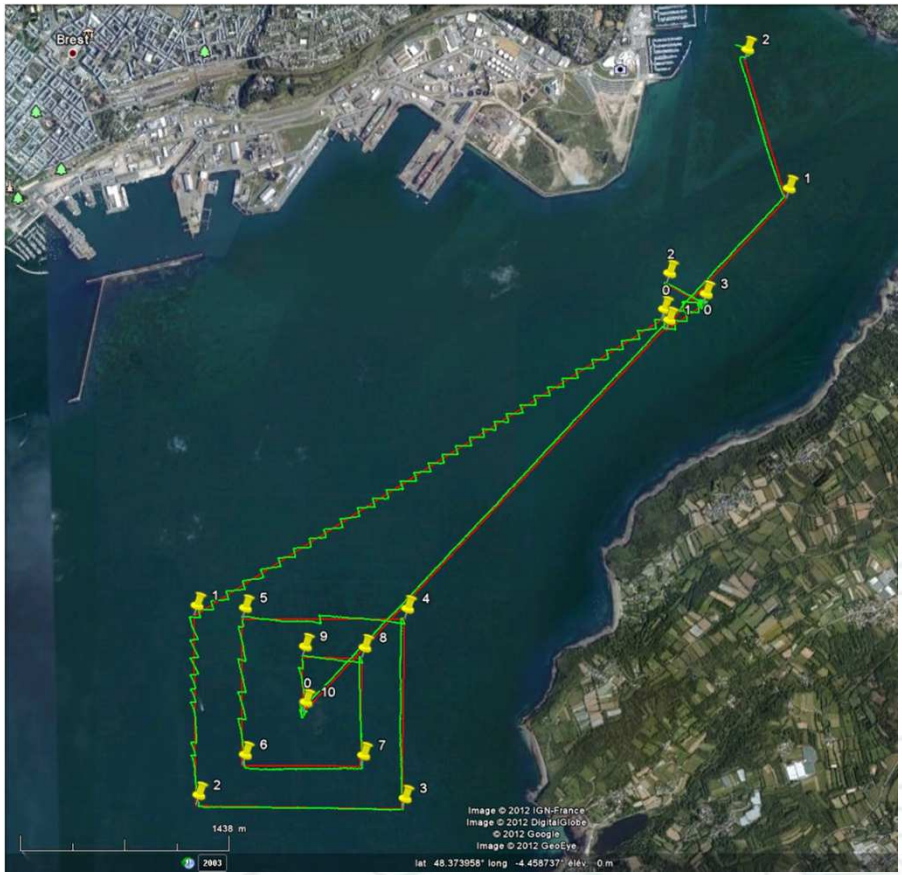
- From waypoints following to line following
  - Primitive heading control loop
  - Existing approaches : basic waypoint following
    - The robot follows a heading in direction of its waypoint
    - Waypoint reached when in a predefined radius
    - Problem : nothing prevent the drift between waypoints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

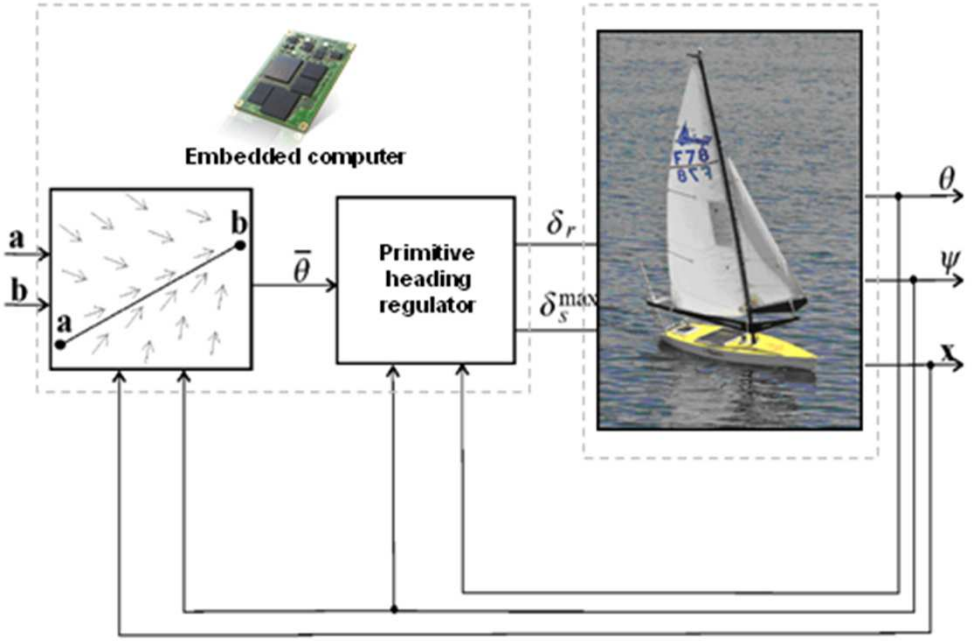


  - => Line following (use the distance to the line)

Autonomy / control

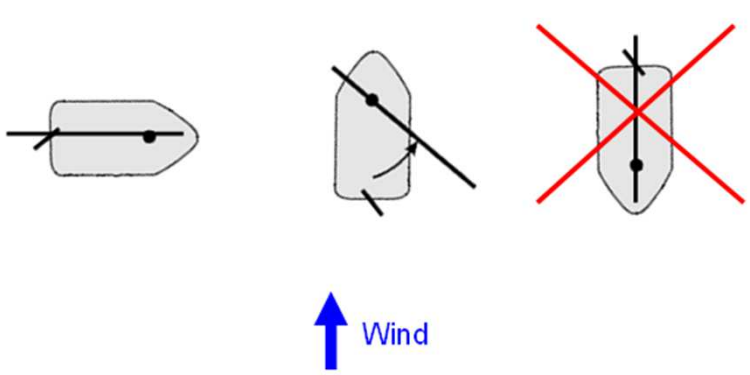
- Line following



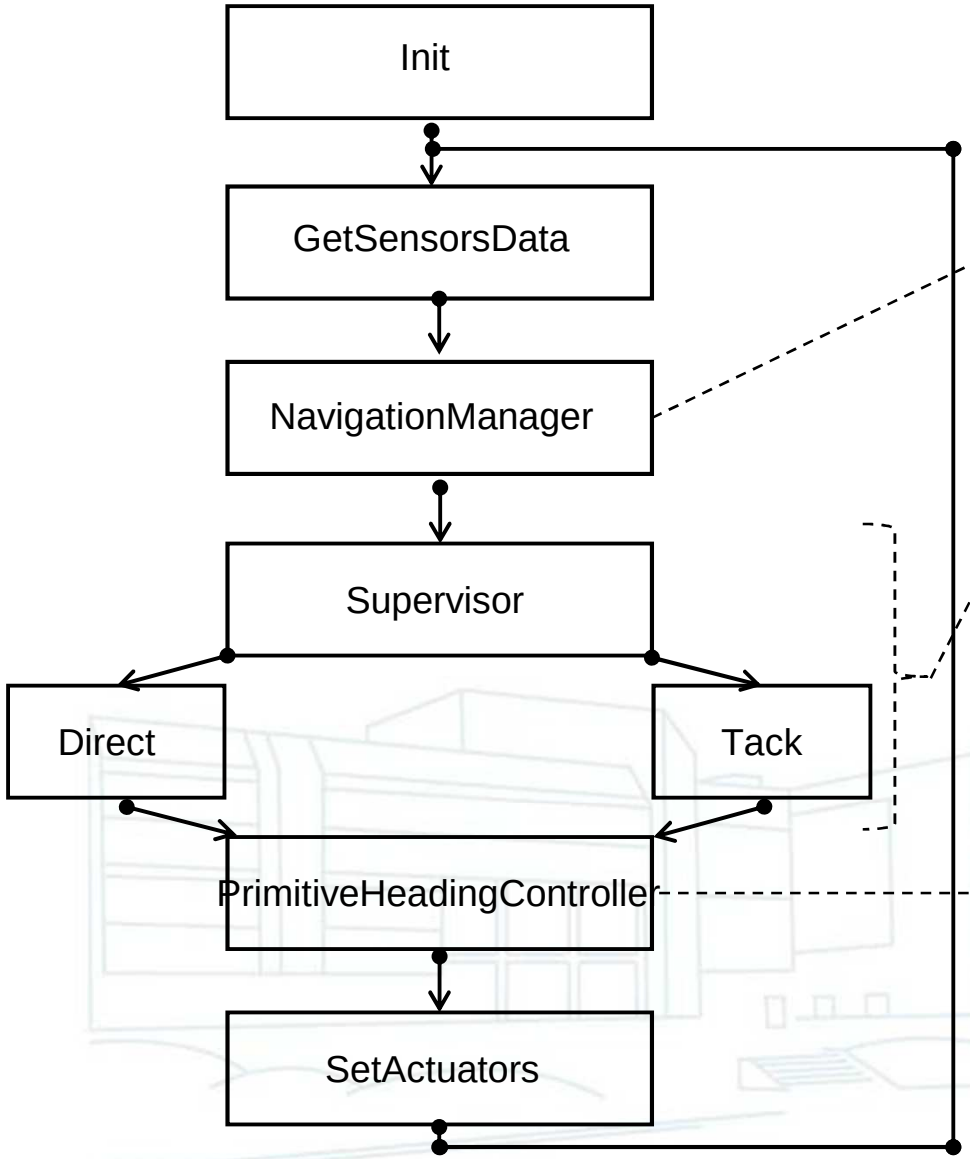


Autonomy / control

- Feasible headings for a sailboat
  - Existence of headings difficult to follow depending on wind orientation
  - Need of 2 types of different strategies : direct route or tack
  - Tack : +or- 45 deg around the wind angle



Autonomy / control



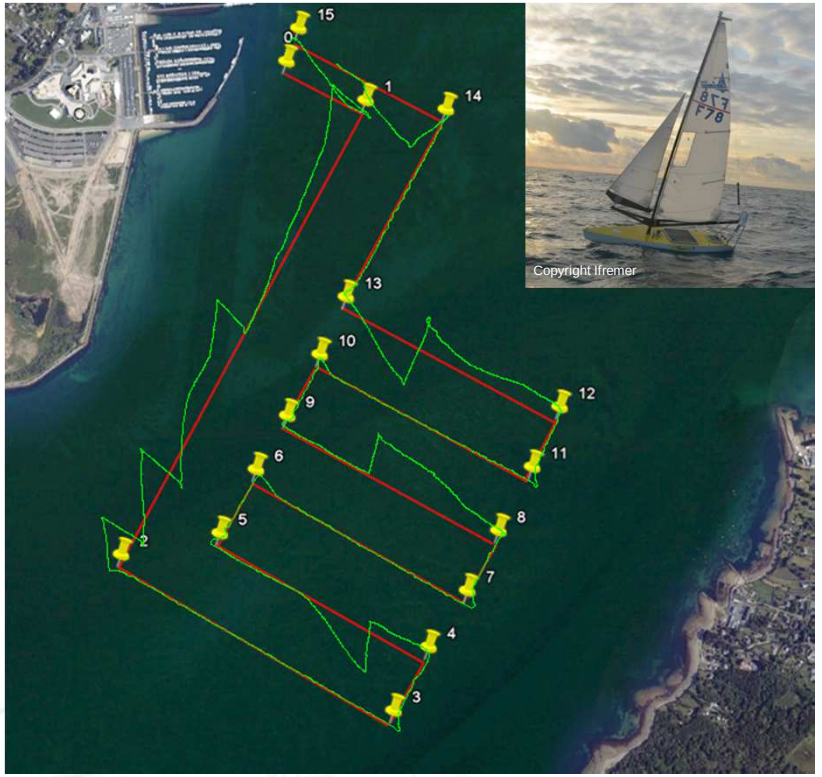
$$\langle \mathbf{b}_j - \mathbf{a}_j, \mathbf{m} - \mathbf{b}_j \rangle \geq 0$$

$$e = \det\left(\frac{\mathbf{b}-\mathbf{a}}{\|\mathbf{b}-\mathbf{a}\|}, \mathbf{m}-\mathbf{a}\right)$$
$$\varphi = \text{atan2}(\mathbf{b}-\mathbf{a})$$
$$\theta^* = \varphi - \frac{2.\gamma_\infty}{\pi}.\text{atan}\left(\frac{e}{r}\right)$$
$$\text{if } \cos(\psi - \theta^*) + \cos\zeta < 0$$
$$\text{or } (|e| < r \text{ and } (\cos(\psi - \varphi) + \cos\zeta < 0))$$
$$\text{then } \bar{\theta} = \pi + \psi - \zeta.\text{sign}(e)$$
$$\text{else } \bar{\theta} = \theta^*$$
$$\text{end}$$

$$\delta_r = \begin{cases} \delta_r^{\max}.\sin(\theta - \bar{\theta}) & \text{if } \cos(\theta - \bar{\theta}) \geq 0 \\ \delta_r^{\max}.\text{sign}(\sin(\theta - \bar{\theta})) & \text{otherwise} \end{cases}$$
$$\delta_s^{\max} = \frac{\pi}{2}.\left(\frac{\cos(\psi - \bar{\theta}) + 1}{2}\right)$$

Autonomy / control

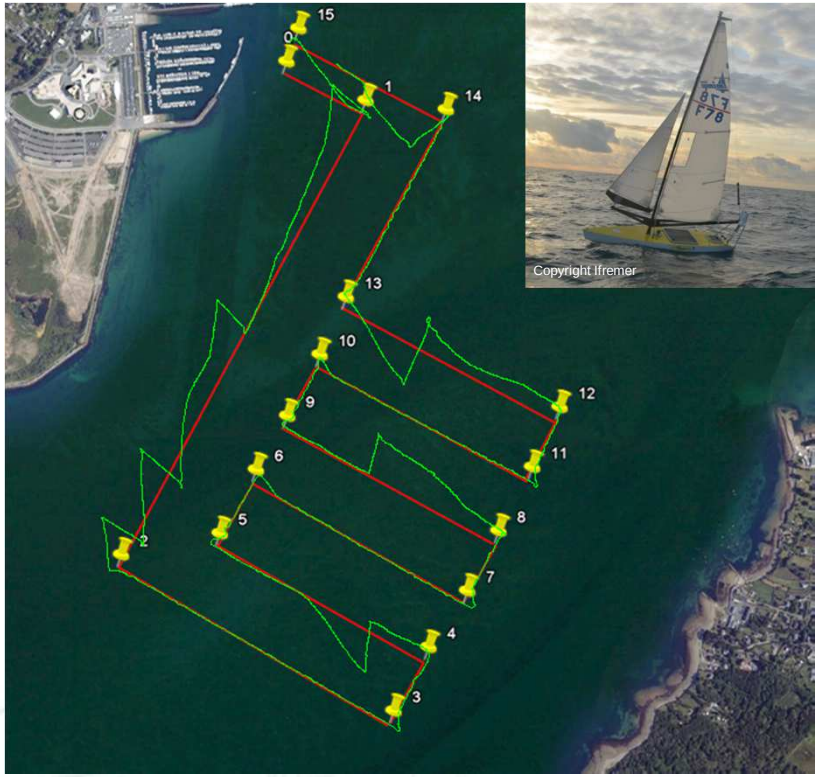
- Example



Theoretical validation of the controller

Theoretical validation of the controller

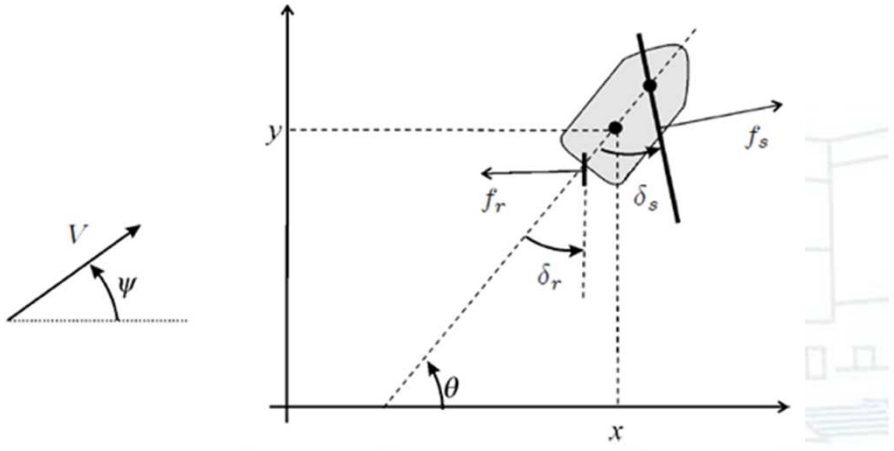
- Representation by differential inclusions and application of Lyapunov analysis methods to transform the stability problem in a set inversion problem
- => Demonstration that the robot will always stay in a strip around its target line and will try to join it if it is outside

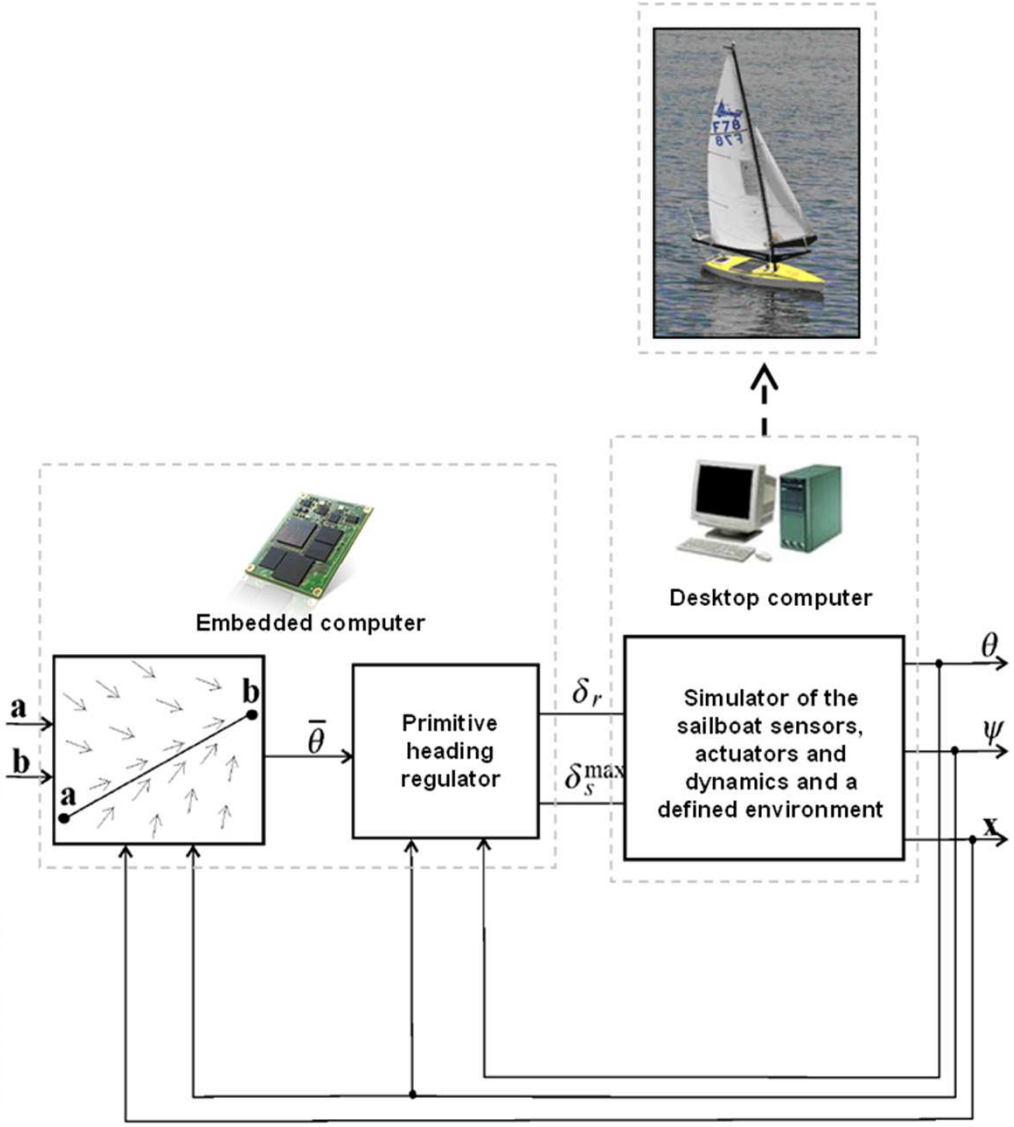


HIL simulation

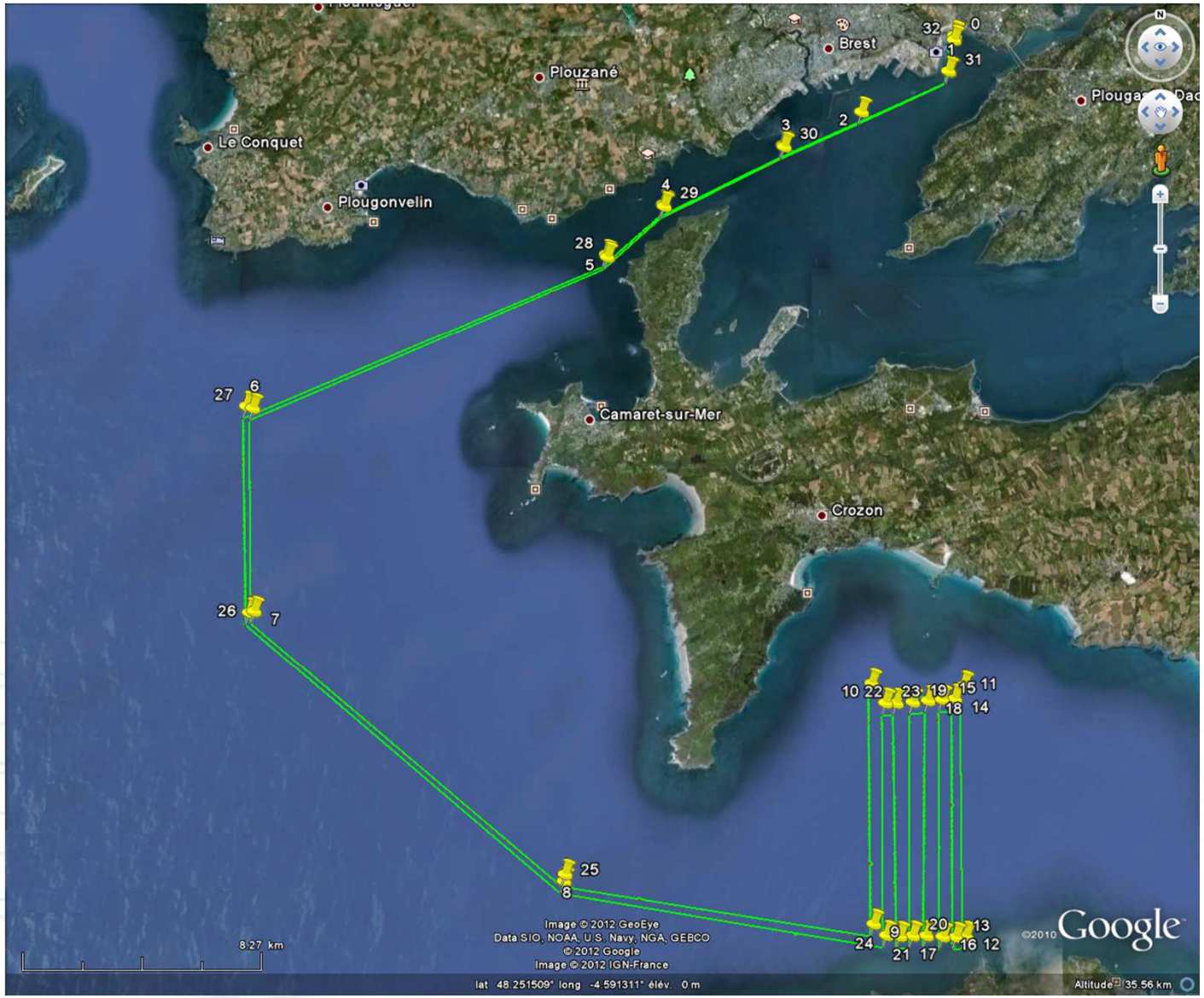
HIL simulation

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

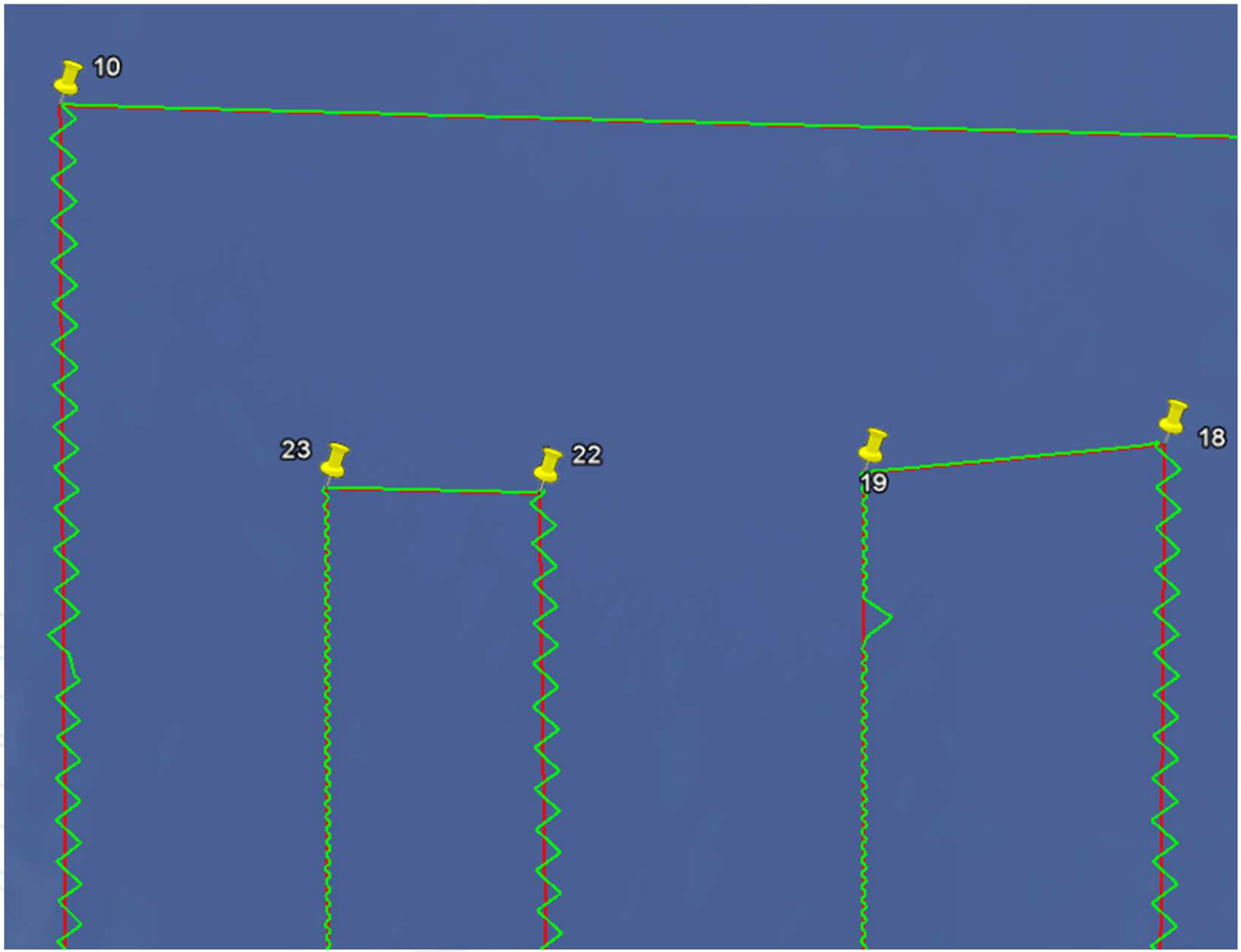




HIL simulation



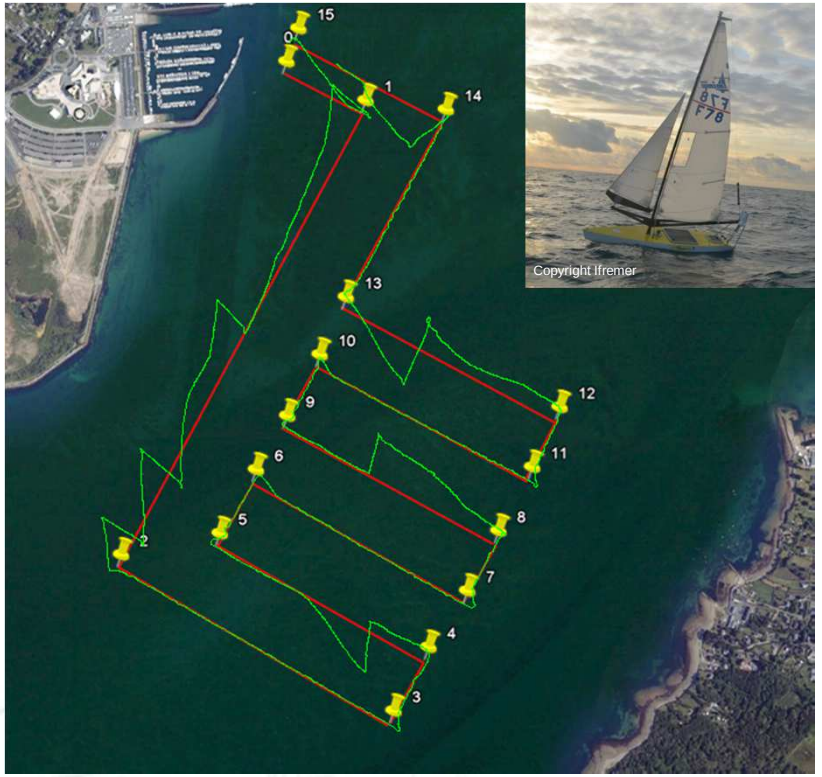
HIL simulation



HIL simulation

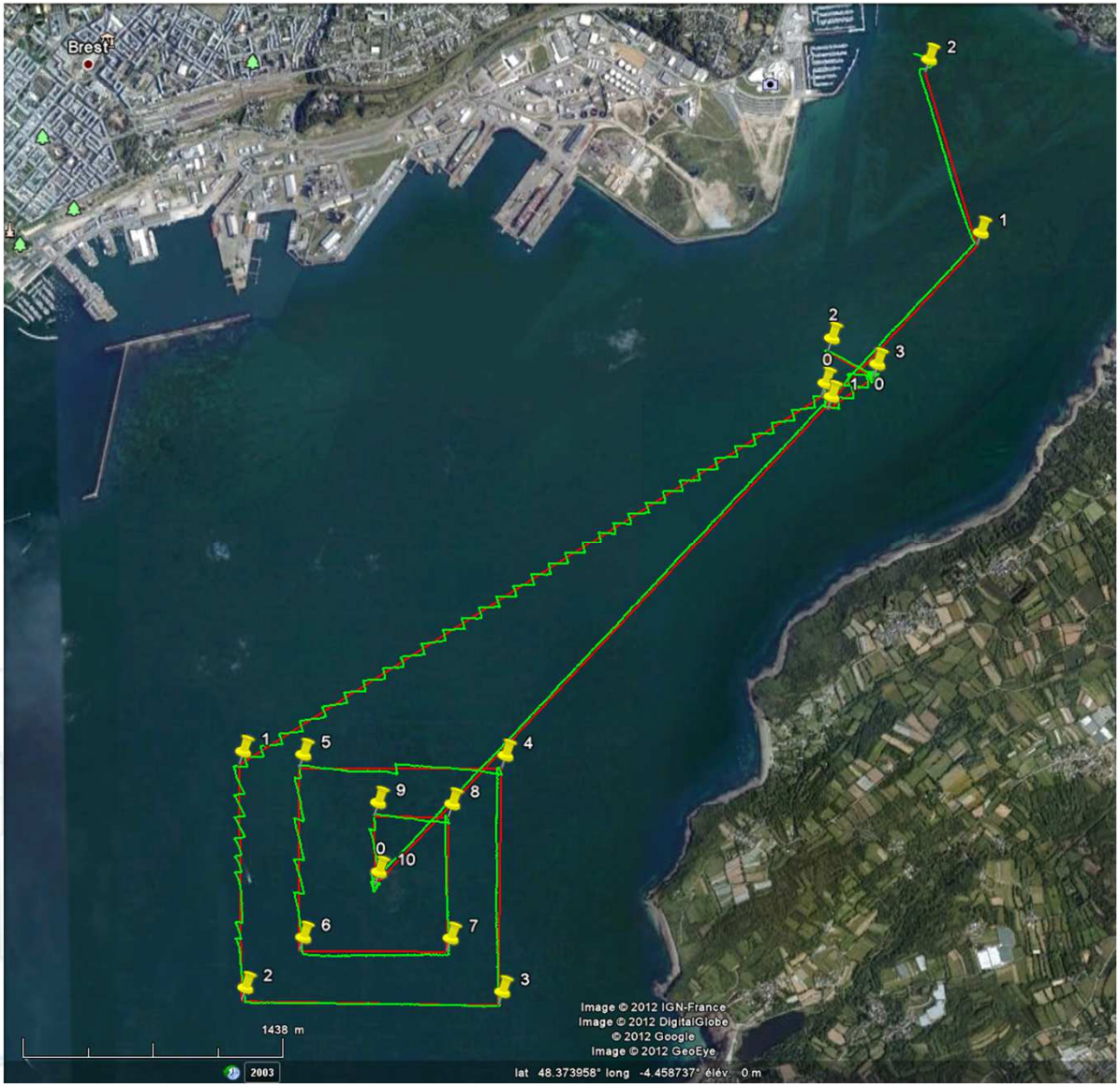

Copyright Ifremer


F78
Copyright Ifremer


15
0
1
14
13
10
9
12
11
6
5
8
2
7
4
3
F78
Copyright Ifremer

Real tests

Real tests

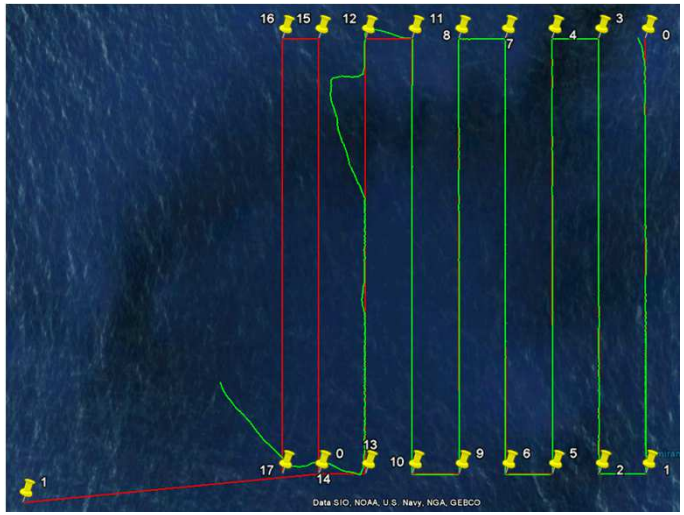

Brest
1438 m
2003
Image © 2012 IGN-France
Image © 2012 DigitalGlobe
© 2012 Google
Image © 2012 GeoEye
lat 48.373958° long -4.458737° élév. 0 m

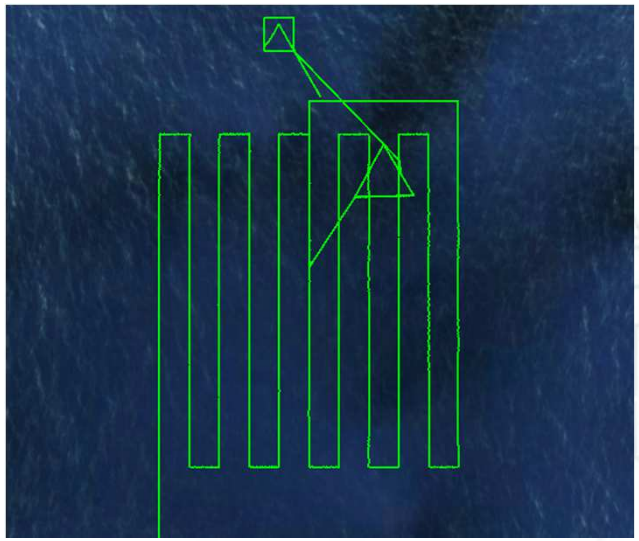
Real tests

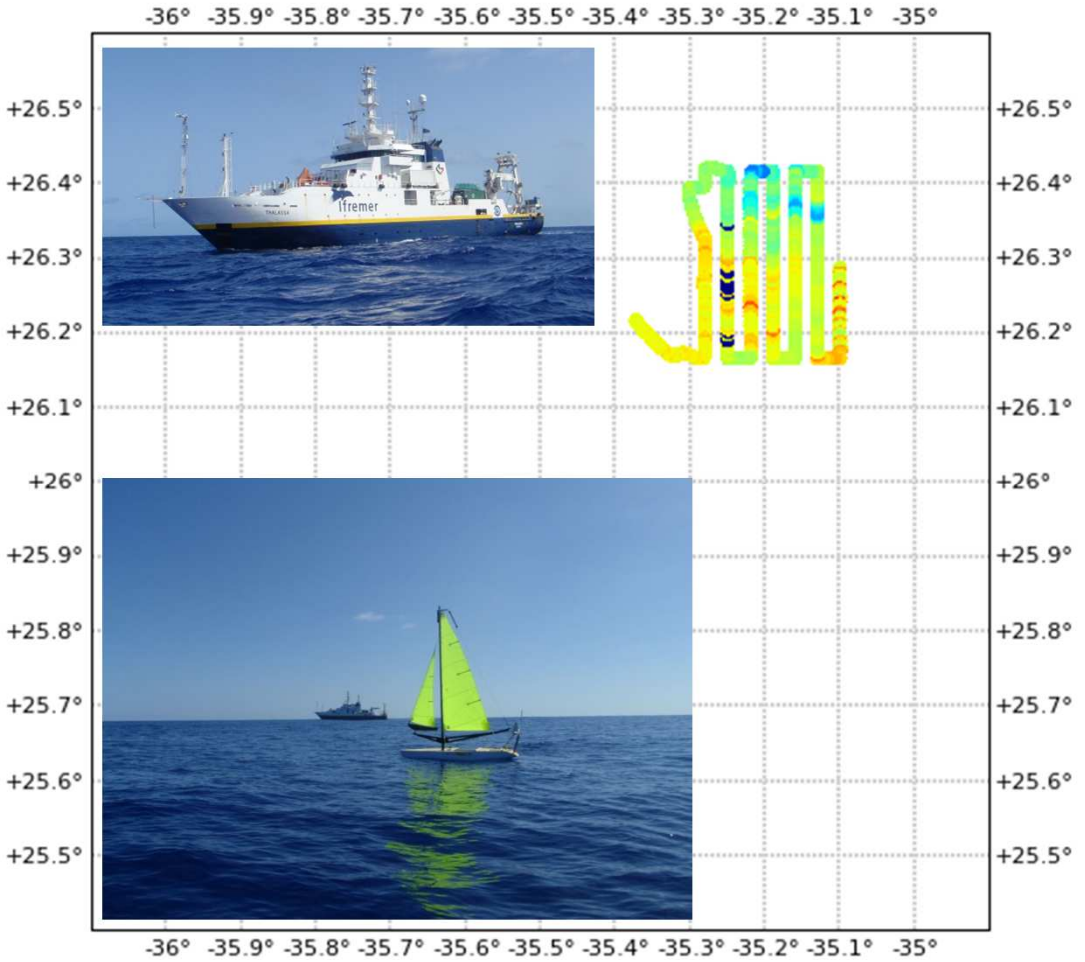
Details of a change between tack and nominal route, decided by the robot

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

Real tests

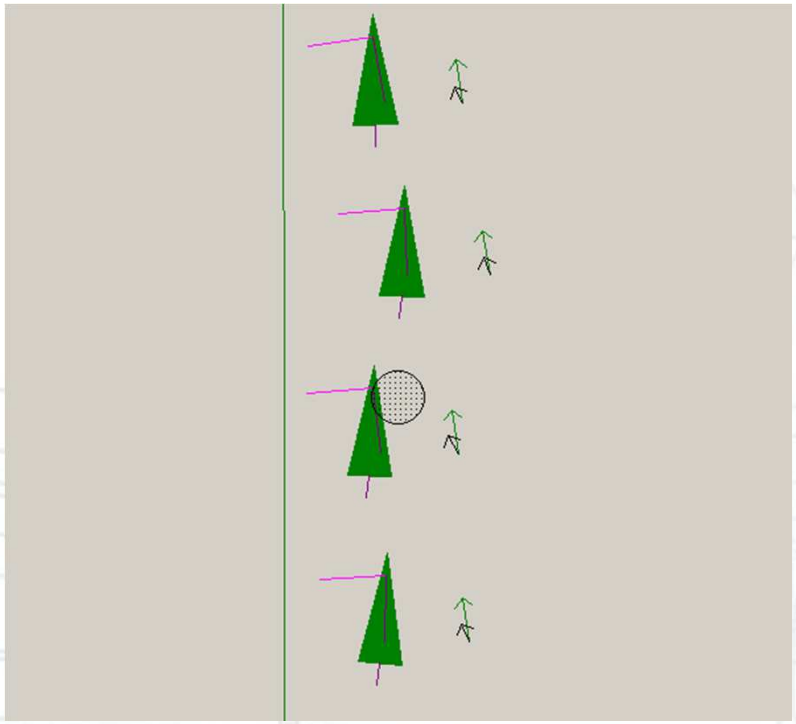

16 15 12 11 8 7 4 3 0
1 17 14 0 13 10 9 6 5 2 1
Data SIO, NOAA, U.S. Navy, NGA, GEBCO

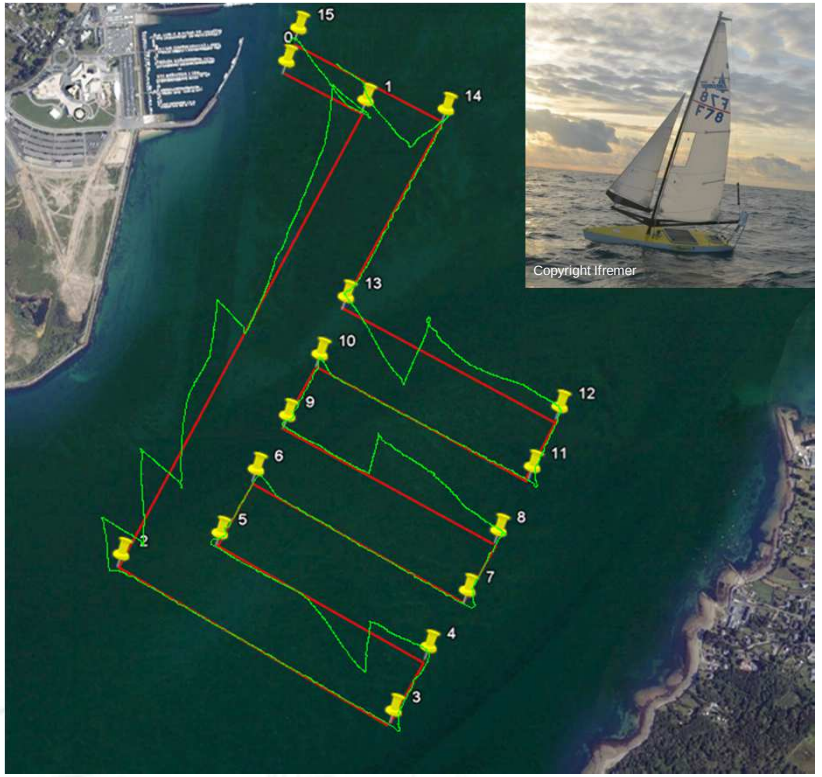



-36° -35.9° -35.8° -35.7° -35.6° -35.5° -35.4° -35.3° -35.2° -35.1° -35°
+26.5° +26.4° +26.3° +26.2° +26.1° +26° +25.9° +25.8° +25.7° +25.6° +25.5°
Ifremer

Real tests

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

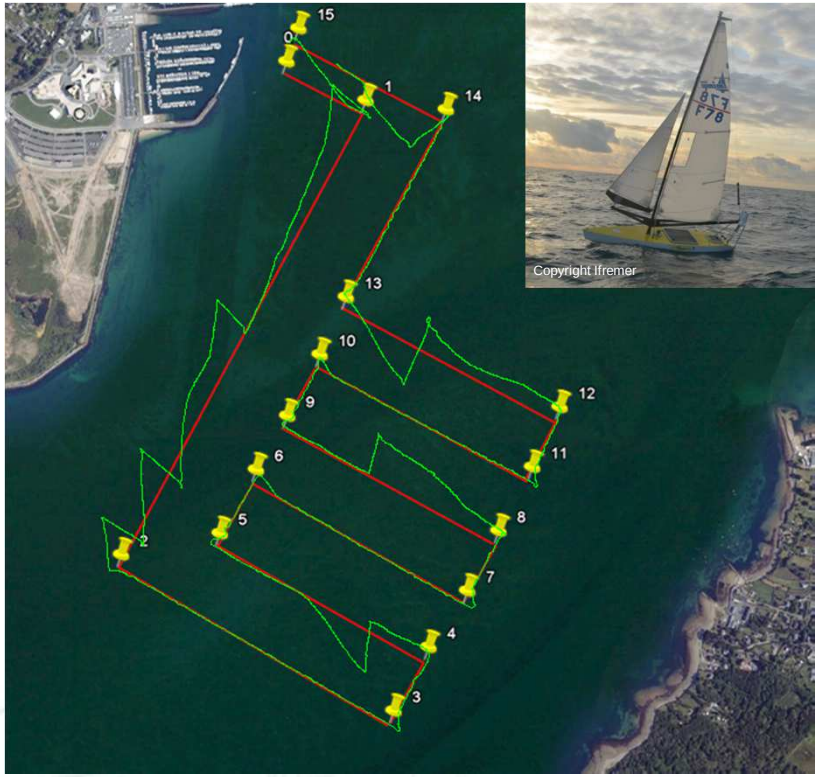




Higher level algorithms

Higher level algorithms

- Obstacle avoidance
- Tracking of other boats
- Communication and cooperation with other robots to act as a swarm
- Tow heavy loads
- Save/retrieve energy



Further work

Further work

- Make the line following algorithm available on ROS
- Use other formal validation methods
- Test and integrate it in other simulation frameworks
- Add fail-safe modes (e.g. without wind sensor, compass…)
- Improve energy management

Questions?