Chapter 1
An experimental validation of a robust controller with the VAIMOS autonomous sailboat

Fabrice Le Bars and Luc Jaulin

Abstract A sailboat is a strongly non-linear system that however has been proven to be easily controllable. Indeed, its mechanical design comes from an evolution from thousands of years with two main concerns: having a fast, reliable and efficient vehicle while being easily controlled by humans. This article describes the functioning, the validation process and the performances of a simple controller, inspired from what navigators do, through tests made on the sailboat robot *VAIMOS* built by IFREMER for oceanography. This controller requires tweaking few parameters with real physical meaning while ensuring accurate trajectory following, needed to make oceanographic measurements in a specific area.

1.1 Introduction

In order to make oceanographic measurements, IFREMER (Institut Français de Recherche pour l'Exploitation de la Mer) has designed a sailboat robot (see e.g. [4] [1] [16] [15] [6] [17] [2] [14] [18] for more information on autonomous sailboat robots) of 3.65 m based on a Mini-J hull: *VAIMOS* (Voilier Autonome Instrumenté pour Mesures Océanographiques de Surface, see figure 1.1). This robot has an oceanographic probe and pumps that make it possible to measure various parameters near the water surface and at a depth of about one meter (temperature, salinity, chlorophyll, turbidity...), a Linux-based embedded computer, a weather station (that measures the wind speed and direction as well as GPS position), an AHRS (Attitude and Heading Reference System), a Wifi and Iridium communication system and actuators for sail and rudder control (step-by-step motor that controls the maximum sail angle and servomotor to control the rudder angle) [7] [11] [12]. Its aim is to assist and / or replace currently used oceanographic boats and fixed or floating buoys, which have several drawbacks: oceanographic boats need a crew and their missions are expensive, it is sometimes difficult to set up fixed buoys in deep seas,

Fig. 1.1 The autonomous sailboat *VAIMOS* in the sea.

floating buoys move randomly according to wind and currents and do not always stay in desired areas... An autonomous sailboat has several advantages:

- Almost unlimited energy: it uses wind to move, sun and sea to charge its batteries while its power consumption is low compared to the one of a boat motor for instance.
- Interesting payload capabilities with respect to its dimensions.
- Accuracy (vs floating buoys) and easiness of setup (vs fixed buoys). The operators just need to program a predefined trajectory and launch the sailboat from a harbor : it should then go on the area of interest and cover it while storing measurements and communicating by satellite to send subsets of data and status information before coming back to its harbor.
- Cheap (about 20000€, probe excluded).

In addition to oceanographic missions, this type of robot could be used for other applications [3] [5]:

- Continuous harbor main entrance monitoring. Thanks to their important energetic autonomy and their low cost, several robots like *VAIMOS* could be deployed to monitor local surface and submarine traffic and would notably reinforce systems currently used.

- Heterogeneous swarms of robots. Submarine swarm of autonomous robots can quickly and furtively help monitoring and operate in a given area, however the use of small submarines alone can have several limitations :
  - It is more difficult to retrieve energy underwater. Therefore, small autonomous submarine robots can rarely work for long distances or time.
  - Localization and communication are difficult in passive mode (it is not always possible to use active sensors if furtivity is a requirement).

Adding autonomous sailboat robots to submarine swarms could solve some problems: long distance transport, energy backup, communications with the base or surrounding boats, localization thanks to GPS…

VAIMOS has been automated to be able to cover autonomously an area as accurately as possible, while saving energy. For this purpose, a line following algorithm [10] [11] has been developed to guarantee that the robot always stays in a predefined row (of 25 m width for example), despite maneuvers inherent to course changes, tacks… In this way, the sailboat becomes as accurate as a motorboat. Because of the existence of courses difficult to follow depending on wind orientation (which is inherent to any sailboat), its regulator has 2 types of strategies: nominal route or tack. A basic controller stage provides heading control. In tack mode, the heading to keep is around the wind orientation $\pm 45^\circ$ (clause hauled angle). Therefore, the boat oscillates around the wind angle, oscillations amplitude being the row width. The sail angle is at its minimum. In nominal route mode, the heading to follow is around the line made by the 2 current waypoints (the previous one and the next one), with an attractiveness angle to the line depending on the distance to the line (maximum of 45° for example). The sail is opened depending on the wind direction and the desired heading using a simple formula.

The main idea of this article is to show that in order to have a reliable autonomous robot, theoretical validation of its algorithms, using interval methods for example (see [9] for more information on interval analysis) is needed but we must also validate the assumptions made (state equations, bounds on errors, coefficients…) using other methods to complete the validation process. For these reasons, we first made a theoretical validation using interval analysis and Lyapunov methods [12]. Then, a HIL (Hardware In the Loop) simulator was developed. Finally, real experiments in Brest harbor and between Brest and Douarnenez (Brittany, France) were made.

The robust sailboat controller developed will be explained in section 1.2. Section 1.3 will be about the theoretical validation of the controller. The HIL simulator used as an additional validation method and tool to plan real experiments will be described in section 1.4. Finally, section 1.5 will show the results of real tests with *VAIMOS*.

1.2 Controller

Due to the functioning of a sailboat, some headings are difficult to follow depending on wind orientation. Therefore, most of the controllers have 2 different modes : nominal, when the heading to follow is feasible, or tack i.e. regulation around the wind angle $\pm 45^\circ$ (clause hauled angle), when it is not directly feasible. Most of existing regulators does waypoint following instead of line following for reachable headings [17]:

- The robot takes a heading in the direction of the waypoint.
- The waypoint is reached when the boat is in a predefined radius.
- Unfortunately, nothing prevents it to drift between waypoints (because of water currents, wind...).

Some use also potential fields to define no-go zones for the sailboat [14], cost functions, fuzzy logic and the polar speed diagram of the sailboat (VMG : Velocity Made Good) [18]. One of the first sailboat using a line following approach was *Atlantis* (and *HWT X-1*, its successor) [5] [4].

The inputs of a sailboat such as *VAIMOS* are δ_r the rudder angle and $\delta_s^{\max}$ the sail maximum angle (δ_s, the angle of the sail should depend on $\delta_s^{\max}$ and the wind orientation w.r.t. the sailboat orientation). The outputs are the position $\mathbf{x}$ obtained from the GPS and expressed in a local coordinate system, the wind speed V and orientation ψ from the weather station and the heading θ of the sailboat from the AHRS, used as a compass (see figure 1.2). Note that it is possible to avoid using a weather station and keep $\delta_s^{\max}$ as a constant using the methods described in [19]. The line following controller of VAIMOS (described in details in [10], [11] and

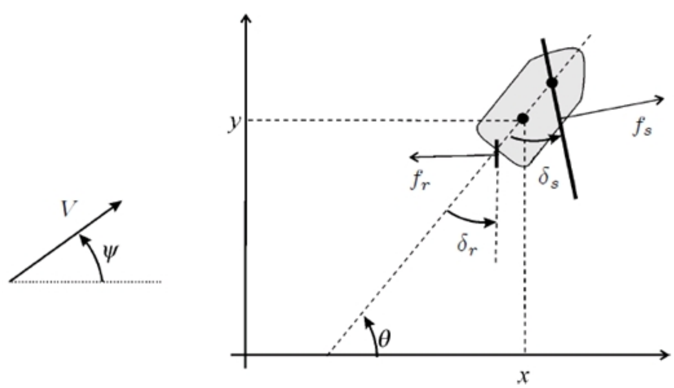

Fig. 1.2 Notations: ψ and V define the wind orientation and speed, f_s is the force of the wind on the sail and δ_s the angle of the sail, f_r is the force of the water on the rudder and δ_r the rudder angle, $\mathbf{x} = (x, y)$ and θ are the boat position and orientation.

[12]) is composed of several parts (see figure 1.3):

- A primitive controller stage for heading control. The angle of the rudder is set by proportional regulation w.r.t desired heading $\bar{\theta}$ if we are close to this desired heading, bang-bang regulation if far from desired heading:

$$\delta_r = \begin{cases} \delta_r^{\max}.\sin\left(\theta-\bar{\theta}\right) & \text{if } \cos\left(\theta-\bar{\theta}\right)\geq 0 \\ \delta_r^{\max}.\text{sign}\left(\sin\left(\theta-\bar{\theta}\right)\right) & \text{otherwise,} \end{cases} \tag{1.1}$$

with $\delta_r^{\max}$ the maximum rudder angle. The sail is opened depending on the wind direction and the desired heading using a simple formula:

$$\delta_s^{\max} = \frac{\pi}{2}.\left(\frac{\cos\left(\psi-\bar{\theta}\right)+1}{2}\right). \tag{1.2}$$

- A supervisor decides between 2 modes: nominal route or tack. It should always send feasible headings to primitive controller. In nominal route mode, the heading to follow is given by the line made by the 2 current waypoints **a** and **b**, with an attractiveness angle to the line depending on the distance to the line. In tack mode, heading is around the wind orientation $\pm 45^\circ$ (clause hauled angle): the sailboat oscillates around the wind angle, oscillations amplitude being the row width around the line.
- A navigation manager sends lines formed by 2 waypoints $\mathbf{a}_j$ and $\mathbf{b}_j$ to the supervisor and validates lines. A line is validated when the sailboat reaches the perpendicular of the line at $\mathbf{b}_j$ i.e. the validation condition is:

$$\left\langle \mathbf{b}_j - \mathbf{a}_j, \mathbf{x} - \mathbf{b}_j \right\rangle \geq 0 \tag{1.3}$$

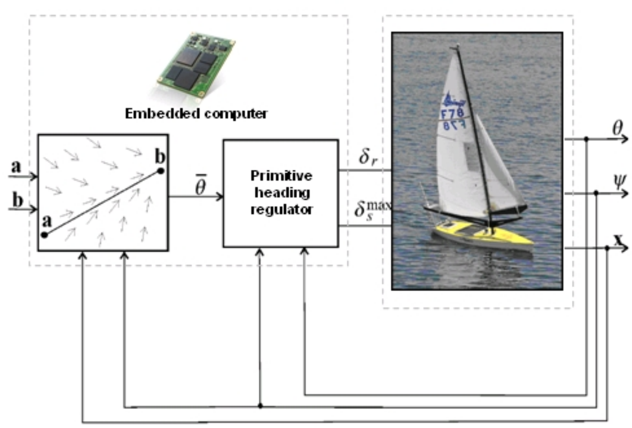

Fig. 1.3 Principle of the line following controller of *VAIMOS*.

1.3 Theoretical validation of the controller

In order to validate the line following controller developed, several tools have been used:

- 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).

A new interval method for nonlinear stability analysis has been developed [12]. The main idea is to represent uncertain systems by differential inclusions and then apply Lyapunov analysis methods to transform the stability problem in a set inversion problem (see [13] and [9] for more information on interval analysis and set inversion problems). In this way, it is possible to demonstrate 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.

However, even if these methods 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 adjust hypothesis (state equations, bounds on sensors errors…). A HIL simulator (inspired from [8]) has been developed to simulate the robot trajectory on a computer as well as sensors data while using the controller directly on the sailboat as if it thought it was in the sea to prepare as much as possible real experiments.

1.4 HIL simulator

Most of existing simulators use the polar speed diagram of the sailboat to determine its movement or several predefined scenarios. Therefore, they might miss some singular situations that one would detect and emphasize to fully validate a controller. State equations inspired from [8] were used for our controller validation purpose:

$$
\begin{cases}
\sigma & = \cos(\theta - \psi) + \cos(\delta_s^{\max}) \\
\delta_s & = \begin{cases} \pi - \theta + \psi & \text{if } \sigma < 0 \\ \delta_{s\max} \operatorname{sign}(\sin(\theta - \psi)) & \text{otherwise} \end{cases} \\
f_r & = \alpha_r v \sin(\delta_r) \\
f_s & = \alpha_s V \sin(\theta + \delta_s - \psi) \\
\dot{x} & = v\cos(\theta) + \beta V \cos(\psi) + V_c \cos(\psi_c) \\
\dot{y} & = v\sin(\theta) + \beta V \sin(\psi) + V_c \sin(\psi_c) \\
\dot{\theta} & = \omega \\
\dot{\omega} & = \frac{(l - r_s \cos(\delta_s)) f_s - r_r \cos(\delta_r) f_r - \alpha_\theta \omega + \alpha_w h_w}{J_z} \\
\dot{v} & = \frac{\sin(\delta_s) f_s - \sin(\delta_r) f_r - \alpha_f v^2}{m} \\
\ddot{\varphi} & = \frac{-\alpha_\varphi \dot{\varphi} + f_s h_s \cos(\delta_v)\cos(\varphi) - m_{eq} l_{eq} g \sin(\varphi)}{J_x} \\
\dot{\varphi} & = \dot{\varphi}
\end{cases}
\tag{1.4}
$$

with v the sailboat speed, ω the rotation speed, φ the roll, assumed to be pendular, with coefficients α_φ (fluid friction), h_s (height of the sail force application point), m_{eq} (mass of the equivalent pendulum), l_{eq} (length of the equivalent pendulum), J_ω (inertial moment), V_c and ψ_c the sea current speed and orientation, h_w the height of waves, β the coefficient of drift due to wind, α_r, α_s, α_f, α_θ, α_w various fluid friction coefficients and r_r, r_s, the distance from the sailboat mass center to the rudder and mast respectively (see also figure 1.2). Then, the behaviour of the state equations was tested on a 3D simulator. Finally, a HIL (Hardware In the Loop) simulator was developed to simulate the robot trajectory and sensors data on a computer depending on the input lines, expected wind and sea conditions and a user-defined initial position while using the developed controller on the embedded computer to control the robot actuators as if the sailboat thought it was in the sea to study mechanical wear as well as robustness of most of the embedded electronics. HIL simulation means that the real hardware (here the embedded computer and actuators) are used in simulations (see figure 1.4):

- First, the simulator using the state equations previously defined is started on a normal computer with a user-defined initial state. It generates simulated sensors data (θ, ψ, $\mathbf{x}$) from rudder (δ_r) and sail ($\delta_s^{\max}$) inputs that will be decided by the controller and user-defined sea (h_w, V_c, ψ_c) and wind (V, ψ) conditions.
- Then, the controller is started on the embedded computer of the sailboat. It takes a list of lines to follow (formed by waypoints $\mathbf{a}_j$, $\mathbf{b}_j$) like for a real experiment and controls its actuators as usual, but sends also a copy of its outputs for the actuators (δ_r and $\delta_s^{\max}$) to the simulator and uses simulated sensors data from the simulator (θ, ψ, $\mathbf{x}$) rather than the data from its real sensors.
- Finally, log files generated by the controller are retrieved and displayed in real time using GOOGLE EARTH and a specific dashboard.

All the communications are made possible by the fact that all embedded devices are accessible by Wifi.

Several simulations were made in different configurations to prepare real experiments such as a trajectory of more than 100 km between Brest and Douarnenez. A

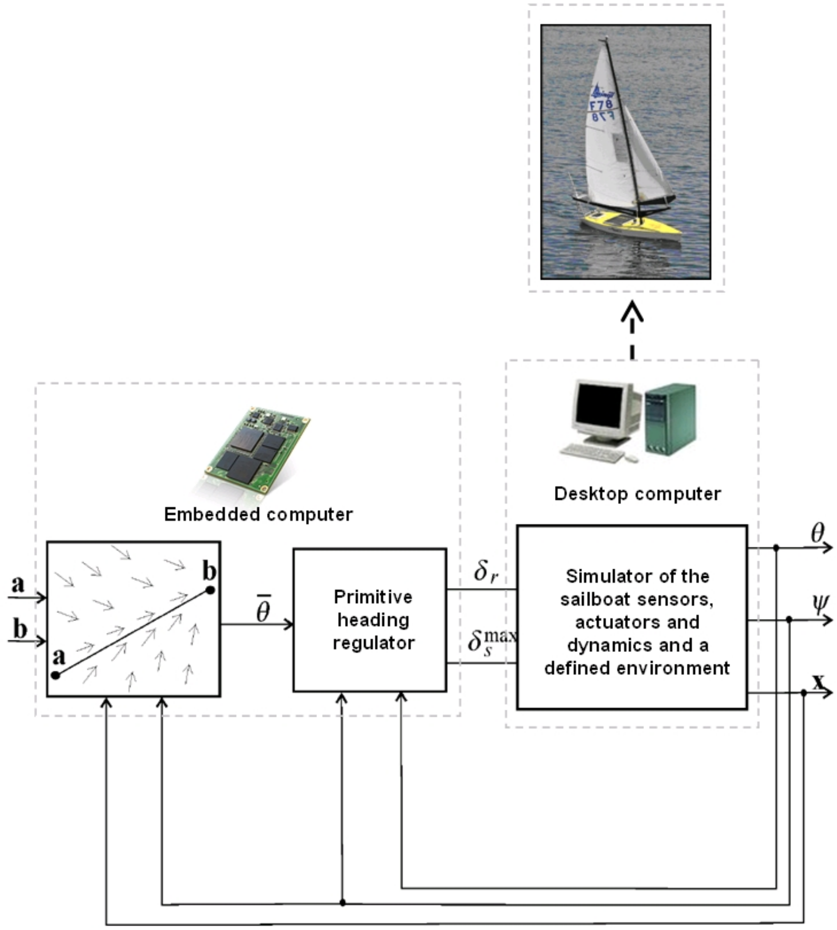

Fig. 1.4 Principle of the HIL simulator of *VAIMOS*.

first simulation made with a North wind of 14 kn is shown in figure 1.5. An initial position just in front of Brest harbor was indicated to the simulator at start. Other simulations were made as the expected weather conditions for the date fixed for the real test was changing, and to test different ways of covering the bay to minimize the tacks and shorten the total time (here around 40 hours).

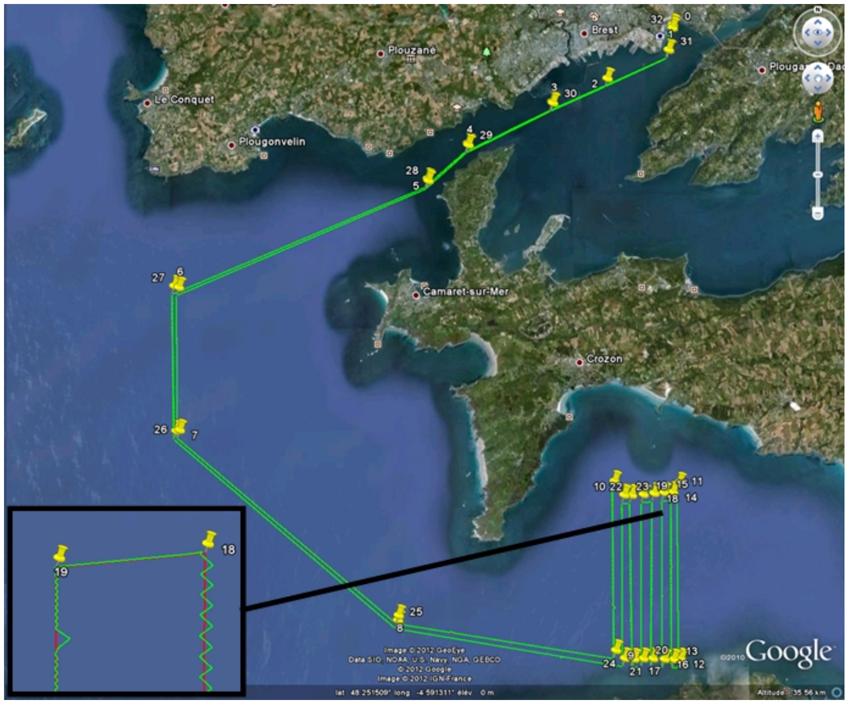

Fig. 1.5 Simulation of *VAIMOS* going from Brest to Douarnenez. The desired trajectory (red lines made by yellow waypoints) and the effective (simulated) trajectory (green) seem to be overlapped. However, if we zoom, we see there were tacks. We see also calibration steps of the sail motor (every 2 hours) that makes the sailboat drifting during one minute.

1.5 Real experiments

Real experiments of singular trajectory patterns have been made in Brest harbor to test *VAIMOS* in all wind configurations while taking oceanographic measurements for IFREMER (see figure 1.6). Small real tests are important. For example, some magnetic perturbation problems necessitating to move the AHRS far from the rest of the electronics were detected during these tests.

Finally, a long autonomous mission between Brest and Douarnenez the 17-18th January 2012 was attempted with *VAIMOS* (see figure 1.7). It made more than 500 oceanographic measurements on 105 km in 19 h. The wind was around 12 kn.

During the mission and after, a dashboard was used to analyse all the log files produced by the embedded program. For example, near the end of the experiment, we see that the sail angle measured by the weather station (which is on top of the sail and has an integrated compass), in purple is incoherent with the one deduced from the input, in pink. This was probably due to the mechanical problem in the sail

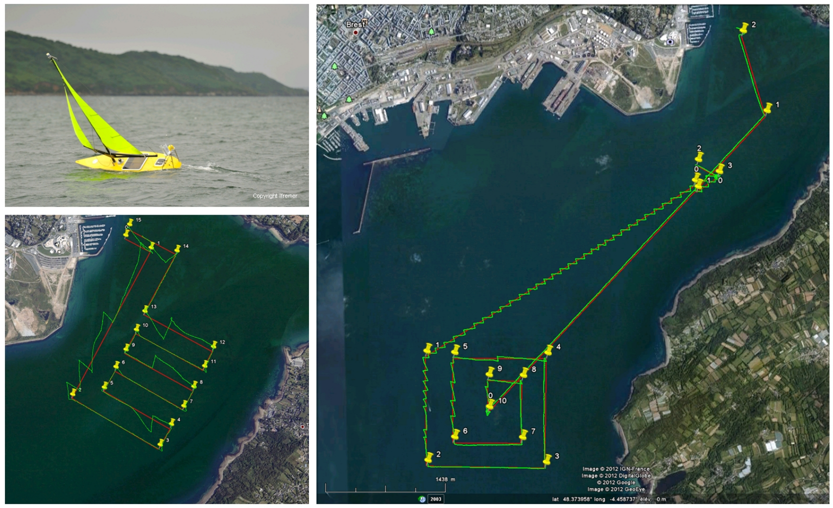

Fig. 1.6 Tests in Brest harbor. Desired trajectory (red lines made by yellow waypoints) and effective trajectory (green). South-West wind on the left, South-West wind of around 15 kn, 27km (17 nm) in less than 5h for the right.

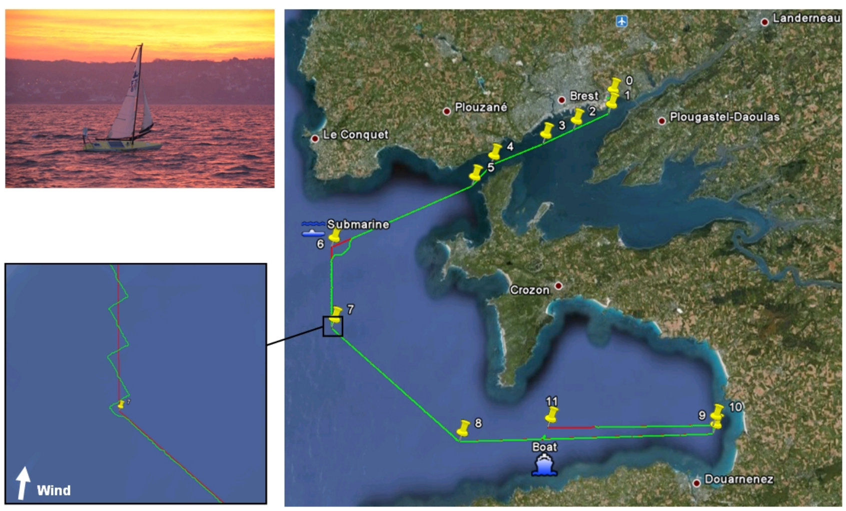

Fig. 1.7 Brest-Douarnenez. The sailboat needed to be deviated twice: first because of a submarine coming back to Brest naval base, then because of a static boat in the sailboat trajectory. During these perturbations, the autonomous program was not changed nor stopped, the sailboat was taken by our chase motorboat. Therefore, the submarine and boat deviations illustrate the robustness of the controller, that was able to continue the mission as if nothing happened.

control system that we discovered at the end. Because it was during the night, it was difficult to see the sail angle without the dashboard (see figure 1.5).

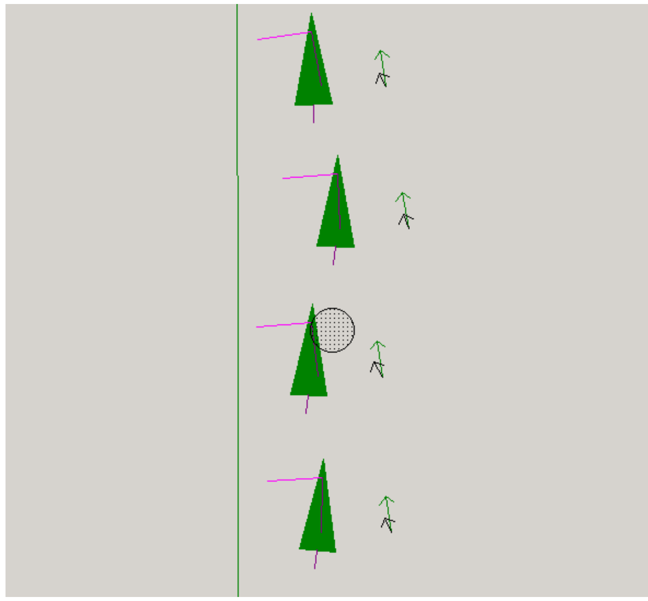

Fig. 1.8 Analysis of log files using a dashboard.

1.6 Conclusion

In this article, we showed that theoretical methods such as 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.

Acknowledgements VAIMOS has been built from a collaboration between LPO (Laboratoire de Physique des Océans), RDT (Recherches et Développements Technologiques) of IFREMER (Institut Français de Recherche pour l'Exploitation de la Mer) and ENSTA Bretagne (Ecole Nationale Supérieure de Techniques Avancées Bretagne). The authors would like to thank LPO and RDT to have enabled the use of VAIMOS as a test platform for algorithms from ENSTA Bretagne. They thank also everyone that has contributed to the project : Yves Auffret, Patrick Rousseaux, Sébastien Prigent, Loïc Dussud, Stéphane Barbot, Loïc Quemeneur, Bertrand Forest (RDT, IFREMER) and Olivier Ménage, Fabienne Gaillard, Thomas Gorgues, Thierry Terre (LPO, IFREMER).

References

1. Y. Brière. *The first microtransat challenge*, http://web.ensica.fr/microtransat. ENSICA, 2006.
2. R. Bruder, B. Stender, and A. Schlaefer. Model Sailboats as a Testbed for Artificial Intelligence Methods. In *IRSC 2009*, 2009.
3. N.A. Cruz and J.C. Alves. Ocean sampling and surveillance using autonomous sailboats. In *IRSC 2008*, 2008.
4. G.H. Elkaim and R. Kelbley. Station Keeping and Segmented Trajectory Control of a Wind-Propelled Autonomous Catamaran. In *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, USA, 2006.
5. G.H. Elkaim and C.O. Lee Boyce Jr. An Energy Scavenging Autonomous Surface Vehicle for Littoral Surveillance. In *ION Global Navigation Satellite Systems Conference*, 2008.
6. H. Erckens, G.A. Büsser, C. Pradalier, and R.Y. Siegwart. Navigation Strategy and Trajectory Following Controller for an Autonomous Sailing Vessel. *IEEE RAM*, 17:47–54, 2010.
7. T. Gorgues, O. Ménage, T. Terre, and F. Gaillard. An innovative approach of the surface layer sampling. *Journal des Sciences Halieutique et Aquatique*, 4:105–109, 2011.
8. L. Jaulin. Modélisation et commande d'un bateau à voile. In *CIFA2004 (Conférence Internationale Francophone d'Automatique), In CDROM*, Douz (Tunisie), 2004.
9. L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001.
10. L. Jaulin and F. Le Bars. A simple controller for line following of sailboats. In *IRSC 2012*, Cardiff, UK, 2012.
11. L. Jaulin, F. Le Bars, B. Clément, Y. Gallou, O. Ménage, O. Reynet, J. Sliwka, and B. Zerr. Suivi de route pour un robot voilier. In *CIFA 2012*, Grenoble, France, 2012.
12. L. Jaulin, F. Le Bars, and O. Ménage. An interval approach for stability analysis; Application to sailboat robotics. *Submitted to IEEE Transaction on Robotics*, 2012.
13. L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, 1993.
14. M.A. Romero-Ramirez. *Contribution à la commande de voiliers robotisés*. PhD dissertation, Université Pierre et Marie Curie, France, 2012.
15. P.F. Rynne and K.D. von Ellenrieder. Unmanned autonomous sailing: Current status and future role in sustained ocean observations. *MTS Journal*, 43(1):21–30, 2009.
16. C. Sauze and M. Neal. An autonomous sailing robot for ocean observation. In *in proceedings of TAROS 2006*, pages 190–197, Guildford, UK, 2006.
17. J. Sliwka, P. Reilhac, R. LeLoup, P. Crepier, H. D. Malet, P. Sittaramane, F. LeBars, K. Roncin, B. Aizier, and L. Jaulin. Autonomous robotic boat of ensieta. In *2nd International Robotic Sailing Conference*, Matosinhos, Portugal, 2009.
18. Roland Stelzer and Tobias Pröll. Autonomous sailboat navigation for short course racing. *Robot. Auton. Syst.*, 56(7):604–614, July 2008.
19. K. Xiao, J. Sliwka, and L. Jaulin. A wind-independent control strategy for autonomous sailboats based on voronoi diagram. In *CLAWAR 2011 (best paper award)*, Paris, 2011.