

# Improving Instrumentation Support and Control Strategies for Autonomous Sailboats in a Regatta Contest

Luís Gomes, Anikó Costa, David Fernandes, Hugo Marques and Filipe Anjos

**Abstract** This paper presents a controller architecture targeted for autonomous sailing of a small yacht, having in mind its control within a regatta contest. The controller architecture considers a three layer hierarchical decomposition, where the bottom layer takes care of the low level control of the sail and the rudder, based on the usage of a fuzzy controller, while the middle one selects the adequate navigation strategy (avoiding no-go zone), and the top layer is responsible for the definition of the regatta sequence of intermediate and final goals. A model-based development strategy is used, relying as much as possible on automatic code generation from models. Initial electronic instrumentation support comes from a compass, a GPS and a wind vane, but current developments include addition of an anemometer (providing wind speed and direction), as well as the introduction of virtual sensors obtained from computation of sensor data (as velocity obtained from GPS). A framework allowing the emulation of the different electronic instrumentation components support a complete offline validation of the controller, in a laboratory environment.

## 1 Introduction

The present paper addresses the development framework for a sailboat controller, starting from a sailboat built for RC control, where the RC control part was replaced by a controller board and adequate electronic instrumentation. The selected sailboat is

---

L. Gomes (✉) · A. Costa  
FCT and UNINOVA, CTS, Universidade Nova de Lisboa, Lisbon, Portugal  
e-mail: lugo@fct.unl.pt

A. Costa  
e-mail: akc@fct.unl.pt

D. Fernandes · H. Marques · F. Anjos  
FCT, DEE, Universidade Nova de Lisboa, Lisbon, Portugal  
e-mail: ds.fernandes@campus.fct.unl.pt

H. Marques  
e-mail: hm.marques@campus.fct.unl.pt

F. Anjos  
e-mail: f.anjos@campus.fct.unl.pt

a 1 m replica of America's Cup racing yacht, while the selected controller board is an Arduino Mega [1], and basic electronic instrumentation includes a compass, a GPS, and a wind vane. All together it is a low-cost sailboat amenable to support autonomous navigation (however without any collision detection or avoidance mechanism).

This work benefits from lesson learnt from previous works on autonomous sailboats as a result of several meetings and interactions with colleagues from other institutions, in particular from the FAST initiative from University of Porto - Faculty of Engineering [2, 3], as well as from several experiments with LaserRC yachts owned by Portuguese Navy School.

One of the goals of the work is to build a simple platform amenable to be used in attracting students to STEM (science, technology, engineering and mathematics) areas through specific initiatives, namely short summer courses for high school students.

The paper starts to present the developed controller, giving information on both the hardware and software architectures. Next section briefly presents computational tools that were used to develop the controller, while following section addresses new electronic instrumentation components under development, as well as an emulation framework to support laboratory full testing of the controller behavior. Finally, some conclusions and future works are presented.

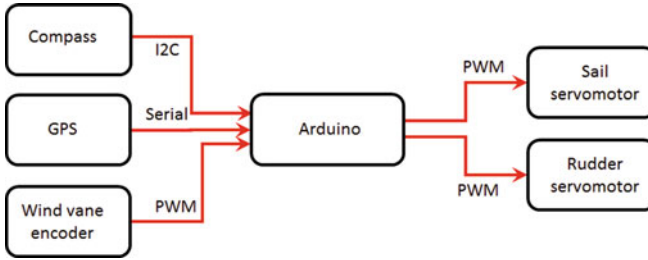
## 2 Hardware and Software Architectures

This section is divided into two main subsections, while the first one briefly presents the hardware architecture, the second addresses the three layer hierarchical software controller. The descriptions expand and update previous work presented elsewhere [4].

### 2.1 *Hardware Architecture*

Selection of the hardware platform took into consideration several criteria, including overall low-cost, moderate power consumption, as well as easy of programming, even for non-experienced programmers (as high school students are also targeted users of the platform). There are several platforms satisfying those requirements, but the selection goes for an open hardware platform, which also has a strong community behind: the Arduino platform [1]. It is also important to note that this selection does not compromise usage of the proposed solution in other platforms (or by more experienced users), due to its modular nature and easy porting of the software solution.

A minimum set of electronic instrumentation components was connected to the Arduino Mega, including:



**Fig. 1** Hardware subsystems interconnections and selected interface protocols

- a tilt compensated compass (providing also information from accelerometers, together with the pitch and roll from 3-axis gyro; I2C interface was elected for interconnection; the CMPS10 and CMPS11 modules have been used),
- a GPS receiver (the MediaTek MT3329 GPS V2.0 module was selected; serial communication was used), and
- the wind vane was homemade based on a contactless magnetic position encoder (the ams AS5040 Adapter board was selected, using PWM interface).

Two PWM controlled servomotors commonly used in RC control were used to control the sail and the rudder. A block diagram of the hardware architecture is illustrated in Fig. 1; it is similar to common architectures found in the literature, namely in [5].

The diversity of interconnection solutions allows a good coverage on common interfaces to interconnect hardware modules (and anticipate the use of the platform as a didactic set-up supporting embedded systems development teaching).

## 2.2 Three Layer Hierarchical Controller

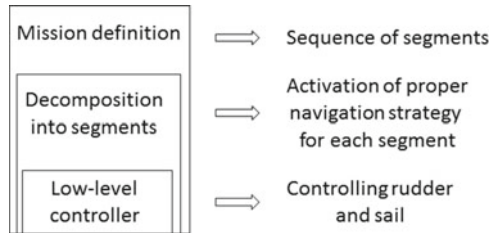
As already referred, the goal is to support autonomous sailing within a regatta path contest. For that end, a three layer hierarchically structured controller is proposed, as presented in Fig. 2.

The top level is responsible for the mission definition, where a list of intermediate (considering the different buoys) and final goals is defined, establishing the list of segments to follow from the start line to the end goal finish line. Starting with the first segment in the list, after detection of goal accomplishment on that segment, the control will iterate on the segment list till reaching final destination.

The middle layer is responsible to elect the right navigation strategy for each segment of the path, taking into consideration current direction of the wind, current orientation of the sailboat, and intended direction of sailboat movement.

Finally, the lower layer is responsible for actuate sail and rudder. Following two subsections give further details on the two lower layers.

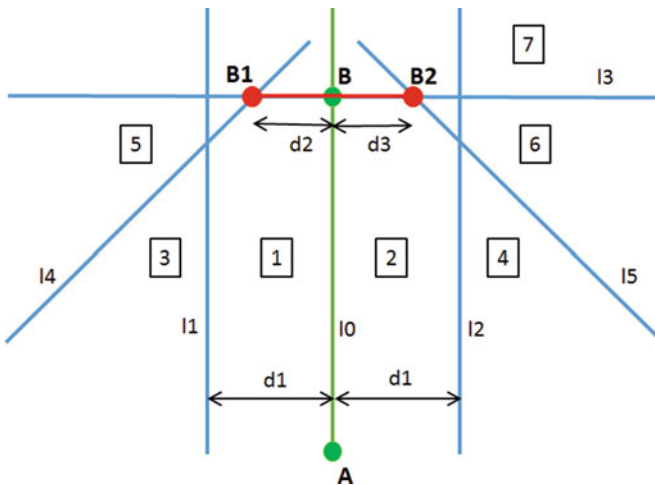
**Fig. 2** Three layer structure of the controller



**2.2.1 Defining Navigation Strategy**

Each navigation segment is characterized by a set of points, namely initial point A, destination point B, as well goal line B1-B2. It is important to note that distance between B1 and B, and B and B2 (denominated as  $d_2$  and  $d_3$  respectively in Fig. 3) can be different. This characterization allows easy adaptation to different requirements, namely when the goal line B1-B2 represents the final goal line (and where  $d_2$  and  $d_3$  would be most probably equal), as well as intermediate goals (as the one going around a buoy, where the destination point B is not far from the buoy and B1 can impose a minimum distance to the buoy, and where B2 can allow some flexibility to adjust to specific navigation constraints).

Current navigation strategy defines an area where navigation is possible (regions 1 and 2 of Fig. 3), including close reach navigation. Starting with the definition of the desired optimal trajectory (line  $l_0$  in Fig. 3), two equally spaced parallel lines are defined considering distance  $d_1$  to  $l_0$  (lines  $l_1$  and  $l_2$  in Fig. 3). Our current setup considers  $d_1$  equal to 5 m (however, future works will study different values



**Fig. 3** Geometric characterization of navigation corridor considering initial position A, and goal line B1-B2

considering specific navigation situations). When approaching goal line, it is necessary to consider navigability minimum conditions (namely for close reach navigation), and two lines (14 and 15 in Fig. 3) were defined having a specific angle in relation with line 13 (which includes the goal line), and allowing detection of moving outside the navigation area. As a very conservative navigation strategy has been adopted, an angle of 45 degrees has been considered (however, future works will address improvements on this aspect, considering specific type of sailboat and speed).

In this sense, considering the geometry presented in Fig. 3, several regions are considered:

- based on splitting the area using goal line (line 13), generating region 7 and the others,
- based on splitting the area using vertical lines, namely regions 3, 1, 2, and 4, separated by lines 11, 10, and 12, respectively,
- based on splitting the area using lines 14 and 15 adequate to split between no-go zone and navigation area (used when the sailboat is in close reach navigation approaching the goal line).

Based on the presented geometric characterization and on wind direction related with goal direction, the adequate navigation mode is selected in order to avoid the no-go zone. Figure 4 summarizes the behavior using a state diagram notation.

When entering into any of the three referred states new initial point and destination point (final goal) for navigation direction are set according with the following:

- for Regular navigation state, initial and final point will be A and B, as initially defined for the segment;
- for state Into the wind (port) as well as for state Into the wind (starboard), the initial point of the navigation direction is set to the current position of the sailboat, while the destination point is set to a point belonging to a line resulting from a sailboat tack (outside the navigation area), using a series of close-hauled segments to beat a course upwind.

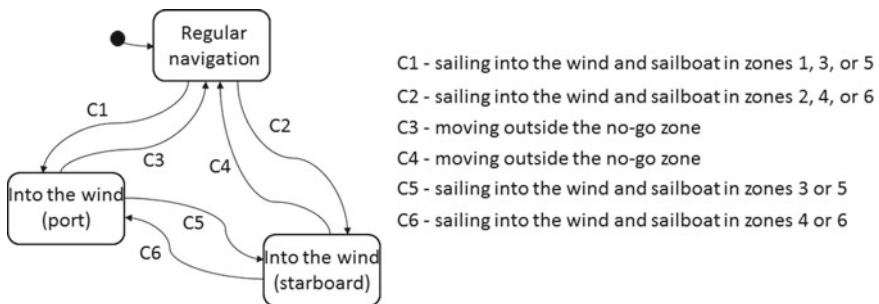
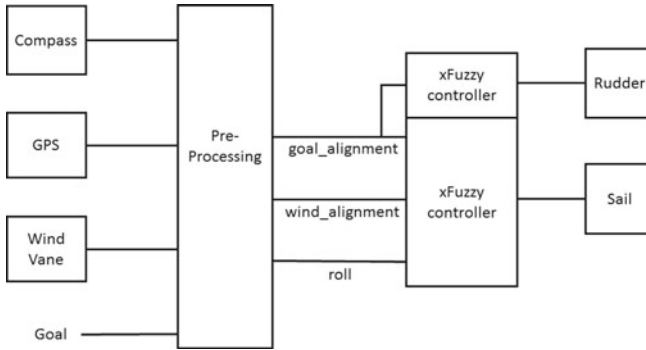


Fig. 4 State diagram governing navigation mode selection



**Fig. 5** Block diagram of the low level controller

### 2.3 Low Level Fuzzy Controller

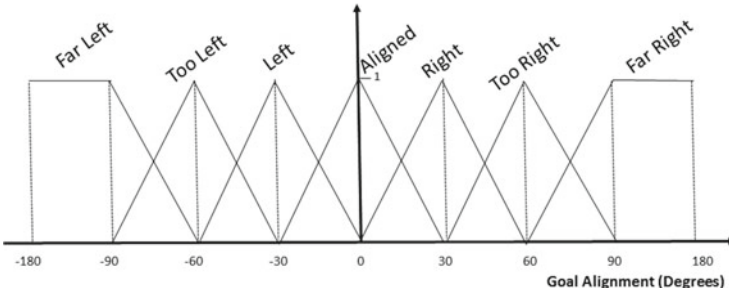
Fuzzy control techniques were selected for the implementation of the low-level controller both for rudder and for sail, as in [6]. Main motivation is associated with the fact that it would be easier to describe behavioral dependencies using a set of rules (as reliable equation models are not available), as well as with this strategy it is possible to include non-experienced users to exercise different control strategies. A block diagram for the low-level controller is presented in Fig. 5.

A preprocessing block is responsible to obtain data from the compass (sailboat current orientation, as well as roll attitude), GPS (current location), and wind vane (direction of the wind in relation with sailboat orientation). This information was considered to provide most relevant dependencies and support development of a simple, although reliable, controller. The variables selected to be inputs of the fuzzy controllers are only three:

- `goal_alignment` (obtained by the difference between the sailboat orientation and the desired direction towards the goal); this variable is used by both rudder and sail controllers;
- `wind_alignment` (obtained by the difference between the wind direction and the sailboat orientation); this variable is used only by the sail controller;
- `roll` (obtained from compass module, and providing an indirect information on wind velocity); this variable is used only by the sail controller.

For the rudder controller, the `goal_alignment` variable is the only input, characterized as illustrated in Fig. 6, a set of fuzzy rules is defined as presented in Fig. 7, where output variable is characterized using five possible singleton values.

For the sail controller, a similar approach was followed, considering three input variables, namely `goal_alignment` (holding the seven fuzzy values presented in Fig. 6), `wind_alignment` (holding three fuzzy values: stern, slant, and bow), and `roll` (holding two fuzzy values: right and heel), while output fuzzy variable holds five possible singleton values (`full_loose`, `loose`, `half`, `tight`,



**Fig. 6** Fuzzy sets associated with goal\_alignment input variable

**Fig. 7** Fuzzy rules associated with rudder controller

IF goal_alignment IS	THEN rudder IS
far_left	full_port
too_left	port
left	centered
aligned	centered
right	centered
too_right	starboard
far_right	full_starboard

full\_tight). The controller is composed by a set of 42 fuzzy rules. A description of a former version of the controller (although satisfying the same structure) can be found in [4].

### 3 Development Environment

One important aspect to refer when coming to development framework, already referred before, is related with capability to adopt a model-based development approach allowing direct support from automatic code generation tools. This is of paramount importance considering that one of the goals of the work is to use the platform to be used by non-experienced users, namely students from high school.

This is the case for the two described software layers, namely the middle layer, which is governed by a state machine and where the goal is to select the proper navigation strategy for each segment. For that, the IOPT-Tools development framework was used [7], allowing automatic generation of C code that can be readily deployed into the Arduino platform. For that end, an IOPT model [8] notation was used to represent the state diagram of Fig. 4.

This is also the case for the lower layer, where the fuzzy controllers were specified using the Xfuzzy framework [9], which allows specification of the set of fuzzy rules, as well as input and output fuzzy sets. Xfuzzy allows generation of C code, as well as C++, Java, and VHDL [10, 11], which after some manual changes can be integrated

with the code generated by the IOPT-Tools and directly deployed into the Arduino platform.

Both development frameworks are publically available, free of charge.

## 4 Improving Electronic Instrumentation and Testing

Experimental results using the presented controllers' characterization proved to be adequate to accomplish the desired control strategies. The presented sailboat controller has been validated in several frameworks, including laboratory as well as navigation in small lakes, and larger and open areas (as in river Tagus, Fig. 8).

However, some weaknesses were identified along experimentation. First of all, the need to have some simulation/emulation environment allowing the developers to test the project even with no navigation conditions, namely in the laboratory. From the three instruments on use, only the wind vane can be operated inside the laboratory forcing any desired value by hand. The GPS receiver is completely useless inside the building. On the other hand, values provided by the compass can be partially tuned (namely different values for the roll and pitch can be read).

Another important weakness is associated with the lack of information about sailboat velocity, as well as wind speed and direction. These two aspects receive special attention, briefly reported in the following subsections.

### 4.1 *Introducing Emulation Environment*

A basic emulator, also based in an Arduino board, was developed allowing two types of operation:

**Fig. 8** Sailboat in autonomous navigation in river Tagus





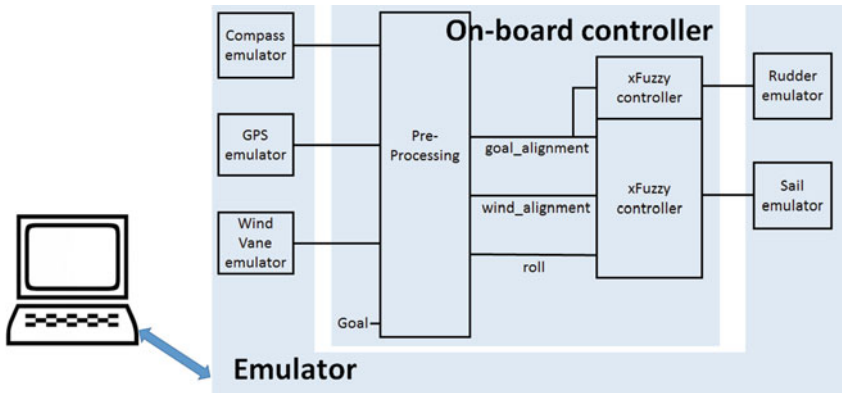


Fig. 9 Emulator framework

- to support laboratory development, where the emulator replaces all the instrumentation devices and communicate with a working station, which can force sequence of values at on-board controller inputs and capture controller’s response. This is illustrated in Fig. 9;
- to support data acquisition and control in real navigation situations, where the emulator is connected through a wireless connection to a base station (currently, this option is still work in progress).

For that end, the different interface protocols in use were successfully emulated, being the emulation of the NMEA 0183 - Standard for interfacing marine electronic devices protocol used for GPS receiver [12, 13] the most challenging one (NMEA stands for National Marine Electronics Association).

Some applications and tools are already available to assist autonomous sailboats or other autonomous vehicles. Among them, special mention to METASail [14] and to Neptus [15].

METASail [14] is an application to plan, analyze and supervise the FAST Project [3]. It offers a mission planning environment, where all the path and strategy can be created interactively, by dragging and dropping way-points and assigning or editing actions to them, or even by loading a file with all the instructions. Simulating missions is another functionality of this tool. The tool is also used as an interface to operate and remotely supervise the evolution of the mission, in real operation or simulated.

Neptus [15] was developed within the scope of a network of vehicle systems. The network is composed by human operators, heterogeneous autonomous vehicles and other sensing devices. Neptus platform is used to plan, simulate, monitor and analyze executed missions.

### 4.2 Introducing an Anemometer

The lack of information about sailboat speed, as well as wind speed and direction, severely constraint the development of advanced control strategies. To mitigate these weaknesses an integrated new sensor is under development. The sensor integrates a homemade anemometer, based on ultrasonic sensors, with GPS and compass instruments, where the anemometer can provide information on wind speed and direction, and GPS and compass can provide information on sailboat speed and direction. Combining both information, one can conclude on true and apparent wind, which is of paramount importance to support advanced sailing control strategies.

The anemometer is currently in final validation phase, using a wind tunnel to calibrate the measurements and allow their use in the sailboat, and will integrate the wind vane previously in use (allowing integration of information from both devices). Preliminary results are very encouraging in terms of accuracy, as well as in terms of cost, where an Arduino Nano is used to assure dedicated computing power.

The ultrasonic based anemometer operates on the measurement of the time of flight of the ultrasonic wave between two ultrasonic transducers separated by approximately 20 cm, as in [16]. The time of flight will depend on the wind speed and direction.

Two pairs of ultrasonic transducers are used, using orthogonal directions, which will allow to determine not only the wind speed as well the wind direction, as in [17]. The structure to support the transducers is lightweight in order not to compromise stability and aerodynamics.

In this sense, it is possible to improve the lower layer of the previously presented controller, associated with the sail and rudder fuzzy controllers, as presented in Fig. 10.

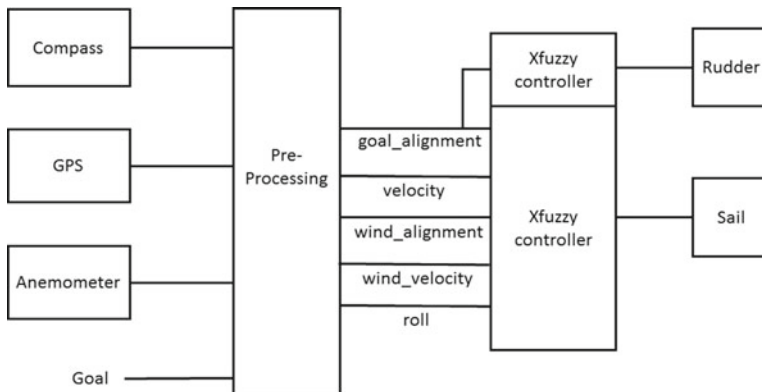


Fig. 10 Block diagram of the new low level controller

## 5 Conclusions and Future Work

The presented work aims to contribute to have an open platform for autonomous sailboat navigation experimentation, with hardware and software open to the community, as well as to validate model-based development approach to support associated controllers development. In the proposed controller development, two freely available tools frameworks (Xfuzzy and IOPT-Tools) were used allowing automatic C code generation, ready to be deployed in implementation platforms, as Arduino boards. In this sense, development activities will intrinsically take advantage of the usage of simulators and automatic code generators, having a strong impact both at the development time, as well as on the robustness of the solution.

Improvements on electronic instrumentation will continue in order to support more flexibility and accurate control, allowing detection of adverse navigation situations, as well as obstacle detection, and adoption of countermeasures.

Several types of activities have been carried on based on the development of the sailboat controller, namely as support for several MSc Thesis works, and more remarkably being the subject to offer several short Summer courses (one week long) for high school students, as presented in [18], supported by Portuguese Agency Ciência Viva (editions on July 2014, 2015, and 2016). The main goal of the courses is to attract high-school students to sciences and engineering, introducing them to research and development activities within STEM (science, technology, engineering and mathematics) areas, through some exploratory and demonstration activities. Autonomous sailboats control has proved to be an ideal topic to reach this objective.

Future works include the addition of wireless communication allowing exchange of information and collaboration with other sailboats as well as other external agents, within a cyber-physical systems interaction framework.

**Acknowledgments** This work was partially financed by Portuguese Agency FCT - Fundação para a Ciência e Tecnologia, in the framework of project UID/EEA/00066/2013.

## References

1. Arduino home page. <http://www.arduino.cc/>
2. Alves, J.C., Cruz, N.A.: An FPGA-based embedded system for a sailing robot. In: Proceedings of the EUROMICRO Conference on Digital System Design, Patras, Greece, August 2009 (2009)
3. Alves, J., Cruz, N.: Fast - an autonomous sailing platform for oceanographic missions. In: Proceedings of the MTS-IEEE Conference Oceans' 2008, Quebec, Canada, September 2008. IEEE (2008)
4. Gomes, L., Santos, M., Pereira, T., Costa, A.: Model-based development of an autonomous sailing Yacht controller. In: 2015 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Vila Real, 2015, pp. 103–108 (2015). doi:[10.1109/ICARSC.2015.20](https://doi.org/10.1109/ICARSC.2015.20)

5. Santos, D., Silva Junior, A., Negreiros, A., Vilas Boas, J., Alvarez, J., Araujo, A., Aroca, R., Gonçalves, L.: Design and implementation of a control system for a sailboat robot. In: *Robotics*, vol. 5(1) (2016). doi:[10.3390/robotics5010005](https://doi.org/10.3390/robotics5010005)
6. Stelzer, R., Proll, T., John, R.I.: Fuzzy logic control system for autonomous sailboats. In: *2007 IEEE International Fuzzy Systems Conference (2007)*. doi:[10.1109/FUZZY.2007.4295347](https://doi.org/10.1109/FUZZY.2007.4295347)
7. Pererira, F., Moutinho, F., Gomes, L.: IOPT-Tools - Towards cloud design automation of digital controllers with Petri nets. In: *2014 International Conference on Mechatronics and Control (ICMC) (2014)*
8. Gomes, L., Barros, J., Costa, A., Nunes, R.: The input-output place-transition petri net class and associated tools. In: *2007 5th IEEE International Conference on Industrial Informatics, June 2007*, vol. 1, pp. 509–514 (2007)
9. Fuzzy logic design tools. [www2.imse-cnm.csic.es/Xfuzzy/](http://www2.imse-cnm.csic.es/Xfuzzy/)
10. Sanchez-Solano, S., del Toro, E., Brox, M., Brox, P., Baturone, I.: Model-based design methodology for rapid development of fuzzy controllers on FPGAs. *IEEE Trans. Indus. Inf.* **9**, 1361–1370 (2012)
11. Brox, M., Sanchez-Solano, S., del Toro, E., Brox, P., Moreno-Velo, F.J.: Cad tools for hardware implementation of embedded fuzzy systems on FPGAs. *IEEE Trans. Indus. Inf.* **60**, 3182–3194 (2012)
12. NMEA, N.: NMEA 0183 Version 4.00 Standard for interfacing marine electronic devices (2008)
13. Langley, R.B.: NMEA 0183: a GPS receiver interface standard. In: *GPS World*, July 1995, pp. 54–57 (1995)
14. Alves, J.C., Cruz, N.A.: METASail a tool for planning, supervision and analysis of robotic sailboat missions. In: *7th International Robotic Sailing Conference Robotic Sailing 2014*, pp. 57–64 (2014)
15. Pinto, J., Calado, P., Braga, J., Dias, P., Martins, R., Marques, E., Sousa, J.B.: Implementation of a control architecture for networked vehicle systems. In: *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*, pp. 100–105 (2012)
16. Dong, H., Jun, Y.: High accuracy time of flight measurement for ultrasonic anemometer applications. In: *2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC) (2013)*. doi:[10.1109/IMCCC.2013.21](https://doi.org/10.1109/IMCCC.2013.21)
17. Xie, K., Wang, K.: Measurement of wind speed and direction with ultrasonic sensor using FPGA. In: *Energy Procedia*, vol.12, pp.837–843 (2011). doi:[10.1016/j.egypro.2011.10.110](https://doi.org/10.1016/j.egypro.2011.10.110)
18. Gomes, L., Costa, A., Moutinho, F., Mota, R.: Attracting students to engineering through autonomous sailing yacht development. In: *ICIT2015 2015 IEEE International Conference on Industrial Technology (2015)*