IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 7 December 2022, accepted 23 December 2022, date of publication 26 December 2022,
date of current version 29 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3232462

RESEARCH ARTICLE

Experimental Studies of Autonomous Sailing With a Radio Controlled Sailboat

AUGUSTIN MORGE[1], VIRGILE PELLE[1], JIAN WAN[2], AND LUC JAULIN[3]
[1]ENSTA Bretagne, 29806 Brest, France
[2]Department of Mechanical, Biomedical and Design Engineering, School of Engineering and Technology, College of Engineering and Physical Sciences, Aston University, B4 7ET Birmingham, U.K.
[3]Lab-STICC, ENSTA Bretagne, 29806 Brest, France

Corresponding author: Jian Wan (j.wan3@aston.ac.uk)

This work was supported in part by the start-up grant of Aston University.

ABSTRACT Autonomous sailing has attracted many interests from both industry and academy due to its great potential in oceanography, research and education. Worldwide researchers have developed various kinds of unmanned sailboat platforms for their specific research purposes and applications. However, most of these autonomous sailing platforms are rather complex to build and to program. The aim of this study is to propose, to build, and to test a new compact sailboat platform using a unique combination of a 1-m class radio controlled model sailboat, a Raspberry Pi 4, a Navio2, and a Calypso ultrasonic solar-powered anemometer. The proposed new sailboat platform makes full use of the agility of the racing model boat with minimal mechanical adjustments as the system does not use electrical wires to connect with additional sensors and a mechanical anemometer. The software architecture of the proposed new sailboat platform is also simplified as it involves only Python programming for the Raspberry Pi 4. The experimental results have demonstrated the feasibility as well as the functionality of the proposed new sailboat platform to conduct autonomous sailing operations successfully and it could become one of the most accessible, generic and flexible autonomous sailing platforms for advanced research and education in robotics, autonomous systems, and artificial intelligence.

INDEX TERMS Autonomous sailing, line following, position keeping, python programming.

I. INTRODUCTION

Robotics and autonomous systems as an industrial and commercial activity has its strong economic contribution and disruptive socioeconomic impact across diverse market sectors worldwide [1]. The rapid development and commercial successes of the drone industry can be an evidence to such an impact. Considering that water covers about 71% of the Earth's surface and the majority of the ocean surface is yet to be explored or mapped, marine robotics has attracted more and more interest from both industry and academy [2]. Various marine robots have already been developed to cover applications in both surface and underwater environments. Among all these marine robots, autonomous sailboats become a significant branch of focus due to its great potential as a green and sustainable platform for long-duration missions such as oceanography. Autonomous sailboats can be of monohull, catamaran, and trimaran type with a rigid or a flexible sail [3]. The dimension of autonomous sailboats can also range from less than 1 meter in length to tens of meters long costing hundreds of thousands dollars to build.

Autonomous sailboats rely on the relevant sensors, actuators and controllers to implement autonomous sailing operations. Typically an autonomous sailboat needs a GPS receiver to provide its position for the navigation, an anemometer to provide the direction and the speed of the wind, and a compass to provide the heading of the sailboat [4]. The sailboat also needs the controllers to process sensor data and to provide the computed instructions to adjust the angle of the rudder and the angle of the sail, respectively. The angle adjustments are usually conducted by the corresponding servo motors that are linked with the rudder and the sail. There are also auxiliary sensors within autonomous sailboats such as communication modules for the purpose of remote

The associate editor coordinating the review of this manuscript and approving it for publication was Bidyadhar Subudhi.

control and online monitoring [5]. Depending on the budget and the dimension of the autonomous sailboat, there are many combinations of these sensors, actuators and controllers in the literature [6]. Early autonomous sailboat platforms usually use a mechanical anemometer that can provide the direction and the speed of the wind [7]. Ultrasonic wind sensors are increasingly used in autonomous sailboats due to its higher resolution and greater reliability with no moving parts [8]. The controllers used in autonomous sailboats can be Arduino-based microcontrollers for easy access of sensor data or Raspberry Pi-based computers to implement Robot Operating System (ROS) based software architecture for obstacle avoidance and computer vision. Arduino-based microcontrollers and Raspberry Pi-based computers can also be combined together to make full use of their individual benefits as in [7]. In order to avoid ROS-based programming, which is usually complex in terms of programming, the autonomous sailboat that was developed in [2] used a Pixhawk V2.4.8 along with a Arduino Mega 2560 as the controller where the Pixhaw V2.4.8 can provide reliable GPS and compass states. A rather complex control system involving a NVIDIA Xavier, a Raspberry Pi with an autopilot hat named Navio2, and a Pixhawk is proposed in [9] for better energy estimation and management. Various autonomous sailing algorithms can then be tested on these developed control systems and the World Robotic Sailing Championships (WRSC) also provides the channel and the stimulus for testing and comparing these autonomous sailing algorithms. Traditional line following and station keeping algorithms can be tested on an Arduino-based control system while obstacle avoidance sailing algorithms can be tested on a Raspberry Pi-based control system with computer vision. Advanced sailing algorithms such as machine learning-based sailing can also be tested on a Raspberry Pi-based or a NVIDIA-based control system [10].

The main challenge or hurdle to conduct autonomous sailing research is to have an agile and compact platform in terms of hardware and software configurations since most of existing sailboat platforms are rather complex to build and to program. This is mainly due to the existing combinations of sensors and microcontrollers are often hybrid involving multiple types of microcontrollers and programming languages. Focusing on the core need of testing autonomous sailing algorithms, radio controlled (RC) model sailboats can provide a credible test platform since nowadays model sailboats are very agile. This is because many competitive sailing events for RC model boat racing has helped to elevate boat design and sailing techniques into new levels. To make full use of the agility of modern model sailboats, minimal mechanical adjustments are needed to convert a model sailboat into an autonomous sailboat. Taking advantage of the latest sensor and Artificial Intelligence (AI) computer developments, particularly the availability of miniature ultrasonic anemometers and Raspberry Pi 4, this paper aims to propose a new autonomous sailboat platform with a unique combination of a Raspberry Pi 4 with Navio2 and a Calypso solar-powered ultrasonic anemometer [11] on a 1-meter class RC model boat and to conduct autonomous sailing experiments on the proposed sailboat platform. The benefits of such a combination include minimal mechanical adjustments of the original model sailboat and a software architecture involving only Python programming. As the Calypso anemometer communicates with the Raspberry Pi 4 via Bluetooth, the sailboat platform also removes extra electrical wires widely seen in other sailing platforms. The platform can also be extended to include computer vision and machine learning algorithms by making full use of Python-supported OpenCV and deep learning packages. Since autonomous sailboats are increasingly used for research and education as in [12], it is expected that the proposed autonomous sailboat platform can become an easily accessible and fast-prototyped platform for advanced research and education in robotics and machine learning as well.

The rest of the paper is organised as follows. The principle of autonomous sailing and two typical sailing algorithms to be tested on the proposed sailboat platform are introduced in Section II. The experimental setup for the proposed sailboat platform is detailed in Section III, which includes the hardware configuration, the software architecture, the online monitoring module, and the assembled platform. The experimental results of using the developed sailboat platform to test autonomous sailing algorithms at a local lake are provided in Section IV. Some conclusions are drawn in Section V to summarize the obtained experimental results and the potential future work on the developed sailboat platform.

II. AUTONOMOUS SAILING

Autonomous sailing means that there is no human intervention for the sailing operations. The principle of autonomous sailing is shown in Figure 1. It can be seen that a control task is given to the controller inside the sailboat and then the controller computes a desired angle of rudder δ_r and also a desired angle of sail δ_s for rudder and sail servos to act on the basis of the GPS position of the sailboat $\mathbf{m}$, the direction of the true wind ψ_{tw}, the speed of the true wind a_{tw}, and also the orientation of the boat θ. The velocity of the sailboat is denoted as v and the corresponding course angle for this sailboat speed is denoted as ϕ. Due to the sideway forces of the wind, the course angle ϕ and the heading angle θ are not necessarily equal [14]. It is worthy to note that $\mathbf{m}$, ψ_{tw}, a_{tw}, θ, and ϕ are defined in a East-North-Up (ENU) coordinate system with its origin fixed on an Earth point while δ_r and δ_s are given in a body-fixed coordinate system with its origin fixed on the gravity center of the boat. These two coordinate systems are shown in Figure 2, where the direction of the true wind is denoted as ψ_{tw}, the speed of the true wind is denoted as a_{tw}, the direction of the apparent wind is denoted as ψ_{aw}, and the speed of the apparent wind is denoted as a_{aw}.

The position of the sailboat $\mathbf{m}$ can be measured by a GPS receiver and usually the GPS receiver can also provide the velocity v and the course angle ϕ of the sailboat. The orientation of the sailboat θ is usually measured by a compass.

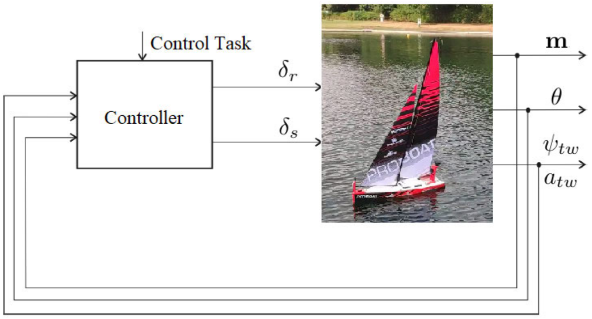

FIGURE 1. The principle of autonomous sailing [4].

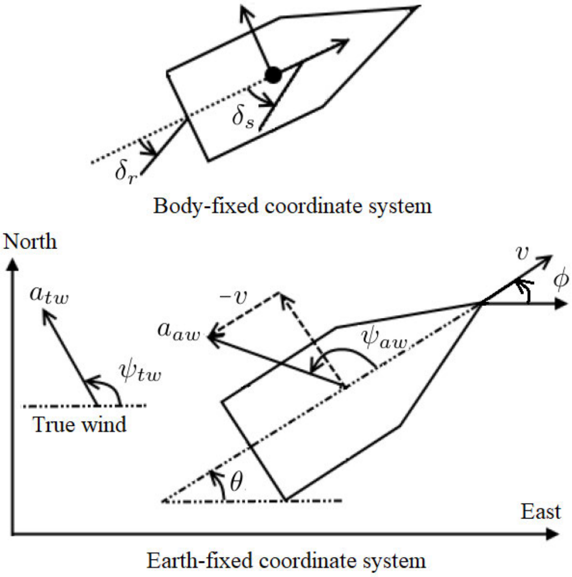

FIGURE 2. The coordinate systems and the true wind [13].

An anemometer fixed on the sailboat can measure ψ_{aw} and a_{aw}, respectively. The relationship between the apparent wind and the true wind is also shown in Figure 2, where the direction and the speed of the true wind can be calculated from those of the apparent wind as follows:

$$\begin{cases} a_{tw} = \|T_w\| \\ \psi_{tw} = \operatorname{atan2}(T_w) \end{cases} \tag{1}$$

where

$$T_w = \begin{bmatrix} v\cos(\phi) + a_{aw}\cos(\psi_{aw} + \theta) \\ v\sin(\phi) + a_{aw}\sin(\psi_{aw} + \theta) \end{bmatrix} \tag{2}$$

and T_w is the Cartesian coordinate of the true wind in the ENU coordinate system [13].

Autonomous sailboats are usually required to arrive to certain location and to stay there for certain time to take some measurements or exchange information with other facilities. Thus typical sailing operations for an autonomous sailboat are line following and station keeping. In the following subsections, a line following algorithm and a station keeping algorithm are introduced and simulated. The notations used for these two sailing algorithms are listed in Table 1.

TABLE 1. Notations.

$\mathbf{m}$	position of the sailboat
θ	orientation of the sailboat
v	velocity of the sailboat
ϕ	course angle of the sailboat
δ_r	angle of the rudder, $\lvert\delta_r\rvert \le \delta_r^{\max}$
δ_s	angle of the sail, $\lvert\delta_s\rvert \le \delta_s^{\max}$
ψ_{tw}, a_{tw}	direction and speed of the true wind
ψ_{aw}, a_{aw}	direction and speed of the apparent wind
$\mathbf{a}, \mathbf{b}$	the starting and the ending points of a specified line
ξ	the close haul angle
r	the cutting distance to a line
β	the angle of the sail in crosswind
q	a binary variable $q \in \{-1, 1\}$ for tacking
d_i	the radius of the inner circle for station keeping
d_o	the radius of the outer circle for station keeping

A. LINE FOLLOWING

Line following is the most basic operation for an autonomous sailboat as it enables the sailboat to reach another destination in a straightforward way. Here the line following algorithm to be tested is taken from [15], where the inputs of the algorithm include the current position of the sailboat $\mathbf{m}$, the heading θ, the true wind direction ψ_{tw}, two points $\mathbf{a}$ and $\mathbf{b}$ defining the line to be followed, and a tacking variable $q \in \{-1, 1\}$ indicating the status of tacking. The key for the line following algorithm is to compute a desired heading $\hat{\theta}$ of the sailboat depending on its distance to the line, the nominal heading τ, and the true wind direction ψ_{tw}. The corresponding ruder angle δ_r and the sail angle δ_s can then be computed on the basis of this desired heading $\hat{\theta}$. The details for the line following algorithm is provided in Table 2, where Step 1 is to calculate the algebraic distance between the sailboat and the line to be followed with e > 0 indicating the sailboat on the left of the line and e < 0 indicating the sailboat on the right of the line; Step 2 is to update the tacking variable q as the sailboat is only allowed to change tacking when $|e| > r$; Step 3 is to calculate the line angle τ; Step 4 is to calculate the desired heading $\hat{\theta}$ based on the line angle; Step 5 is to update the desired heading to be close hauled heading in case of unfeasible directions in the *no-go* zone; Step 6 is to update the rudder angle based on the desired heading $\hat{\theta}$; and Step 7 is to update the sail angle based on the desired heading $\hat{\theta}$.

Line following can be fully tested by a triangle racing task since the wind conditions for these three lines from the triangle racing task are different and the sailboat needs to change the status of tacking to fulfil this task. Using the mathematical model of the sailboat in [13] and the line following algorithm in Table 2, a simulation is conducted for a triangle racing task via line following with a constant wind direction of 270^0. The simulation result is shown in Figure 3, where the three points are at (0,0), (200,0), and (180,180), respectively. It can be seen that the line following algorithm is working perfectly by following these three lines sequentially with a tacking

TABLE 2. The controller for line following.

```
Line following algorithm [15]:
inputs: $\mathbf{m}, \theta, \psi_{tw}, \mathbf{a}, \mathbf{b}, q$
1. e = det($\frac{\mathbf{b}-\mathbf{a}}{||\mathbf{b}-\mathbf{a}||}, \mathbf{m} - \mathbf{a}$)
2. if $|e| > r$, then $q = sign(e)$
3. $\tau = angle(\mathbf{b} - \mathbf{a})$
4. $\hat{\theta} = \tau - \arctan(e/r)$
5. if $(\cos(\psi_{tw} - \hat{\theta}) + \cos(\xi) < 0)$
       or $((|e| - r)$ and $(\cos(\psi_{tw} - \tau) + \cos(\xi) < 0))$
   then $\hat{\theta} = \pi + \psi_{tw} - q * \xi$
6. $\delta_r = \frac{\delta_r^{\max}}{\pi} * sawtooth(\theta - \hat{\theta})$
7. $|\delta_s| = \frac{\pi}{4} * (\cos(\psi_{tw} - \hat{\theta}) + 1)^{\frac{\log(\frac{\pi}{2\beta})}{\log(2)}}$
outputs: $\delta_r, \delta_s, q$
```

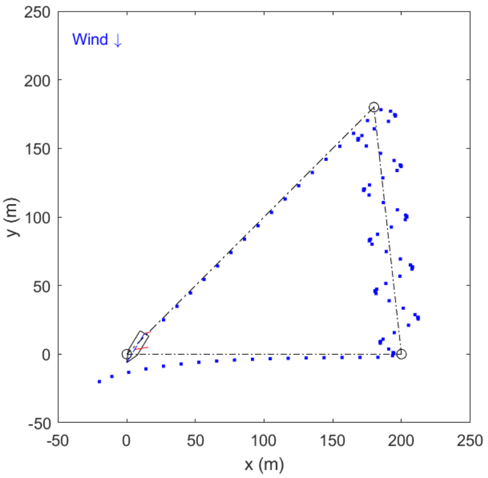

FIGURE 3. Triangle racing via line following.

strategy implemented automatically in case of sailing against the wind. A similar triangle racing task via line following is to be tested on the proposed sailboat platform to verify its functionality in the following experimental studies.

B. STATION KEEPING

Station keeping requires the sailboat to reach and to maintain at a specific position under any wind condition. Many control methods can be used to control the rudder and the sail during station keeping as the one proposed in [16]. An advanced station keeping algorithm was studied in [17] and this method does not use any prior knowledge on the dynamic model of the sailboat. A simplified version of this station keeping algorithm is adopted for the experimental studies. The simplification is mainly on the sail control as we use the same sail control to the line following algorithm. This is because the rudder control plays a bigger role in station keeping while the sail control is auxiliary. The principle of station keeping is similar to line following, which is to compute a desired heading $\hat{\theta}$ depending on the circumstances of the sailboat. The corresponding ruder angle δ_r and the sail angle δ_s are to be computed on the basis of this desired heading $\hat{\theta}$. The process of the method is divided into three steps depending on the sailboat's distance to the target point, where the distance is denoted as d_t and the target point is denoted as **b**. An inner circle with a radius of d_i and an outer circle with a radius of d_o are defined to quantify this distance to the target point. The detailed station keeping algorithm for the experimental studies is listed in Table 3, where Step 1 is to use the line following algorithm to reach the target when $d_t > d_o$; Step 2 is to compute the corresponding desired heading $\hat{\theta}$ when the sailboat enters into the outer circle, i.e., $d_i < d_t > d_o$; Step 3 is to compute the corresponding desired heading $\hat{\theta}$ when the sailboat enters into the inner circle, i.e., $d_t < d_i$; Step 4 is to compute the corresponding ruddle angle δ_r and the sail angle δ_s according to the desired heading $\hat{\theta}$.

TABLE 3. The controller for station keeping.

```
Station keeping algorithm:
inputs: $\mathbf{m}, \theta, \psi_{tw}, \mathbf{b}, q$
1. If $d_t > d_o$, compute $\delta_r$ and $\delta_s$ via line following
2. if $d_i < d_t > d_o$, compute $\delta_r$ and $\delta_s$ as follows:
        if $cos(\tau - (\psi_{tw} - \frac{\pi}{2})) > 0$
            if $cos(\alpha - (\psi + \frac{\pi}{2})) > cos(\alpha - (\psi - \frac{\pi}{2}))$
                $\theta^* = \theta^* + \frac{\pi}{2}$
            else
                $\theta^* = \theta^* + \frac{\pi}{2}$
        if $\cos(\psi_{tw} - \theta^* + \cos(\xi) > 0$
            if $\sin(\psi - \theta^*) > 0$
                q=-1
            else
                q=1
            $\hat{\theta} = \psi_{tw} + pi + q\xi$
        else
            $\hat{\theta} = \theta^*$
3. if $d_t < d_i$, compute $\delta_r$ and $\delta_s$ as follows:
        if $\cos(\psi_{tw} - \theta^* + \cos(\xi) > 0$
            if $\cos(\theta^* - (\psi_{tw} - \xi)) > \cos(\theta^* - (\psi_{tw} + \xi))$
                q=-1
            else
                q=1
            $\hat{\theta} = \psi_{tw} + pi + q\xi$
        else
            $\hat{\theta} = \theta^*$
4. if $\cos(\theta - \hat{\theta}) > 0$
      $\delta_r = \delta_r^{\max} \sin(\theta - \hat{\theta})$
   else
      $\delta_r = \delta_r^{\max} sign(\sin(\theta - \hat{\theta}))$
   $|\delta_s| = \frac{\pi}{4} * (\cos(\psi_{tw} - \hat{\theta}) + 1)$
outputs : $\delta_r, \delta_s$ and $q$
```

In order to test and demonstrate this simplified station keeping algorithm, a simulation study has also been conducted by using the mathematical model of the sailboat in [13]. The simulation result is shown in Figure 4, where the target point is at (0,10) with $d_i = 7$ m, $d_o = 14$ m, and a constant wind direction of 90^0. It can be seen that the sailboat can arrive to the target point by sailing against the wind in Step 2 and to maintain inside the inner circle in Step 3. This station keeping algorithm is also to be tested on the proposed sailboat platform in the following experimental studies.

III. EXPERIMENTAL SETUP

The two autonomous sailing algorithms introduced in Section II are to be tested on the proposed sailboat platform

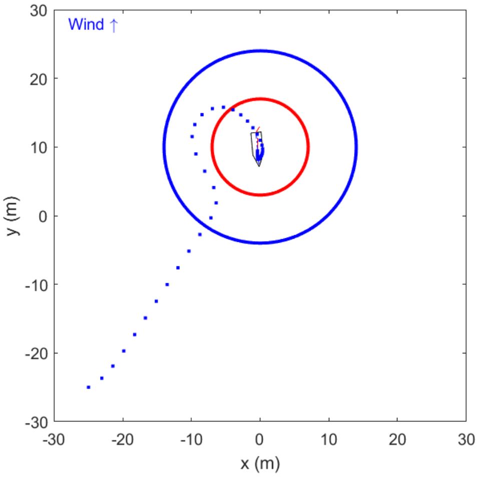

FIGURE 4. The simulation for station keeping.

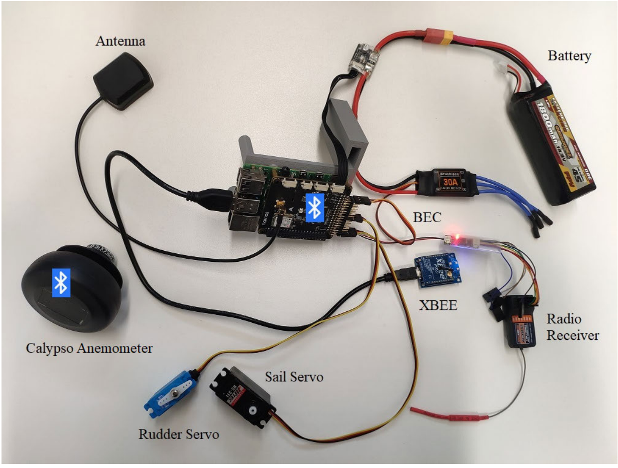

FIGURE 5. The hardware configuration.

for verifying its feasibility and functionality. The following subsections describe the detailed hardware configuration, software architecture, online monitoring, and the assembled sailboat of the proposed sailboat platform, respectively.

A. HARDWARE CONFIGURATION

The hardware for the control system is configured around a Raspberry Pi 4 along with an autopilot hat named Navio2 for navigation and control. Navio2 has two embedded inertial measurement unit (IMU) sensors: MPU9250 and LSM9DS1. They provide the head of the sailboat θ for the sailing algorithms. Navio2 has also an embedded GNSS receiver that provides the GPS location of the sailboat $\mathbf{m}$ with the NAV_POSLLH message and also the speed of the sailboat v with the NAV_VELNED message. A Calypso ultrasonic solar-powered anemometer is connected to Raspberry Pi 4 via Bluetooth and it provides the direction ψ_{aw} and the speed a_{aw} of the apparent wind. Navio2 also has the servo trail that connects with the rudder servo and the sail servo, respectively. The whole system can be powered by a single Li-Po battery. A radio receiver and a XBEE module can be connected to the system to provide the function of data communication and remote control. Figure 5 shows the connected hardware for the configured control system.

B. SOFTWARE ARCHITECTURE

As the configured hardware system has only three main parts (a Raspberry Pi 4, a Navio2, and a Calypso anemometer), the corresponding software architecture is relatively compact as well: each sensor has its own program; an extra program is to receive the raw sensor data and to filter them into a state vector in thread; and then the main program receives this state vector and provides the corresponding instructions to control the rudder servo and the sail servo according to the sailing algorithm to be tested. The software architecture is shown in Figure 6. It is worthy to note that all the programs are written by Python without the use of any ROS command.

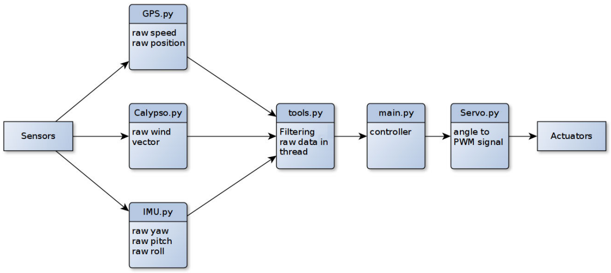

FIGURE 6. The software architecture.

The software architecture can also be extended to include computer vision and other machine learning packages.

C. ONLINE MONITORING

In order to monitor the status of the sailboat on a nearby laptop, a XBEE module is connected to Raspberry Pi 4 and the other XBEE module is connected to the laptop. These two XBEE modules are configured to use the same channel to establish a communication between them. The position of the sailboat, the direction of the true wind, the heading of the sailboat, and other relevant information can be transmitted to the monitoring laptop. The monitoring laptop can also take over the control of the sailboat by sending new instructions to update the rudder angle and the sail angle via XBEE. In order to display the movements of the sailboat on a Google map in real time, a server is set up on the monitoring laptop. The procedures to set up this server is shown in Figure 7, where a Python script receives the data sending from the XBEE module and puts them into a MYSQL database; a php file gets the stored data and send them to a JavaScript function; then a HTML file uses this function to display the real-time status of the sailboat on the Google map.

D. SAILBOAT PLATFORM

A 1-meter class model sailboat from Proboat$^{\text{TM}}$ is used to host the configured hardware and to conduct the experimental

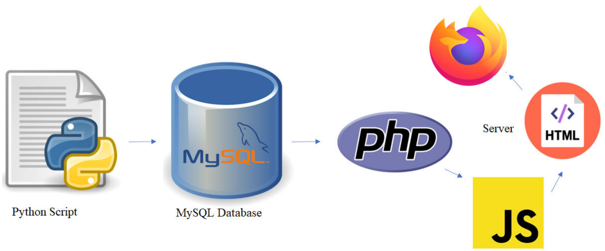

FIGURE 7. The setup of the server for online monitoring.

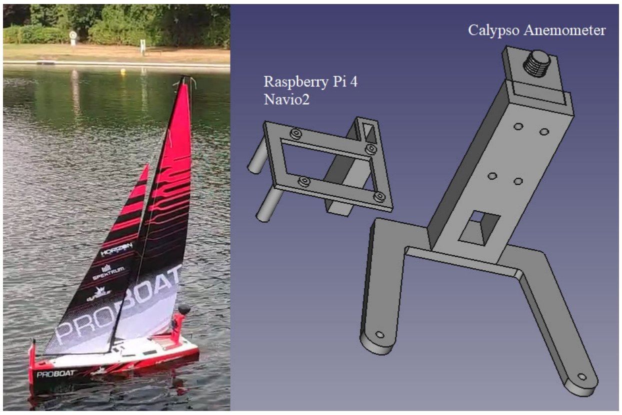

FIGURE 8. The RC sailboat with minimal mechanical adjustments.

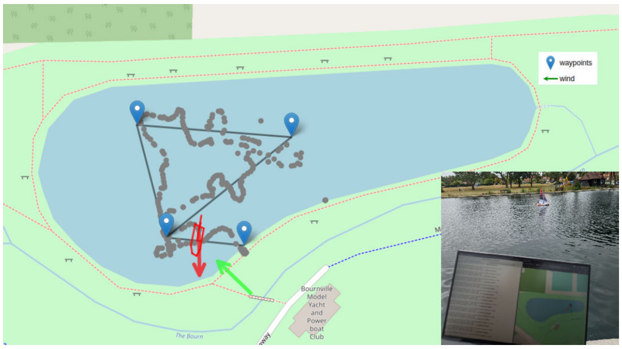

FIGURE 9. Triangle racing test via line following.

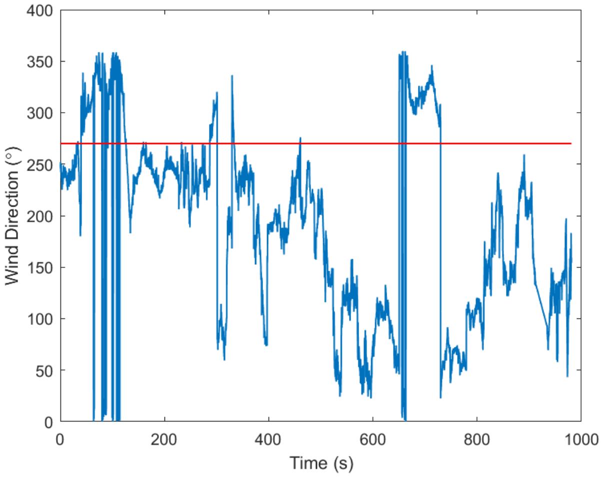

FIGURE 10. The wind direction trajectory during the triangle racing test.

studies. Since the hardware is rather compact, it is straightforward to put the Raspberry Pi 4 and Navio2 into the hatch of the sailboat while the antenna for the GNSS receiver and the Calypso anemometer are to be placed outside the hatch. Two mechanical support systems are 3D printed to hold the Raspberry Pi 4 and the Calypso anemometer into their positions, respectively. The assembled sailboat platform with all the hardware onboard is shown in Figure 8. It can be seen that there are minimal mechanical adjustments to the original RC model sailboat, which could help to maintain its original agility. An additional camera support is also added in the front of the sailboat and thus the converted autonomous sailboat can also implement the function of computer vision. The total cost for building the proposed sailboat platform can be around €1,000 including the model sailboat. Overall, the proposed sailboat platform is rather compact in terms of both hardware and software, which is advantageous for the purpose of fast prototyping and testing of autonomous sailing algorithms.

IV. EXPERIMENTAL RESULTS

The experimental studies were conducted at a local concrete-floored lake using the developed new sailboat platform. The first test was the triangle racing via the line following algorithm. The test results are shown in Figure 9. It can be seen that the monitoring laptop can display the movements of the sailboat in real time and the sailboat can indeed finish the triangle racing by roughly following three lines sequentially. However, it is worthy to note that the experimental results are not as good as the simulated results in Figure 3. This is mainly due to the constant changes of the wind direction and the wind speed during this experiment while the wind direction and the wind speed are kept constant for the simulation. The recorded wind direction trajectory during the triangle racing test is shown in Figure 10, where the red line is the wind direction used in the simulation while the blue lines are the wind direction trajectory experienced in the test. The lake for the test is also relatively small with about 50 m in width. Nevertheless, the robustness of the current line following algorithm needs to be improved to cope with the constant changes of wind direction and wind speed. More measures such as the filtering of wind direction data are to be taken in the future experimental studies.

The second test was for the station keeping algorithm along the wind. The radius of the inner circle is $d_i = 1.5$ m and the radius of the outer circle is $d_o = 3$ m. The monitored results are shown in Figure 11 while the station keeping movements inside the outer circle are shown in Figure 12. It can be seen that the sailboat did sail along the wind initially via line following, arrived to the inner circle, and the stayed within the outer circle for 87 seconds. It is more difficult to maintain the sailboat within the inner circle than in the simulation due to the complex environmental conditions encountered

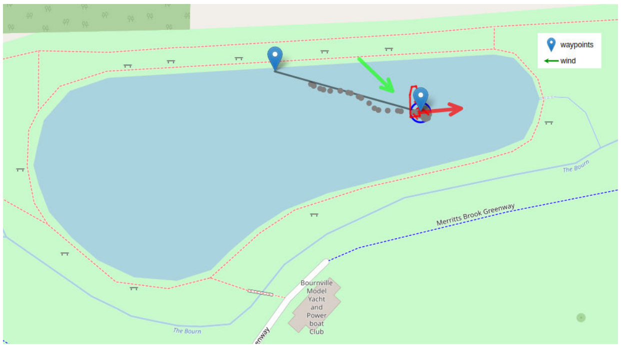

FIGURE 11. Station keeping test along the wind.

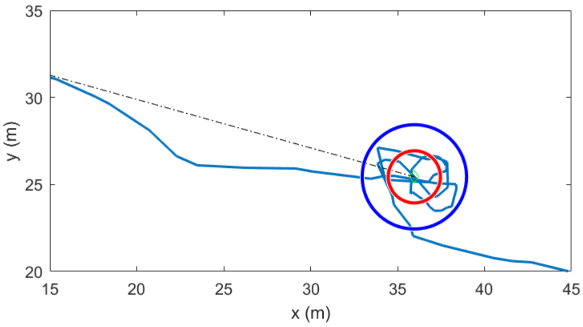

FIGURE 12. Station keeping movements within the circles (along the wind).

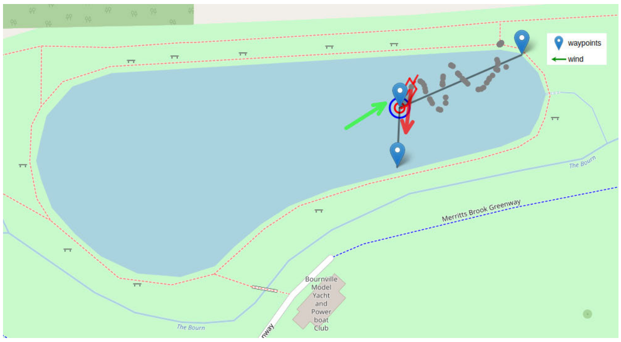

FIGURE 13. Station keeping test against the wind.

in the test. It is also worthy to explore and develop robust station keeping algorithms to cope with all these changing environmental conditions.

The third test was for the station keeping against the wind. The radius of the inner circle is $d_i = 1.5$ m and the radius of the outer circle is $d_o = 3$ m as well. The monitored results are shown in Figure 13 while the station keeping movements inside the outer circle are shown in Figure 14. The sailboat did use a tacking strategy to sail against the wind initially via line following, arrived into the inner circle, and stayed within the outer circle for 60 seconds. This third test further confirms the functionality of the proposed sailboat platform to implement autonomous sailing algorithms with the benefits of simpler hardware and software configurations.

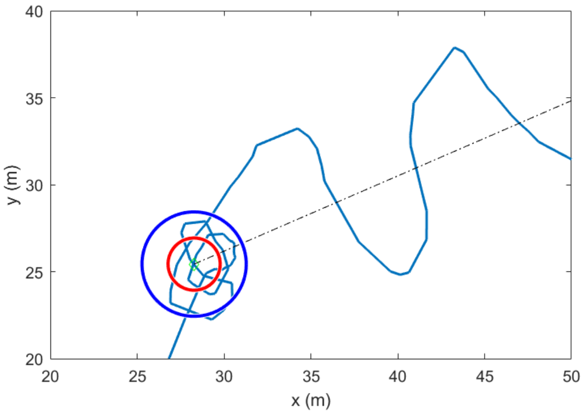

FIGURE 14. Station keeping movements within the circles (against the wind).

Based on the above three experimental results, it is fair to say that the selected combination of the 1-meter RC sailboat, the Raspberry Pi 4, the Navio2, and the Calypso anemometer meets all the needs to conduct experimental studies of autonomous sailing straightforwardly, which is advantageous than other complex and expensive sailboat platforms.

V. CONCLUSION

A new autonomous sailboat platform has been built and tested to meet the need of an easily-accessible, fast-prototyped and economic platform in autonomous sailing research. The developed sailboat platform is composed of a 1-meter class RC sailboat, a Raspberry Pi 4, a Navio2, and a Calypso ultrasonic anemometer in hardware and it uses only Python programming in software. Such a unique combination of hardware and software makes the platform very compact and agile. Experimental studies on the proposed sailboat platform have demonstrated its full functionality to implement typical autonomous sailing algorithms such as line following and station keeping. The developed sailboat platform can also be extended to include the functionality of computer vision and machine learning. More studies are needed to improve the robustness of the tested sailing algorithms to cope with changing environmental conditions and noisy sensor data encountered in the experimental studies. The developed new sailboat platform is also to be used for testing other advanced autonomous sailing algorithms in the future work.

ACKNOWLEDGMENT

The authors would like to thank extra experimental assistance from the other two internship students named Laurent Droudun and Mathieu Drillon, also would like to thank Dr. J. Ricardo Sodré (the Head of the Department of Mechanical, Biomedical and Design Engineering) to approve the internship, Prof. Stephen Garrett (the Pro-Vice Chancellor and the Executive Dean of the College of Engineering and Physical Sciences) to sign the host agreement with ENSTA Bretagne, France, and the Research Support Team led by Dr. Emma Estevez to provide administrative support throughout this internship, from May 2022 to August 2022, and also would like to thank anonymous reviewers for their constructive comments to improve the paper greatly. This research work was the research outcome of several summer internship students' work at Aston University, in 2022.

REFERENCES

[1] *Rolling Plan 2021 for ICT Standardisation: Robotics and Autonomous Systems*. Accessed: Oct. 17, 2022. [Online]. Available: https://joinup.ec.europa.eu/collection/rolling-plan-ict-standardisation/robotics-and-autonomous-systems

[2] S. Yang, C. Liu, Y. Liu, J. An, and X. Xiang, "Generic and flexible unmanned sailboat for innovative education and world robotic sailing championship," *Frontiers Robot. AI*, vol. 8, Mar. 2021, Art. no. 630081, doi: 10.3389/frobt.2021.630081.

[3] Y. An, J. Yu, and J. Zhang, "Autonomous sailboat design: A review from the performance perspective," *Ocean Eng.*, vol. 238, Oct. 2021, Art. no. 109753, doi: 10.1016/j.oceaneng.2021.109753.

[4] L. Jaulin and F. Le Bars, "A simple controller for line following of sailboats," in *Proc. 5th Int. Robotic Sailing Conf.* Cardiff, U.K.: Springer, 2012, pp. 107–119.

[5] T. Akiyama, J.-F. Bousquet, K. Roncin, G. Muirhead, and A. Whidden, "An engineering design approach for the development of an autonomous sailboat to cross the Atlantic Ocean," *Appl. Sci.*, vol. 11, no. 17, p. 8046, Aug. 2021, doi: 10.3390/app11178046.

[6] Q. Sun, W. Qi, H. Liu, X. Ji, and H. Qian, "Toward long-term sailing robots: State of the art from energy perspectives," *Frontiers Robot. AI*, vol. 8, p. 416, Jan. 2022, doi: 10.3389/frobt.2021.787253.

[7] U. Vautier, J. Wan, M. Dai, C. Viel, and R. Hone, "Development and test of an open source autonomous sailing robot with accessibility, generality and extendability," in *Proc. 11th Int. Robotic Sailing Conf. (IRSC)*, Southampton, U.K., 2018, pp. 11–19.

[8] D. Fernandes, L. Gomes, and A. Costa, "Wind speed and direction measurement based on time of flight ultrasonic anemometer," in *Proc. IEEE 26th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2017, pp. 1417–1422, doi: 10.1109/ISIE.2017.8001453.

[9] A. P. F. Negreiros, W. S. Correa, A. P. D. de Araujo, D. H. Santos, J. M. Vilas-Boas, D. H. N. Dias, E. W. G. Clua, and L. M. G. Gonçalves, "Sustainable solutions for sea monitoring with robotic sailboats: N-boat and F-boat twins," *Frontiers Robot. AI*, vol. 9, p. 93, Apr. 2022, doi: 10.3389/frobt.2022.788212.

[10] E. F. C. Mosquera, "Antonomous sailboat prototype sensors and electronics implementation with machine learning for navigation," M.S. thesis, BarcelonaTech, Barcelona, Spain, 2021.

[11] *Calypso Instruments*. Accessed: Oct. 17, 2022. [Online]. Available: https://calypsoinstruments.com/

[12] X. Lin, H. Liu, Q. Sun, X. Li, H. Qian, Z. Sun, and T. L. Lam, "Applying project-based learning in artificial intelligence and marine discipline: An evaluation study on a robotic sailboat platform," *IET Cyber-Systems Robot.*, vol. 4, no. 2, pp. 86–96, Jun. 2022, doi: 10.1049/csy2.12050.

[13] J. Melin, "Modeling, control and state-estimation for an autonomous sailboat," M.S. thesis, Uppsala Univ., Uppsala, Sweden, 2015.

[14] H. Saoud, M.-D. Hua, F. Plumet, and F. Ben Amar, "Routing and course control of an autonomous sailboat," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Sep. 2015, pp. 1–6, doi: 10.1109/ECMR.2015.7324218.

[15] L. Jaulin, *Mobile Robotics*. Hoboken, NJ, USA: Wiley, 2015.

[16] K. Liu, Z. Sun, Y. Qu, J. Yu, Y. Huang, and W. Zhao, "Research on key methods of autonomous sailboat's position keeping control," in *Proc. OCEANS*, Feb. 2022, pp. 1–6, doi: 10.1109/OCEANSChennai45887.2022.9775464.

[17] C. Viel, U. Vautier, J. Wan, and L. Jaulin, "Position keeping control of an autonomous sailboat," *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 14–19, 2018, doi: 10.1016/j.ifacol.2018.09.462.

AUGUSTIN MORGE is currently pursuing the degree with ENSTA Bretagne, France. He did his internship as an assistant engineering in robotics and autonomous systems at Aston University, Birmingham, U.K., from May 2022 to August 2022.

VIRGILE PELLE is currently pursuing the degree with ENSTA Bretagne, France. He did his internship as an assistant engineering in robotics and autonomous systems at Aston University, Birmingham, U.K., from May 2022 to August 2022.

JIAN WAN received the Ph.D. degree in control engineering from the University of Girona, Catalonia, Spain, in 2007. He has been a Lecturer in mechatronics and robotics at Aston University, since February 2022. He was a Lecturer in control systems engineering at the University of Plymouth, U.K., from 2015 to 2021. His research interests include statistical processing monitoring and control, set-membership methods for control, constrained optimization and control with applications to manufacturing processes, renewable energy, robotics, and autonomous systems.

LUC JAULIN received the Ph.D. degree in automatic control from the University of Orsay, France, in 1993. He has been a Professor in robotics at ENSTA Bretagne, Brest, France, since 2004. He does his research on underwater robotics using interval methods and constraint propagation. He was awarded the Moore Prize, in 2012, for applications of interval methods to robotics.