Agent-based Platform for Autonomous Sailing Research and Competition

Bruno Alves[1], Bruno Veloso[2] and Benedita Malheiro[1,2]

Abstract This paper presents a platform for real and simulated autonomous sailing competitions, which can also be used as a research tool to test and assess navigation algorithms. The platform provides back-end services – competition server, boat modelling and data storage – and supports external browsers and software agents as front-end clients. The back-end adopts the Multi-Agent System (MAS) paradigm for the internal modelling of sailing boats and offers a Web Service Application Programming Interface (API) for the external software agents and a Web application for Web browsers. As a whole, the platform offers tracking (real competitions) and simulation (simulated competitions) modes. The testing and assessment of navigation algorithms and boat models correspond to private simulated competitions. In simulation mode, the back-end internal boat agent implements a simplified physical model, including the weight, sail area, angle of the sail and rudder, velocity and direction of the wind and position and velocity of the hull, whereas the front-end external boat agent implements the navigation algorithm on the team side, ensuring the privacy of strategic knowledge. The Web application allows the configuration and launching of competitions, the registration of teams and researchers, the uploading of boat physical features for simulation as well as the live or playback viewing of real and simulated competitions. The simulation mode is illustrated with the help of a case study. The proposed platform, which is open, scalable, modular and distributed, was designed for the research community to prepare, run and gather data from real and simulated autonomous sailing competitions.

(1) ISEP-IPP, School of Engineering, Polytechnics of Porto, Rua Dr. António Bernardino de Almeida, 431, Porto, Portugal
(2) INESC TEC, Rua Dr. Roberto Frias, Porto, Portugal

1 Introduction

In recent years the research community has shown a significant interest in autonomous sailing competitions. The most important competitions involving autonomous sailing boats are the Microtransat Challenge [10], the World Robotic Sailing Championship (WRSC) [15] and the Sailbot competition [11].

The Microtransat Challenge [10] was created in 2005 as a transatlantic race for fully autonomous sailing boats comprising two classes: sailing boats propelled exclusively by wind power with hull load waterline length (LWL) restricted to 4 m; and non-sailing boats propelled by any other source of energy and with a LWL up to 2 m. One year later, the Sailbot International Robotic Sailing Regatta began at the Queen's University, Kingston, Ontario, Canada. This annual event takes place in North America and involves teams of students from different universities and colleges, including a 1 m class, a 2 m class and an open class for boats with a maximum length of 4 m [11]. In 2008, the World Robotic Sailing Championship (WRSC) and the International Robotic Sailing Conference (IRSC) started as a spin-off of the Microtransat Challenge. The WRSC contemplates the Micro-sailboat class, for boats with length overall (LOA) less than 1.5 m and a weight up to 100 kg, and the Sailboat class, for boats up to 4 m LOA and less than 300 kg [15].

Increasingly, researchers participate in these competitions in order to refine their navigation platforms and algorithms. However, since the logistics and costs of conducting real experiments with sailing boats are heavy and expensive, there is a lack of tools for the initial modelling, experimentation and debugging of sailing boat models and navigation algorithms. Moreover, such a platform can also be used to collect data for data mining as well as to follow in real time or playback competitions.

The goal of this project is to design, develop and test a versatile platform for simulated and real autonomous sailing competitions. In terms of general requirements, the platform should be open, scalable, modular and supported by open source technology. Additionally, it should implement the simulation and real operation modes. Users should be able to: (*i*) participate and follow live simulated competitions in simulation mode; (*ii*) follow live real sailing boat competitions in real tracking mode; and (*iii*) simulate their platforms and test their navigation algorithms in private simulation mode. Users include competition administrators and researchers (individuals or teams).

Our contribution is a platform providing a Representational State Transfer (REST) Web Service API for researchers to test navigation strategies and boat models, using private simulations, as well as to create, configure and follow in 2D real and simulated competitions, using a Web browser.

This paper is organized in five sections. Section 2 presents the related work. Section 3 details the project development. Section 4 presents a simulation mode case study. Finally, Section 5 concludes the paper, discussing the outcomes and future developments.

2 Related Work

This section provides an overview of open source sailing boat simulators, agent based competition platforms and autonomous sailing platforms.

Sailing Boat Simulators Sail7 is an open source tool designed for the analysis of sail boat performance. It was developed for XFLR5, an analysis tool for airfoils, wings and planes operating at low Reynolds Numbers, and is based on the Vortex Lattice Method and the 3D Panel Method [5]. Sail7 performs the polar analysis of the specified sail and hull combination.

Tracksail is an open source game developed in Java where players compete for the fastest time, guiding small sailing boats via the keyboard [8]. The implemented physical model contemplates the wind velocity and direction, the boat position, velocity and rudder angle.

Tracksail-AI, which is a modified version of the Tracksail sailing game, is intended to be played by Artificial Intelligence (AI) applications instead of humans [12]. It offers an open source API based on the original Java code to implement a simple sailing boat model, contemplating the wind velocity and direction, the boat position and velocity and the rudder and sail angles.

Multi-Agent Systems for Simulated Competitions RoboCup [6] is an annual international scientific competition in the areas of robotics and artificial intelligence. The RoboCup simulator consists of three large modules [7]: client – responsible for the dynamics of its team players and communicates with the server module via a socket, sending messages (messages) controlling the actions of the players through the protocol of the transport layer User Datagram Protocol (UDP); server and visualization.

The Simulated Car Racing Championship (SCRC) [9] is a competition platform based on the The Open Racing Car Simulator (TORCS). TORCS [9, 16, 3] provides a way to create, test and use simulated cars (BOT). Each running BOT consists of a server and a corresponding client, communicating via a dedicated application layer protocol over UDP. The BOT server sends the status of the sensors (perceptions obtained from the simulated environment) to the client every 20 ms and the client has 12 ms to decide and communicate the BOT server which actions to take. The client implements locally and independently the control algorithm.

The Trading Agent Competition (TAC) [14] involves eight trading agents in an auction-based marketplace. The main purpose of the agents (travel agents) is to create travel packages – a round-trip flight, a five-day stay in a hotel and entertainment – according to the preferences of their clients. Each agent communicates with the TAC-Server through messages on Transmission Control Protocol (TCP). The server provides a specific API for clients to access the market information and submit proposals.

Autonomous Sailing Platforms Several researchers have designed computational platforms for autonomous sailing addressing the problems of track-

ing and collision avoidance [2], testing of sailing missions [1] or AI boat control algorithms [4] as well as real and simulated boat control and modelling [13].

In 2011, Ammann et al. proposed a world server approach for the WRSC competition in order to gather, broadcast and store data from all vessels and, thus, provide tracking and collision avoidance functionalities [2].

Davison, in 2013, developed YachtSim, a yacht simulator written in C#, to aid in developing AI algorithms for an autonomous yacht [4]. The code was inspired by TrackSail-AI and includes three main applications: YachtSimServer, YachtSimClient and EyeInTheSky. The YachtSimServer provides the interface between the virtual yacht hardware and the AI running on the YachtSimClient and includes a list of regatta environment simulators. The EyeInTheSky is a client module responsible for the visual display of where and how the virtual yacht is sailing.

In 2015, Alves et al. proposed a simulation platform for testing sailing missions called MetaSail [1]. It takes into account the boat's heeling effect on the sail forces and the leeway angle, along with the righting moment produced by the keel and the hull's hydrostatic forces.

Taylor et al. designed in 2016 the experimental robotic sailing boat daemon (boatd) [13] to support real and simulated boat control, modelling, telemetry and logging. Boats are externally controlled via an HTTP API and the commands are forwarded through drivers to the Tracksail-AI simulator or to the controller of an autonomous sailing boat.

Our Agent based Platform for Autonomous Sailing Research and Competition (APASail) provides a dual mode – real and simulated – competition platform for autonomous sailing boats, offering a REST Web Service API and a Web application. The 2D simulation mode is based on Tracksail [8] and, additionally, implements the influence of the mass and the sail angle and area. Furthermore, it supports the test and simulation of navigation algorithms and boat models as private simulations.

3 APASail

Use Cases The contemplated use cases were: (*i*) registration and authentication; (*ii*) creation of real and simulated public competitions; (*iii*) creation of private simulations to test navigation algorithms; (*iv*) competition tracking; and (*v*) platform administration.

System Architecture Figure 1 displays the overall system architecture comprised of front-end and back-end. The back-end includes, in terms of components, two database servers (PostgresSQL and RethinkDB), a Web server (Apache), a JavaScript Object Notation (JSON) server (Node.js), an agent execution platform (JADE) and a RESTful API framework (RESTlet). The Apache HyperText Transfer Protocol (HTTP) server deploys the Web appli-

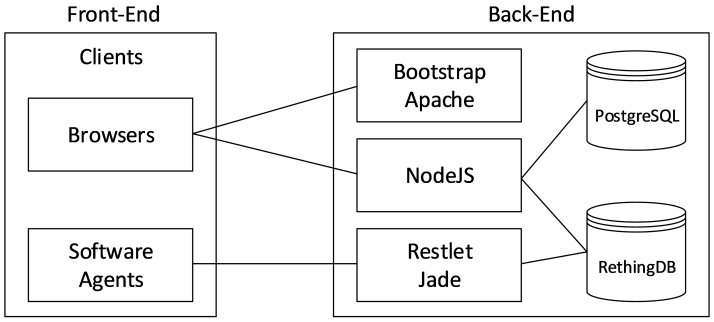


Fig. 1 System Architecture

cation, which was developed using Boostrap. The Node.js, while a server-side JavaScript module, automatically refreshes the client Web pages and acts as a bridge between the client browser and both databases. Finally, the Java Agent DEvelopment (JADE) and the Restlet frameworks are the core of the platform, *i.e.*, provide support to simulated and real competitions. Each competition is modelled as a multi-agent system, corresponding to a JADE container, where the participating boats (simulated or real) are modelled as JADE agents. The interaction between the JADE platform and the competitor boat agents is achieved through the implemented REST interface. The JADE platform is responsible for communicating all live competition data to the RethinkDB database.

The front-end contemplates two types of clients: browsers and software agents. The browser obtains the Web pages from the Apache HTTP server and interacts with the Node.js using WebSockets to update and control Web page contents. Competitors use browsers for user registration and authentication, team and boat profiling, tracking real time competitions (real and simulated) as well as to define and follow tests. In addition, administrators can manage users, teams and competitions. The client agents execute on the competitor platform. Each client agent represents an autonomous sailing platform and interacts with its JADE representative using the implemented JADE Restlet API. While in simulation mode, the simulated boat agent runs the navigation algorithm, in real mode the boat agent executes the navigation algorithm on board of the platform. In both modes, the boat agent communicates with its JADE representative to report its status and location and receive the competition status.

Functionalities In terms of permanent data modelling and storage, there are two databases. PostgreSQL database stores all the information related with users, teams, boats and competitions. The RethinkDB database stores real time competition data, *i.e.*, all competition events, including weather conditions and the position, velocity and time (PVT) of the boats.

The Web application allows users to register, authenticate and manage the profile, create private competitions and participate in real and simulated competitions. Depending on the user role, the user will be able to create (administrator), configure boat and team profiles (administrator and competitor), create, configure, follow live or playback real and simulated competitions (administrator and competitor), using an interactive 2D graphical interface.

4 Simulation Case Study

In order to test the adopted physical modelling approach, a simulation mode case study, involving two sailboats ($boat_A$ and $boat_B$), was performed. Figure 2 displays the sail course layout defined by four marks (buoy 1 to buoy 4) with a length of 400 m/leg. The tests contemplated: (*i*) Beating between the start-line and buoy 1; (*ii*) Beam Reach between buoys 1–2 and 3–4; (*ii*) Running and Reaching between buoys 2–3; and (*v*) the influence of the mass and sail area. The true wind velocity and direction were set to 9 kn±1 kn and

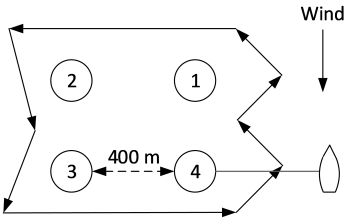


Fig. 2 Course layout used in the simulation tests

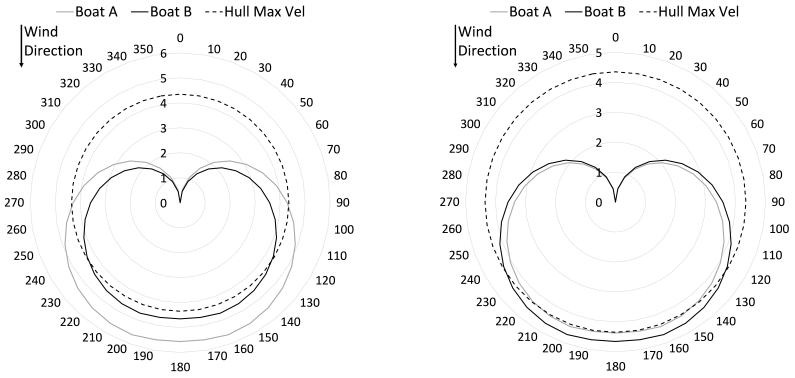
90°±10°, respectively. The sailing boats have a mass of 200 kg, a length of 3.7 m, a LWL of 3.2 m and a sail area of 3.4 m^2 in the case of $boat_A$ and of 3.0 m^2 in the case of $boat_B$.

Beating In this test, which corresponds to the first leg of the course, $boat_A$ was set to sail at an angle of 45° to the wind, while $boat_B$ adopted an angle of 40° to the wind. $boat_A$ took 692 s at an average velocity of 1.51 kn, whereas $boat_B$ spent 699 s at an average speed of 1.41 kn. As expected, $boat_A$ is faster than the sail boat $boat_B$, but takes longer since it sails a longer path.

Beam Reach This test, which corresponds to the second and forth legs of the course, was performed with both boats navigating perpendicularly to the wind. The average time and velocity in these legs was 455 s and 1.66 kn for $boat_A$ and 489 s and 1.56 kn for $boat_B$. Not only the boats sailed faster than in beating, but $boat_A$, with larger sail area, was faster.

Running In this test, which corresponds to the third leg of the course (buoy 2–buoy 3), both boats are running downwind. The average time and velocity in these legs was 348 s and 2.16 kn for $boat_A$ and 367 s and 2.03 kn for $boat_B$. As expected, this was the fastest leg and $boat_A$, with larger sail area, was the fastest.

Boat Mass and Sail Area The influence of the mass and the sail area on the velocity of a sail boat was tested using the boat polar diagrams generated from the implemented boat physical model. In this case, the wind velocity



(a) Different weights. (b) Different sail areas.

Fig. 3 Polar Diagrams

remained at 10 kn, while the boat direction varied between 0° and 360°. Figure 3a represents the velocity polar diagrams of two sail boats with equal LWL (3.2 m) and sail area (3.4 m^2), but different mass ($boat_A$ with 140 kg and $boat_B$ with 200 kg). As expected, the lightest sail boat is faster than the heavier sail boat. Figure 3b shows the velocity polar diagram of two sail boats with equal equal LWL (3.2 m) and mass (200 kg), but different sail area ($boat_A$ with 3.0 m^2 and $boat_B$ with 3.4 m^2). As expected, the boat with larger sail area is faster. The polar diagrams display also the maximum hull velocity based on the LWL of the boats.

5 Conclusion and Future Work

APASail supports real and simulated autonomous sailing competitions and interfaces seamlessly with external sailing boat controllers, client navigation agents and browsers. Internally, this research and competition platform uses dedicated agents to model sailing boats, *i.e.*, to process real time events in

real and simulation modes and implement the physical modelling in simulation mode. The latter takes into account the position, velocity, mass, rudder angle, sail area and sail angle of the sailing boat as well as the direction and velocity of the wind. When compared with Tracksail, our model additionally contemplates the influence of the mass and sail area and angle. Furthermore, researchers can create, participate and follow (live or playback) real and simulated public competitions in 2D as well as create private simulations to test different boat models (implemented in JADE agents) and navigation strategies (implemented in client agents).

In the future, we intend to improve the physical modelling in order to contemplate the boat LWL, wave height, tide and current as well as test the platform thoroughly both with real and simulated competitions. Finally, we would like to explore with 3D boat rendering.

Acknowledgements

This work was partially financed by the European Regional Development Fund (ERDF) through the Operational Programme for Competitiveness and Internationalisation (COMPETE Programme), within project «FCOMP-01-0202-FEDER-023151» and project «POCI-01-0145-FEDER-006961», and by national funds through the Fundação para a Ciência e Tecnologia (FCT) – Portuguese Foundation for Science and Technology – as part of project UID/EEA/50014/2013.

References

1. Alves, J.C., Cruz, N.A.: METASail – A Tool for Planning, Supervision and Analysis of Robotic Sailboat Missions, pp. 57–64. Springer International Publishing, Cham (2015)
2. Ammann, N., Hartmann, F., Jauer, P., Krüger, J., Meyer, T., Bruder, R., Schlaefer, A.: Global Data Storage for Collision Avoidance in Robotic Sailboat Racing – the World Server Approach, pp. 157–166. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
3. Corneliussen, S.C., Westergaard, M.: Combining offline and online learning in developing an adaptive controller for a simulated car racing environment (2011)
4. Davison, B.: YachtSym. Online; `https://github.com/rbdavison/YachtSim` (2016). Accessed in June 2016
5. Deperrois, A.: Sail7. Online; `http://www.xflr5.com/sail7/sail7.html` (2012). Accessed in June 2016
6. Federation, T.R.: About robocup: What is robocup? (2015). URL `http://www.robocup.org/about-robocup/`
7. Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., Matsubara, H.: Robocup: A challenge problem for AI. AI Magazine **18**(1), 73–85 (1997)
8. Kuusela, T., Brockmann, S., Johnson, R.: Tracksail. Online; `http://tracksail.sourceforge.net/` (2005). Accessed in June 2016

9. Loiacono, D., Cardamone, L., Lanzi, P.L.: Simulated car racing championship: Competition software manual. Online: http://arxiv.org/pdf/1304.1672.pdf (2013). Accessed in June 2016
10. microtransat.org: The microtransat challenge. Online: http://www.microtransat.org/ (2016). Accessed in June 2016
11. sailbot.org: Sailbot. Online; http://sailbot.org/ (2016). Accessed in June 2016
12. Taylor, L., Nicholls, S., Stone, L., Cartwright, T.: TrackSail-AI. Online; http://github.com/boatd/tracksail-ai (2013). Accessed in June 2016
13. Taylor, L., Nicholls, S., Stone, L., Cartwright, T.: Experimental robotic sailing boat daemon. Online; http://boatd.readthedocs.io/ (2016). Accessed in April 2017
14. Wellman, P., Wurman, P., O'Malley, K., Bangera, R., Lin, S.D., Reeves, D., Walsh, W.: Designing the market game for a trading agent competition. Internet Computing, IEEE **5**(2), 43–51 (2001)
15. WRSC 2016 Organising Committee: World robotic sailing championship. Online; http://www.wrsc2016.com/competition.html (2016). Accessed in June 2016
16. Wymann, B., Dimitrakakis, C., Sumner, A., Espié, E., Guionneau, C., Coulom, R.: TORCS, the open racing car simulator. Online; http://www.torcs.org (2013). Accessed in June 2016