Reinforcement-learning robotic sailboats: simulator and preliminary results

Eduardo C. Vasconcellos
Fluminense Federal University
Niterói, RJ, Brazil
evasconcellos@id.uff.br

Ronald M. Sampaio
Fluminense Federal University
Niterói, RJ, Brazil
ronaldmaymone@id.uff.br

André P.D. Araújo
Fluminense Federal University
Niterói, RJ, Brazil
andrepda@id.uff.br

Esteban W. G. Clua
Fluminense Federal University
Niterói, RJ, Brazil
esteban@ic.uff.br

Philippe Preux
Center Inria de l'Université de Lille
Villeneuve d'Ascq, Lille, France
philippe.preux@univ-lille.fr

Raphael Guerra
Fluminense Federal University
Niterói, RJ, Brazil
rguerra@ic.uff.br

Luiz M. G. Gonçalves
Federal University of Rio Grande do Norte
Natal, RN, Brazil
lmarcos@dca.ufrn.br

Luis Martí
Center Inria Chile
Las Condes, Chile
lmarti@inria.cl

Hernan Lira
Center Inria Chile
Las Condes, Chile
hernan.lira@inria.cl

Nayat S. Pi
Center Inria Chile
Las Condes, Chile
nayat.sanchez-pi@inria.cl

Abstract

This work focuses on the main challenges and problems in developing a virtual oceanic environment reproducing real experiments using Unmanned Surface Vehicles (USV) digital twins. We introduce the key features for building virtual worlds, considering using Reinforcement Learning (RL) agents for autonomous navigation and control. With this in mind, the main problems concern the definition of the simulation equations (physics and mathematics), their effective implementation, and how to include strategies for simulated control and perception (sensors) to be used with RL. We present the modeling, implementation steps, and challenges required to create a functional digital twin based on a real robotic sailing vessel. The application is immediate for developing navigation algorithms based on RL to be applied on real boats.

1 Introduction

The development of Unmanned Surface Vessels (USVs), also known as Autonomous Surface Vessels (ASVs), is currently an interesting and exciting research field, with applications in surveillance, rescue missions, structural inspection, and environmental monitoring, among others [7, 16, 9]. These ASVs can be applied to fulfill dangerous or long recursive missions that could risk human lives or have a higher cost when executed by a human crew.

Developing a fully autonomous USV is not easy, and a significant challenge is the Guidance, Navigation, and Control (GNC) system [11, 12]. System prototyping and testing could be costly with a real USV. An undesired or unexpected behavior can damage the USV or other environmental entities. Therefore, it is usual that researchers fall back on simulated environments, where one can run different experiments without being afraid of losing the USV or causing damage to others. Simulation also has the advantage that one can control various aspects of it, performing several simultaneous experiments. For example, one can submit the USV to rare (or extreme) environmental conditions, which might not always be possible in the real environment.

A USV motion can be sourced by motorized propellers (most common boats), using the wind (sailboats), or even using a mix of them, which is the one that will be treated here. While other unmanned vehicles, like cars and many different types of robots, are often simulated using the physics of rigid bodies and collisions, a USV is under the influence of other forces generated by the water and air. For a USV simulation, one needs to compute not only the dynamic and static forces of a rigid body but also the hydrodynamic, hydrostatic, and aerodynamic forces acting on the hull. In the case of a sailing robot, aerodynamic forces from the wind also act on the sail's surface.

In this direction, this paper aims to present some challenges, problems, and a possible approach for building a virtual oceanic environment application, considering that it will be populated with USV digital twins, which include sailing robots. Our proposal is based on a literature review and on experiences in building real autonomous vessels and their simulated digital twin scenarios. The virtual world has been created based on our experience developing real USVs for environmental monitoring (including autonomous motor-propelled and sailboats). Its application is immediate, with the main goal of saving time and effort to develop navigation algorithms to be applied on our physical boats. Also, we are developing reinforcement learning algorithms and other machine learning tasks to be applied to our real USVs on top of this platform. Hence, the contribution of this work resides in introducing several aspects, challenges, and requirements necessary for creating the proposed digital twin.

2 Key features on building a reliable simulator for sailing robots

A simulated world for robotic development must implement physics, sensors, and actuators so the robot can perceive and interact with its environment. To enable the implementation of sensors like cameras and LiDAR, the simulator must be capable of rendering a realistic 3D scene representing the robot's surrounding environment. Therefore, a system with reliable physics and rendering capabilities is very desirable. Some authors advocate that modern game engines will be good candidates for creating realistic simulations, joining all characteristics that a developer desires [8].

2.1 Physics

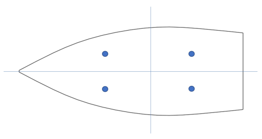
A physics engine is an important variable for building a reliable virtual world. Several physics engines can compute collision, contact, and reaction forces among rigid bodies.

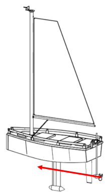
On the other hand, water and wind simulations can be achieved using parametric models to describe the hydrodynamic, hydrostatic, and aerodynamic forces acting on different vessel parts (hull, keel, rudder, sail, etc). USV simulations were proposed using Gazebo [12, 3] and Unity [5]. Although, the only sailing robot simulator is within the work of Paravisi et al. (2019) [12].

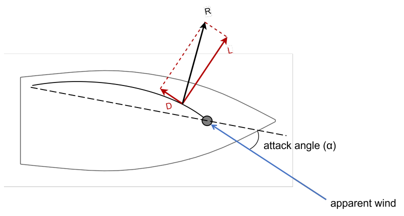
2.2 ROS support

ROS (Robot Operational System) [13] is a largely used API within the robotics development community. ROS is a free, open-source set of libraries and tools that enables a developer or a team to integrate all systems that compose a robot, like motors, sensors, batteries, and control software. Implementing sensors and actuators is very important so the robot can perceive its environment and navigate through it. ROS enables us to build a communication infrastructure for collecting sensor

(a) Real boat sailing.


(b) Hull division to apply the buoyancy force.


(c) Propeller pushing force.



(d) Aerodynamic lift and drag forces as a function of attack angle.

Figure 1: E-Boat physics characterization.

data and sending commands to the actuators. Integrating the simulated environment with ROS makes commuting between a real and a virtual system easy.

2.3 Sensors and actuators

In a USV, the sensor array is essential to autonomous navigation and control. For so, they must be simulated as accurately as possible in a virtual world [6]. Depending on the simulation platform, one can find many implementations for various sensors, such as cameras, LiDAR, and GPS [2, 14, 12]. These implementations seek to reproduce actual sensors in all aspects, including noise, data structure, and transmission.

For a digital twin, the simulated sensor data is aimed to feed the navigation and control agent with information that describes the current state of the vessel. Then, the agent makes decisions, and through the actuators, it seeks to change the vessel's state. For a sailing robot, there are three basic actuators that the agent can control: the boom (which controls the main sail), the rudder (which controls the boat direction), and the propeller (forward propulsion, if available).

2.4 Integration of USV digital twin into the environment

This challenge is related to building the sailing robot's digital twin to emulate all the real boat's characteristics as accurately as possible, starting with creating a 3D model to represent the sailing robot in the virtual world.

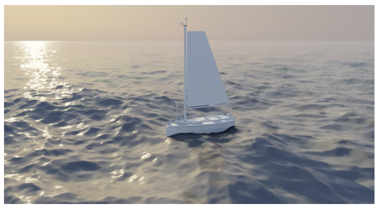
We built a sailing robot digital twin, the E-Boat, to address this challenge. The 3D model, mass distribution, inertia matrix, and collision shapes were created using Onshape (https://www.onshape.com/), a Computer-aided Design (CAD) software delivered on the Internet using a software-as-a-service model. Nevertheless, this physics is just part of the whole environment [1]. The boat motion under the influence of the water is modeled using the 6-DOF motion model proposed by Fossen [4], following an implementation strategy similar to the works of Paravisi [12] and Bingham [3]. The hydrodynamic and hydrostatic forces are computed separately, and instead of computing a single buoyance force, we split the hull into four parts, calculating the buoyance force for each part as seen in Figure 1b. This solution helps to create a more realistic motion under wave conditions.

The sailing robot's main propulsion is the wind force, and we compute this force using equations:

$$L = 0.5\,\rho_{\text{air}} V_A^2\, A\, C_L(\alpha); \quad D = 0.5\,\rho_{\text{air}} V_A^2 A\, C_D(\alpha)\,. \tag{1}$$

where ρ_{air} is the air density, V_A is the apparent wind speed, A is the area of the sail, α is the attack angle, i.e., the angle between the sail chord line and the apparent wind direction, C_L is the lift coefficient and C_D is the drag coefficient. The $C_L(\alpha)$ and $C_D(\alpha)$ were estimated from real data from experimental missions performed with the sailing robot in the real world (see Figure 1a). The propulsion force generated by the propeller is set by a discrete function described by the manufacturer and applied in the center of the propeller. Figure 1c illustrates the position where the force generated by the propeller is applied, and Figure 1d depicts the scheme for the wind forces acting on the center of effort of the sail.

We model the hydrodynamic forces acting on the rudder and the keel similarly we did for the sail. As the sailing robot moves, the water generates lift and drag forces on the keel and the rudder. The keel is static, but the rudder can change its attack angle and generate lift that will cause the robot to


(a) Omniverse rendering.

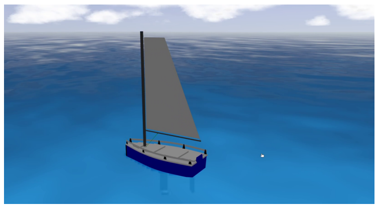

(b) Gazebo rendering.

Figure 2: Sea surface rendering in Omniverse and Gazebo.

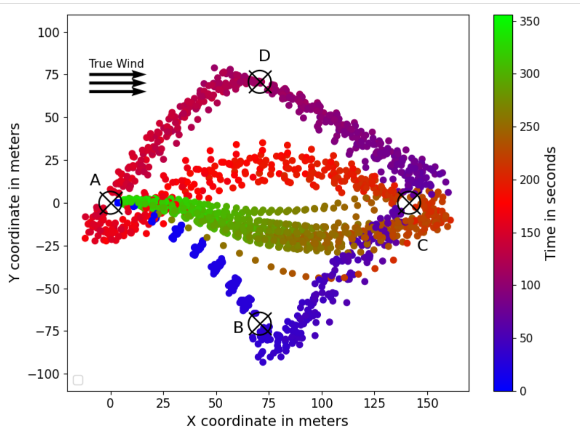


Figure 3: RL agent policy behavior in twenty runs of the same mission. The mission path is defined as BCDACA.

change direction. The equations used to calculate the forces are similar to equations in (1), where we replace the air density with the water density. The lift and drag coefficients $C_L(\alpha)$ and $C_D(\alpha)$ are also estimated by experiments with the real sailing robot.

3 Towards autonomous navigation and control with reinforcement learning

Using the E-Boat and an open ocean environment we built based on the project VRX [3], we conducted some experiments in autonomous GNC using reinforcement learning. Our main simulation platform is Gazebo [10]. It offers a reliable physics engine, and the modularity to add sensors, actuators, and new physics to the simulation using C++ plugins. We are also developing an open-world simulation for Nvidia Omniverse. Omniverse has more robust physics and graphics engines, but as a broad tool to create virtual worlds, it is more complex and demands more computational power. Figure 2 depicts the sea surface rendering in Omnivers and Gazebo 11.

Using the Gazebo simulation combined with stable-baselines3 and Gymnasium, we managed to train an agent to navigate the E-Boat through a mission composed of six waypoints. The agent was trained using PPO [15].

Figure 3 shows the progress of the E-Boat in twenty runs of the same mission. The mission path is defined as BCDACA. This initial result indicates the maneuvers executed by the agent are consistent, with small variations through the runs. The variation in the path from one run to another is due to the waves hitting the boat during the mission.

4 Conclusions

There are many challenges to developing autonomous GNC systems for sailing robots. We believe that pursuing a reliable simulated environment is a key resource to address this challenge.

We have been developing an oceanic virtual environment mainly to support research in sailing robots, which will also support other types of USVs. We are focusing our efforts on Gazebo and Omniverse, due to their robust physics engines and connections with ROS and reinforcement learning APIs, like stable-baselines3. Although Omniverse is superior in rendering capabilities and offers a connection with a broad specter of software and development tools, Gazebo offers a more direct approach to building worlds for robotic simulations as it was conceived as a tool for the robotic community.

Our initial results with an RL agent to control and navigate the sailing robot were promising. In the future, we will develop this agent and compare its results with other algorithms in the literature.

Acknowledgments

We acknowledge and are thankful to CAPES AMSUD (proj. 88881.522966/2020-01), NVIDIA, CNPq-FNDCT-MCTI (405535/2022-8) and the City of Niterói (Brazil) for supporting and funding this research.

References

[1] Anderson, B. D. (2003). *The Physics of Sailing Explained.* Sheridan House.

[2] Aranha, C., Carvalho, S., and Goncalvez, L. (2002). Cambio: realistic three dimensional simulation of humanoids based on computer vision and robotics. In *Proceedings. XV Brazilian Symposium on Computer Graphics and Image Processing*, pages 388–395.

[3] Bingham, B., Aguero, C., McCarrin, M., Klamo, J., Malia, J., Allen, K., Lum, T., Rawson, M., and Waqar, R. (2019). Toward maritime robotic simulation in gazebo. In *Proceedings of MTS/IEEE OCEANS Conference*, Seattle, WA.

[4] Fossen, T. I. (2011). *Handbook of marine craft hydrodynamics and motion control*. Wiley.

[5] Gan, W., Qu, X., Song, D., Sun, H., Guo, T., and Bao, W. (2022). Research on key technology of unmanned surface vehicle motion simulation based on unity3d. In *OCEANS 2022, Hampton Roads*, pages 1–5.

[6] Harris, A. and Conrad, J. M. (2011). Survey of popular robotics simulators, frameworks, and toolkits. In *2011 Proceedings of IEEE Southeastcon*, pages 243–249.

[7] Jorge, V. A., Granada, R., Maidana, R. G., Jurak, D. A., Heck, G., Negreiros, A. P., dos Santos, D. H., Gonçalves, L. M., and Amory, A. M. (2019). A survey on unmanned surface vehicles for disaster robotics: Main challenges and directions. *Sensors*, 19(3):702.

[8] Juliani, A., Berges, V.-P., Teng, E., Cohen, A., Harper, J., Elion, C., Goy, C., Gao, Y., Henry, H., Mattar, M., et al. (2018). Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*.

[9] Júnior, A. G., Araújo, A. P., Silva, M. V., Aroca, R. V., and Gonçalves, L. M. (2013). N-boat: An autonomous robotic sailboat. In *2013 Latin American Robotics Symposium and Competition*, pages 24–29. IEEE.

[10] Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154.

[11] Liu, Z., Zhang, Y., Yu, X., and Yuan, C. (2016). Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control*, 41:71–93.

[12] Paravisi, M., H. Santos, D., Jorge, V., Heck, G., Gonçalves, L. M., and Amory, A. (2019). Unmanned surface vehicle simulator with realistic environmental disturbances. *Sensors*, 19(5).

[13] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y., et al. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan.

[14] Schluse, M., Priggemeyer, M., Atorf, L., and Rossmann, J. (2018). Experimentable digital twins—streamlining simulation-based systems engineering for industry 4.0. *IEEE Transactions on Industrial Informatics*, 14(4):1722–1731.

[15] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

[16] Silva Junior, A. G. d., Lima Sa, S. T. d., Santos, D. H. d., Negreiros, Á. P. F. d., Souza Silva, J. M. V. B. d., Alvarez Jacobo, J. E., and Garcia Goncalves, L. M. (2016). Towards a real-time embedded system for water monitoring installed in a robotic sailboat. *Sensors*, 16(8):1226.