Monitoring Safety and Reliability of Underwater Robots: A Case Study

Mahsa Varshosaz and Andrzej Wąsowski

IT University of Copenhagen, Copenhagen, Denmark
{mahv@itu.dk, wasowski@itu.dk

Abstract. Exploring oceans and monitoring underwater infrastructure is becoming ever more important. Autonomous underwater robots are used to operate tasks in hostile and dangerous underwater environment without human intervention. To achieve their full potential, the safety and reliability of their behavior is crucial, as their malfunction or loss can lead to catastrophic consequences. In this paper, we provide a case study involving the simulation of underwater robots and the analysis of a set of safety properties of the robots using runtime monitoring. We demonstrate the challenges in checking such properties in this context.

Keywords: Autonomous underwater robots · Safety · Monitoring · Testing.

1 Introduction

Oceans cover around 70% of the surface of the earth, a source of unlimited resources and food. Yet, a large part of the oceans remains unexplored due to the hostile and inaccessible underwater environment which poses significant risks for human operations. Underwater robots are increasingly used for monitoring marine environments and undersea installations [3]. Despite advancements in underwater robotic missions, their capabilities remain constrained. This is exacerbated by the high cost of failures in such missions, which can lead to harm to fragile marine environments or damage to, and loss of, expensive robots [7, 14, 5].

While there is an increase in the use of underwater robotics, testing the properties of their missions and behavior remains understudied [28]. The common practice in academia and industry for checking the behavior of the robots in a mission is to run simulations and observe the behavior. An example of a mission for an underwater robot with a safety concern is “The robot inspects a pipeline while keeping a safe distance x from it.” Multiple simulators have been proposed for modeling the behavior of underwater robots and their environment [25, 21, 23, 12, 26, 29, 13, 17, 9]. However, so far no systematic methods exist for checking the properties of robot behavior in the simulations. There is little research on the formal techniques for testing and monitoring the properties of such systems.

In this short paper, we demonstrate the use of offline monitoring for properties of underwater missions using a simple case study. We discuss some of the challenges in checking the safety and reliability of such systems. The contributions of this paper are:

- A demo simulation of a simple underwater mission involving two robots cooperating to inspect an underwater area, using the HoloOcean simulator [30].
- An application of a stream-based monitoring tool RTLola [27] to check examples of safety properties of the designed mission.
- A discussion of current challenges and limitations in application of formal methods and testing techniques and future directions.

Related work. There are a number of established simulators for underwater robotics such as, UWsim [25], UUV simulator [21] based on Gazebo [18], a simulation platform commonly used in robotics. Other examples of simulators tailored for underwater robotic systems include Free-Floating [17], Rock-Gazebo [31], URSim [16], Unmanned Surface Vehicle simulator (USVSim) [24]. HoloOcean [30] is among the state of the art underwater simulators that has been developed recently. Despite the abundance of tools, the field of underwater robotic simulation has not yet reached the level of maturity of comparable applications such as terrestrial and aerial robotics [10]. To the best of our knowledge, there are no existing frameworks for testing and monitoring of autonomous underwater robotics safety and performance that are using a methodic approach. There are several works on risk factor identification and assessment in the operations and missions of marine robotics. Chen et al. [8] divide the risk factors in three main categories, namely, technical, human and environmental factors. A number of works consider using risk analysis methods such as Monte Carlo simulations [2, 32, 22, 33], fault tree analysis [6, 4, 15, 1], and Fuzzy system dynamics risk analysis [19, 20]. However, these studies do not cover the test generation and monitoring of the safety properties.

The rest of the paper is organized as follows: in Section 2, we give a brief description of the concepts and background on simulation of underwater robot missions in HoloOcean and monitoring cyber-physical systems using RTLola. In Section 3, we discuss the simulation of the mission, followed by the discussion of monitoring safety and reliability properties. In the last section we discuss also some of the challenges in this context and future directions.

2 Background

In this section, we introduce some of the basic concepts used in simulation and monitoring of the safety properties of underwater robotics in this paper.

2.1 Simulation using HoloOcean

HoloOcean [30] is an open source simulator for underwater robotics and their environments that is built upon unreal engine 4 [11]. HoloOcean provides a range of predefined models of underwater robots. These robots which are referred to as agents can be configured to perform different tasks. Simulating an underwater mission in HoloOcean, requires defining scenarios. The scenarios are defined as JSON files. A simple example of a scenario is given in figure 1.

```
scenario = {
    "name": "example1",
    "world": "SimpleUnderwater",
    "ticks_per_sec": 60,
    "agents": [
        {
            "agent_name": "auv0",
            "agent_type": "HoveringAUV",
            "sensors": [
                {
                    "sensor_type": "RGBCamera"
                },
                {
                    "sensor_type": "LocationSensor"
                }
            ],
            "control_scheme": 0,
            "location": [-10, 1, -5]
        }
}
```

Fig. 1: Scenario Configuration JSON file

This scenario has a name, a world (environment), and defines the number of frame ticks per unreal seconds. HoloOcean allows for adding an arbitrary number of agents to be used in a scenario. In this scenario, one of the pre-defined agents in HoloOcean, named, *HoveringAUV* is added. This is a simple Autonomous Underwater Vehicle (AUV) with 8 thrusters. The scenario attaches two sensors, the RGBCamera and Location sensors, to the agent. The control scheme *0*, one of the available schemes for HoveringAUV agent, uses an 8-length floating point vector to specify the force for each thruster of the agent. In the scenario, the initial state, including the coordinates in which the agent starts in, is defined. As presented in the figure 1, a scenario is used for configuring the scene of the simulation and the entities that are involved in the simulation. There are several other parameters that can be configured to create more complex scenarios.

2.2 Safety Monitoring

There are different approaches for monitoring properties of robotic systems. RTLola [27] is a real time monitoring toolkit for monitoring properties of cyber-physical systems. RTLola evaluates streams of input data, called input streams, such as sensor readings, and provides statistical and logical assessments based on these inputs. In this paper, we use the off-line monitoring capabilities of RTLola. We store the readings of the sensors and check the properties of system using RTLola. The tool checks specifications against input streams and outputs defined based on the input streams. An example of a specification is shown in figure 2.

```
input a : Float32
input b : Float32
output z := a+b

trigger z< 100.0
```

Fig. 2: RTLola specification example

In this specification, there are two input values of type *Float32*. RTLola supports a range of types for inputs including Bool, String, Int, or Float. The specification also includes an output stream named z. Output streams are computed in terms of the values of the other streams. Additionally, a trigger specification defines thresholds and other logical conditions on the values of output streams, and raise an alarm or execute some other predefined action if the condition becomes true [27]. In this case, the trigger raises an alarm if the value of output value z is smaller than 100.0.

3 Monitoring Safety for Underwater Missions

In this section, we provide an example of monitoring safety properties of an underwater mission. The scenario in this mission includes two underwater robots performing inspection of an area in cooperation. The goal for the robots is that they start in an initial position and move in coordination to a target point. The robots should move without colliding with each other and should not move too far from each other. This is a simplified example of a mission where one robot needs to collect images from the underwater area while the other provides the necessary light to improve the quality of the images. Such mission can be complicated due to the limitations in communication and noise in underwater environment.

We have configured the scenario with two *Hovering AUVs* selected from HoloOcean set of agents. Figure 3 shows the two hovering AUVs in different weather conditions in the environment. Both AUVs have location sensors which publish the location of the AUV in the world. The AUVs are supposed to cover 5 meters vertical distance starting from an initial position. We consider monitoring two safety specifications for these AUVs. The specifications are presented in Figure 4. In the first specification, it is checked that if the AUVs come closer than a safe distance (1.0 m), then they should go back to the safe distance within two time steps. In this specification the input stream is the distance of the AUVs.

Fig. 3: An example of two hovering AUVs performing missions in different weather conditions.

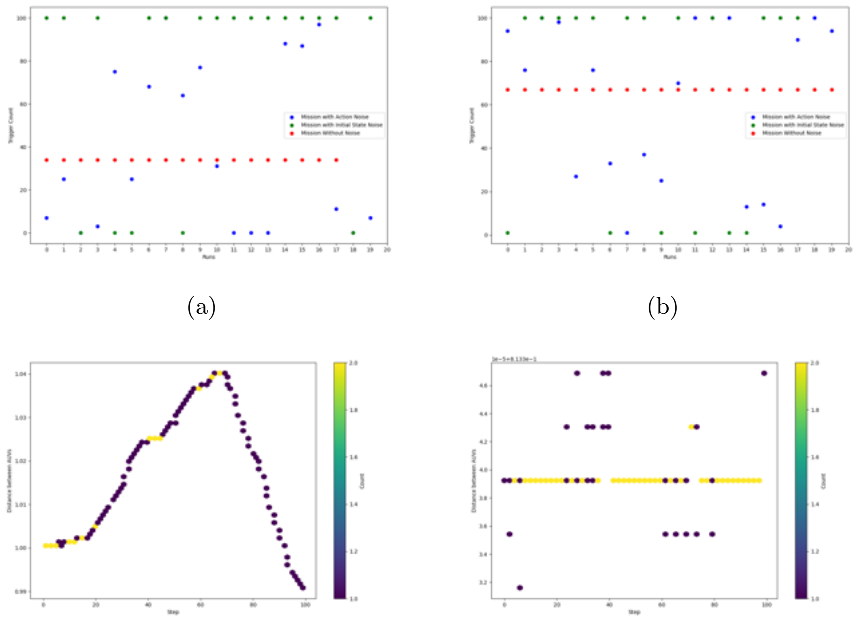
```
//specification 1
input distance: Float64

output below_one := distance < 1.0
output back_to_one := distance >= 1.0

output below_one_steps: UInt32 := if below_one then 1 else 0
output steps_below_one: UInt32 := if below_one then (
    steps_below_one[-1].defaults(to: 0) + 1) else 0

trigger  (!(steps_below_one >= 2) || back_to_one)
```

```
//specification 2
input distance: Float64
trigger distatnce< 1.0 || distance> 1.2
```

Fig. 4: RTLola specifications for case study.

The output *steps_below_one* keeps track of the number of steps where the safe distance is violated. If this value if greater than 2 then a trigger is recorded. This specification is concerned with the recoverability of the system when a violation of the property happens. The second specification indicates an invariant that is the AVUs should keep a safe distance (greater than 1.0 m) and at the same time they should not be too far apart as it can affect the quality of the mission.

Given the defined scenario, we run the simulations in HoloOcean. The behavior of the AUVs is monitored against these specifications using RTLola. The data published by the location sensors is used as input for RTLola. In the simulations, we incorporate two types of environmental noise and monitor the system's behavior both with and without the presence of noise. We apply uniformly distributed random noise $U(0, 1)$ to the initial state of AUVs. This noise can represent sensor inaccuracies and environmental disturbances. Furthermore, we introduce a discrete random noise ranging from 0 to 5 to the forces applied to each thruster in the AUVs. The distance is monitored in presence and absence of noise. The noise on the initial location is added as a part of the configurations in scenario definition. The distance of AUVs in the missions that is derived from



(a) (b)

(c) (d)

Fig. 5: (a): the trigger counts for specification 1, (b): the trigger counts for specification 2, in presence and absence of environmental noise. (c): the distance between AUVs in 100 steps in presence of noise in thrusters force, (d): the distance between AUVs in presence of noise in initial states.

sensor data and the results of monitoring the safety specifications are presented in Figure 5.

Discussion. The results in figure 5a and figure 5b show that when no noise is introduced, there is no significant variation in the measured distance and hence the result of monitoring the specifications remain the same between different runs. The most variation in the number of triggers (violation of properties) is when the noise on thrusters' force is applied. This causes visible changes in the number of triggers. Also the results show that the variance in the distance is higher when applying noise to the amount of force applied to the eight thrusters of each of the two AUVs.

The results show that the behavior of AUVs with respect to the safety specifications vary in presence of noise and under different scenario configurations. As it is common for sensors and the system to experience noise in underwater environments, checking their behavior under different noisy setups is paramount to ensuring their safety. There are a range of configurations that are possible considering the scenarios that can be defined in HoloOcean. This is not specific to HoloOcean simulator, but a property of the domain as there are several environmental factors that can affect the behaviour of the robots. This experiment shows that just manually creating scenarios may not be enough for checking the properties of such systems. It is important to develop techniques for automatic generation of such scenarios and then monitoring the behavior under generated scenarios. As an example, the frequency that the sensors publish their data is configurable in the scenarios. The results of safety monitoring can be highly depending on the values of frequencies as depending on the nature of the events, some frequencies might lead to missing violations of safety specifications. Furthermore, simulation scenario generation techniques are needed that can generate interesting scenarios which are more likely to violate the specifications.

As a direction for the future work, we aim at developing a framework for generating interesting and diverse scenarios for underwater robotic systems. Furthermore, we aim at covering richer specifications of the underwater missions, and more diverse safety and reliability related properties of such systems.

4 Conclusion

In this paper we presented a case study on monitoring safety properties of a simulated underwater mission. We demonstrate that due to the complexity of the systems and the underwater environment, the generation of interesting and failure revealing scenarios is challenging. There are several aspects such as limited communication and noise that need to be taken into account in simulation and safety monitoring of underwater robotic missions. We show the results and effects of monitoring safety properties in presence of different types of noise in the environment for the case study.

Acknowledgments. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 956200 REMARO, Reliable AI for Marine Robotics.

References

1. Aslansefat, K.: A strategy for reliability evaluation and fault diagnosis of autonomous underwater gliding robot based on its fault tree (matlab simulink) (06 2015). https://doi.org/10.13140/RG.2.1.4616.4240
2. Bian, X., Mou, C., Yan, Z., Xu, J.: Simulation model and fault tree analysis for auv. In: 2009 International Conference on Mechatronics and Automation. pp. 4452–4457 (2009). https://doi.org/10.1109/ICMA.2009.5246716
3. Blidberg, D.R., Turner, R.M., Chappell, S.G.: Autonomous underwater vehicles: Current activities and research opportunities. Robotics Auton. Syst. **7**, 139–150 (1991), https://api.semanticscholar.org/CorpusID:37681764
4. Brito, M.P.: Uncertainty management during hybrid autonomous underwater vehicle missions. In: 2016 IEEE/OES Autonomous Underwater Vehicles (AUV). pp. 278–285 (2016). https://doi.org/10.1109/AUV.2016.7778684

5. Brito, M.P., Griffiths, G., Challenor, P.: Risk analysis for autonomous underwater vehicle operations in extreme environments. Risk Analysis **30**(12), 1771–1788 (2010). https://doi.org/https://doi.org/10.1111/j.1539-6924.2010.01476.x, https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1539-6924.2010.01476.x
6. Chang, Y., Brito, M.: On the reliability of experts' assessments for autonomous underwater vehicle risk of loss prediction: Are optimists better than pessimists? (09 2018)
7. Chen, X., Bose, N., Brito, M.P., Khan, F., Thanyamanta, B., Zou, T.: A review of risk analysis research for the operations of autonomous underwater vehicles. Reliab. Eng. Syst. Saf. **216**, 108011 (2021), https://api.semanticscholar.org/CorpusID:239265830
8. Chen, X., Bose, N., Brito, M.P., Khan, F., Thanyamanta, B., Zou, T.: A review of risk analysis research for the operations of autonomous underwater vehicles. Reliab. Eng. Syst. Saf. **216**, 108011 (2021), https://api.semanticscholar.org/CorpusID:239265830
9. Cieślak, P.: Stonefish: An advanced open-source simulation tool designed for marine robotics, with a ros interface. In: OCEANS 2019 - Marseille. pp. 1–6 (2019). https://doi.org/10.1109/OCEANSE.2019.8867434
10. Ciuccoli, N., Screpanti, L., Scaradozzi, D.: Underwater simulators analysis for digital twinning. IEEE Access **12**, 34306–34324 (2024). https://doi.org/10.1109/ACCESS.2024.3370443
11. Epic Games: Unreal Engine. https://www.unrealengine.com (2019)
12. Ganoni, O., Mukundan, R., Green, R.: A generalized simulation framework for tethered remotely operated vehicles in realistic underwater environments. Drones **3**(1) (2019). https://doi.org/10.3390/drones3010001, https://www.mdpi.com/2504-446X/3/1/1
13. Garg, S., Quintas, J., Cruz, J., Pascoal, A.M.: Netmarsys - a tool for the simulation and visualization of distributed autonomous marine robotic systems. In: 2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV). pp. 1–5 (2020). https://doi.org/10.1109/AUV50043.2020.9267922
14. Griffiths, G., Collins, K. (eds.): Masterclass in AUV technology for Polar science: collaborative Autosub science in extreme environments. Proceedings of the International Masterclass, 28-30 March 2006, National Oceanography Centre, Southampton, UK. Society for Underwater Technology (2007), https://eprints.soton.ac.uk/46019/, cD-ROM in back pocket
15. Harris, C.A., Phillips, A.B., Dopico-Gonzalez, C., Brito, M.P.: Risk and reliability modelling for multi-vehicle marine domains. In: 2016 IEEE/OES Autonomous Underwater Vehicles (AUV). pp. 286–293 (2016). https://doi.org/10.1109/AUV.2016.7778685
16. Katara, P., Khanna, M., Nagar, H., Panaiyappan, A.: Open source simulator for unmanned underwater vehicles using ros and unity3d. In: 2019 IEEE Underwater Technology (UT). pp. 1–7 (2019). https://doi.org/10.1109/UT.2019.8734309
17. Kermorgant, O.: A dynamic simulator for underwater vehicle-manipulators. In: Brugali, D., Broenink, J.F., Kroeger, T., MacDonald, B.A. (eds.) Simulation, Modeling, and Programming for Autonomous Robots. pp. 25–36. Springer International Publishing, Cham (2014)
18. Koenig, N., Howard, A.: 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). vol. 3, pp. 2149–2154 vol.3 (2004). https://doi.org/10.1109/IROS.2004.1389727
19. Loh, T.Y., Brito, M.P., Bose, N., Xu, J., Tenekedjiev, K.: Fuzzy system dynamics risk analysis (fusdra) of autonomous underwater vehicle operations in the antarctic. Risk

Analysis **40**(4), 818–841 (2020). https://doi.org/https://doi.org/10.1111/risa.13429, https://onlinelibrary.wiley.com/doi/abs/10.1111/risa.13429

20. Loh, T.Y., Brito, M.P., Bose, N., Xu, J., Tenekedjiev, K.: Policy recommendations for autonomous underwater vehicle operations through hybrid fuzzy system (2019), https://api.semanticscholar.org/CorpusID:213270763
21. Manhães, M.M.M., Scherer, S.A., Voss, M., Douat, L.R., Rauschenbach, T.: Uuv simulator: A gazebo-based package for underwater intervention and multi-robot simulation. In: OCEANS 2016 MTS/IEEE Monterey. pp. 1–8 (2016). https://doi.org/10.1109/OCEANS.2016.7761080
22. Merckelbach, L.: On the probability of underwater glider loss due to collision with a ship. Journal of Marine Science and Technology **18** (03 2012). https://doi.org/10.1007/s00773-012-0189-7
23. Morency, C., Stilwell, D.J., Hess, S.: Development of a simulation environment for evaluation of a forward looking sonar system for small auvs. In: OCEANS 2019 MTS/IEEE SEATTLE. pp. 1–9 (2019). https://doi.org/10.23919/OCEANS40490.2019.8962650
24. Paravisi, M., H. Santos, D., Jorge, V., Heck, G., Gonçalves, L.M., Amory, A.: Unmanned surface vehicle simulator with realistic environmental disturbances. Sensors **19**(5) (2019). https://doi.org/10.3390/s19051068, https://www.mdpi.com/1424-8220/19/5/1068
25. Prats, M., Pérez, J., Fernández, J.J., Sanz, P.J.: An open source tool for simulation and supervision of underwater intervention missions. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 2577–2582 (2012). https://doi.org/10.1109/IROS.2012.6385788
26. Razzanelli, M., Casini, S., Innocenti, M., Pollini, L.: Development of a hybrid simulator for underwater vehicles with manipulators. IEEE Journal of Oceanic Engineering **45**(4), 1235–1251 (2020). https://doi.org/10.1109/JOE.2019.2935801
27. Schwenger, M.: Monitoring cyber-physical systems: From design to integration. In: Deshmukh, J., Nickovic, D. (eds.) Runtime Verification - 20th International Conference, RV 2020, Los Angeles, CA, USA, October 6-9, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12399, pp. 87–106. Springer (2020). https://doi.org/10.1007/978-3-030-60508-7_5, https://doi.org/10.1007/978-3-030-60508-7_5
28. da Silva Araujo, H.L., Mousavi, M.R., **Mahsa Varshosaz**: Testing, validation, and verification of robotic and autonomous systems: A systematic review. ACM Transactions on Software Engineering and Methodology **32**(2), 1–61 (2023). https://doi.org/10.1145/3542945, https://doi.org/10.1145/3542945
29. Smith, P., Dunbabin, M.: High-fidelity autonomous surface vehicle simulator for the maritime robotx challenge. IEEE Journal of Oceanic Engineering **44**(2), 310–319 (2019). https://doi.org/10.1109/JOE.2018.2875571
30. Smith, P., Dunbabin, M.: High-fidelity autonomous surface vehicle simulator for the maritime robotx challenge. IEEE Journal of Oceanic Engineering **44**(2), 310–319 (2019). https://doi.org/10.1109/JOE.2018.2875571
31. Watanabe, T., Neves, G., Cerqueira, R., Trocoli, T., Reis, M., Joyeux, S., Albiez, J.: The rock-gazebo integration and a real-time auv simulation. In: 2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR). pp. 132–138 (2015). https://doi.org/10.1109/LARS-SBR.2015.15
32. Xu, H., Li, G., Liu, J.: Reliability analysis of an autonomous underwater vehicle using fault tree. In: 2013 IEEE International Conference on Information and Automation (ICIA). pp. 1165–1170 (2013). https://doi.org/10.1109/ICInfA.2013.6720471

33. Zeng, Z., Sammut, K., Lian, L., He, F., Lammas, A., Tang, Y.: A comparison of optimization techniques for auv path planning in environments with ocean currents. Robotics Auton. Syst. **82**, 61–72 (2016), https://api.semanticscholar.org/CorpusID:205009322