Toward an Accurate Hydrologic Urban Flooding Simulations for Disaster Robotics

Marcelo Paravisi[1,2], Vitor A. M. Jorge[2] and Alexandre Amory[2]
[1]*Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, Brazil*
[2]*School of Technology, PUCRS University, Brazil*

Keywords: Robotics Simulation, Disaster Robotics, Disaster Simulation, Urban Flooding, Hydrological Models, USV.

Abstract: Testing and benchmarking robots in actual disaster scenarios is risky and sometimes nearly impossible. The lack of adequate tests could make robots more vulnerable and less effective in an actual disaster situation. However, even if it was possible to test them in a disaster scenario, the test itself would have high risks for the equipment and the robot operator. For this reason, simulations can be a powerful alternative to validate unmanned systems in safe and controlled virtual environments. The main challenge is to devise an accurate virtual scenario as close as possible to an actual disaster scenario. This problem is particularly harder if the robot in question is an unmanned surface vehicle (USV), mainly due to the numerous disturbances which can affect the robot. This paper presents and discusses the simulation of an urban flooding scenario with accurate environmental disturbances faced by an USV, such as water currents, waves and winds. Results demonstrate that these environmental disturbances have a relevant effect on the USV's ability to perform basic navigational tasks. The main conclusion of this work is that there is a long road ahead of USV simulators in order to validate USVs in realistic disaster scenario simulations.

1 MOTIVATION

Extreme events have terrible effects wherever they occur. Be it a natural disaster or one caused by men, the result is often loss of lives, injured people, as well as the destruction of both the individual possessions (houses, cars, and so on) and the basic utilities infrastructure (water supply, electricity, communications, and hospitals). The result is often homeless people with all sort of basic needs with their lives at risk not only due to their injuries, but also due to hazards imposed by the new harsh and dangerous post-disaster environment, where first responders often must put their lives at stake to reach and rescue others.

Thus, unmanned systems can be a valuable tool for disaster responders by going to places which otherwise would be too dangerous for first responders. As such, rescue robots typically face dull, dirty, and harsh environments with poor infrastructure (e.g. blocked streets, destroyed buildings, poor connectivity, dust, extreme heat, and so on) (Murphy, 2014) and harsh meteorological conditions, such as strong winds, storms, river overflows, large waves, and so on.

Back in the early 2000's, robots started to be deployed for disaster response (Murphy, 2012) and soon it became clear that disaster robotics was a field on its own. Since then, they were used in several disaster sites, including collapsed mines & buildings, earthquakes, tornadoes, landslides, and floods (Murphy, 2012; Murphy, 2014). While some degree of success was achieved, conventional robot design has proven to lack robustness, mainly due to extreme harsh conditions of disaster sites. In 2004, a study on robots used for rescue in urban areas, encompassing 15 robots from three different manufacturers, has shown that the Mean Time Between Failures (MTBF) of rescue robots was 24 hours while their availability was around 54% (Carlson et al., 2004). The study of disaster robotics was on its infancy, but the importance of robots for risky interventions soon became clear (Habib and Baudoin, 2010), along with the extreme challenges faced by robots in harsh environments (Habib and Baudoin, 2010; Wong et al., 2017).

Another challenge preventing the success of disaster robots may be the difficulty to evaluate them properly, since there are very few places to test them which are similar to actual disaster sites. Without proper testing, the robot will likely fail during the actual use in a disaster. The importance of evalu-

Paravisi, M., Jorge, V. and Amory, A.
Toward an Accurate Hydrologic Urban Flooding Simulations for Disaster Robotics.
DOI: 10.5220/0006904704250431
In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2018) - Volume 2*, pages 425-431
ISBN: 978-989-758-321-6

Table 1: UXV Disturbances' Influence.

Unmanned System	Winds	Water Current	Waves	Nearby UXV
UGV	weak	—	—	weak
UAV	strong	—	—	strong (close by)
UUV	—	strong	strong (surface)	moderate (close by)
USV	moderate	strong	strong	moderate

ation and benchmarking for disaster robots in controllable conditions (Murphy et al., 2008) soon became a recognized research problem. Initiatives such as the Robocup-Rescue Project (Takahashi and Tadokoro, 2002) and League (Balakirsky et al., 2007) brought the problem to the spotlight. Actual disaster emulation sites, such as the National Institute of Standards and Technology (NIST)'s Disaster City®(Disaster City encompasses a 52 acre real-world benchmarking environment (Khoury and Kamat, 2009; Wilde et al., 2015).), were designed for testing disaster robots in extreme conditions. Competitions focusing on marine robotics, such as SAUC-E (http://sauc-europe.org/), featuring AUVs, and euRathlon (http://www.eurathlon.eu/), featuring underwater, surface and aerial robots, were designed to test unmanned systems in harsh water environments.

Given such challenging environments and the risks for costly robot platforms and humans, simulations could come as important assets for reducing costs of disaster robotics research, by detecting early problems in robot design and avoiding epic mission failures when human lives are at stake. Even though there is little argument that simulations are easier than real life scenarios (Murphy et al., 2008), when it comes to water environments and unmanned surface vehicles (USVs), the number of freely available simulators is small when compared to other platforms (Torres-Torriti et al., 2016). In fact, when we talk about realistic simulations of harsh water environments this number drops to zero. All that, given the risks associated with water disturbances and the importance of USVs.

USVs are valuable assets for disaster missions involving maritime or flooded environments. They can serve as emergency data/communication relays, or the source of real-time video from disaster sites. In addition, they can perform underwater assessments (through bathymetry), detecting debris, accretion or erosion caused by floods transports. Furthermore, they can serve as emergency carriers of 1st aid kits, potable water, or food. Finally, they can even serve as docking stations and communications base for other unmanned systems. Compared to other platforms, however, USVs are the ones which are influenced by the greater number of environmental factors: winds, waves; and wind currents. In addition, they float over the water surface, which makes USVs easily influenced by other USVs and manned embarkations, even if they are not close to one another, making it challenging to simulate the operation of multiple vehicles simultaneously. Table 1 compares the influence of important environmental disturbances for UGVs, UAVs, UUVs and USVs control.

We collected information about ten open source simulators for marine robots available in the literature (Kelpie (Mendona et al., 2013), USARSim (Nourbakhsh et al., 2005; Sehgal and Cernea, 2010), MARS (Tosik and Maehle, 2014), Stage (Gerkey et al., 2003), MOOS-IVP (Benjamin et al., 2010), UW-Morse (Henriksen et al., 2016), UWSim (Prats et al., 2012), the Gazebo (Koenig and Howard, 2004), "FreeFloating" plugin (Kermorgant, 2014), V-REP (Rohmer et al., 2013)). The number of issues involving currently available simulators include: lack of code availability (Kelpie, UW-Morse); discontinued software (MARS); and outdated libraries (USARSim). In addition, some simulators which do not lack setup problems, still have limited to none built-in simulation of disturbances (Stage and MOOS-IVP). The remaining simulators do have some environmental disturbances simulation capabilities, however none of them are ready to simulate harsh conditions for USVs. Besides, they do not even make it easy to spawn USVs, lacking a ready-to-use USV examples – i.e., USVs must be designed from scratch including model and control features. From Table 1, we see that disturbances are rather important for USVs, even though disturbances in available simulators are limited. None include harsh wave conditions, usually handling calm waters without turbulence. Besides, current wind and water disturbances simulations ignore the influence of the actual terrain relief, preventing physically and hydrologically accurate simulations – e.g., strong winds, river currents, turbulent waters, large waves and so on. All that makes it difficult to validate simulations of USVs in actual disaster sites.

In this paper, we advocate for improved simulations for USVs, both in terms of disturbances and sensor noise to enable minimum proper validation of USV systems. The *goal* of this work is to discuss the

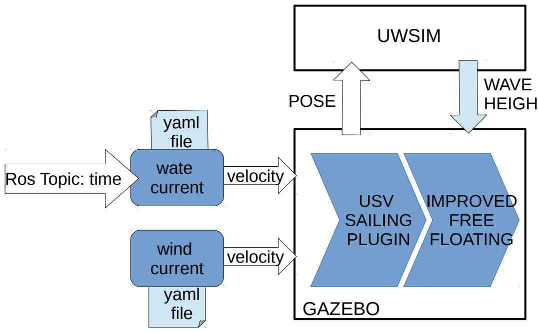


Figure 1: Overall simulation architecture and main modules. The blue modules are new or improved.

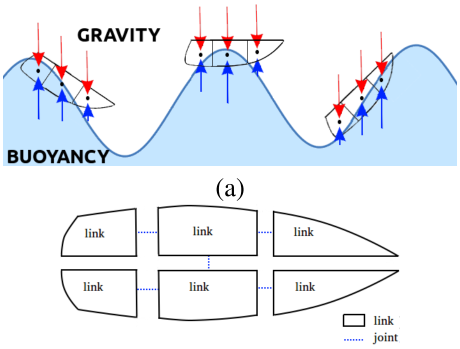
creation of a simulated flooding scenario of an actual urban environment (Porto Alegre, Brazil) and implement an initial version in a robotic simulator. Given a Digital Elevation Model (DSM) of the terrain, hydrological and wind models are generated to accurately reproduce the environmental forces (wind and water current) of a large scale flooded site. We present initial results while evaluating an USV, where experiments show that such disturbances have a strong effect on USV behavior. This work also discusses the numerous gaps and requirements for current and future open-source disaster simulators.

This paper is presented as follows. Section 2 presents the simulation environment. Section 3 describes the construction of the urban scenario, while Section 4 shows a case study of a USV crossing a flooded environment considering winds and water disturbances. Section 5 concludes the paper with a discussion and future work.

2 SIMULATION CAPABILITIES

The present work is based on the USVSim simulator, along with its modules, and the scenarios (freely available for download at https://github.com/disaster-robotics-proalertas/usv_sim_lsa). This section presents an overview of the architecture of the simulator (see Fig. 1) — the blue boxes represent the new or customized simulation modules. The complete description is presented in (Paravisi et al., 2018).

The simulator works a series of Gazebo modules which enable it to simulate environment disturbances for USVs, while UWSim is only used for visualization purposes due to its improved water rendering capability compared to Gazebo. As represented in Fig. 1, the core of Gazebo is not modified but a new modules are included, such as the usv_sailing_plugin. The USV Sailing plugin can simulate the forces of water current over the boat rudder. In order to estimate those forces, it reads the true water velocity from usv_water_plugin. After that, it computes the forces applied to the foil of the rudder, considering the speed of the fluid and the foil to compute the lift and drag forces.



(a)

(b)

Figure 2: The black dots represent the center of buoyancy, while red and blue arrows represent, respectively, the gravity and buoyancy force vectors. In (a), the resulting buoyancy effect of the boat when represented by a set of links and joints (b). Note that the gravity and buoyancy forces are applied to each of the links' center of buoyancy.

When a vehicle pose is updated in UWSim, the wave height relative to the vehicle's center(s) of buoyancy is sent to the Improved Freefloating Gazebo to compute the buoyancy effect. The Improved Freefloating Gazebo plugin allows USVs to suffer roll and pitch orientation changes caused by wave motion. In such strategy, the USV hull must be subdivided into several parts, i.e. modeled by a set of links bounded together by fixed joints (see Fig. 2b). The resulting effect of the gravity (red arrow) and buoyancy (blue arrow) forces is depicted in Fig. 2a, allowing the boat to roll and pitch more naturally, following the shape of the wave.

The water and wind current generators are modeled as ROS nodes which receive requests from Improved FreeFloating Gazebo to enhance the boat motion realism through wind and current information. Besides, the water current generator is used by the usv_sailing_plugin to compute the force which is directly applied to the boat, using as input the velocities of the boat and of the water/wind currents.

The usv_water_current module loads data exported from the HEC RAS (W. Brunner, 1995) hydrological simulator. Thus, users can simulate the flow of rivers by inserting simple height maps, then exporting Hierarchical Data Format (HDF) files, which store the water velocities for each time step of the simulated water flow. Then, the simulation architecture requests the velocity of the water at each of the USV's position of links. Similarly, the usv_wind_current module

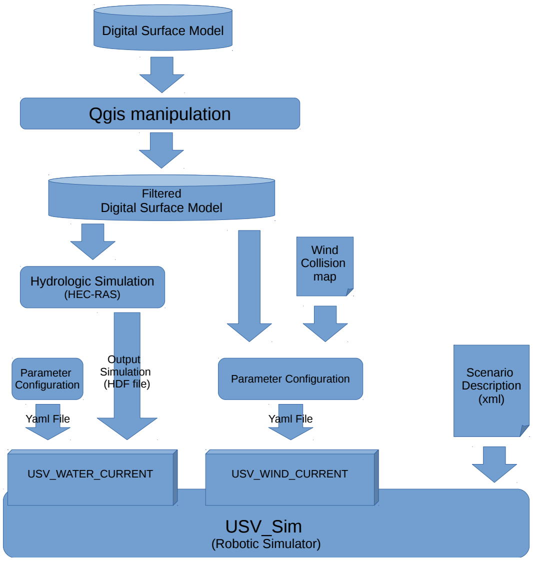


Figure 3: Pre-processing steps to generate hydrological and wind current maps.

simulates the wind over a 2D region with of multiple obstacles using a method based on the fluid dynamics module with the Lattice-Boltzmann method (Qian et al., 1992). The wind force affects the part of the hull above the line of water. The wind collision map is extracted from the scenario.

3 MODELING A DISASTER SCENARIO

Fig. 3 shows the steps required to build the hydrological and wind current maps for a given 3D scenario. To simulate accordingly the flow of rivers with hydrological models, we should consider the direct impact the topology of terrain has in the flow of water. Thus, the first step is to acquire a Digital Surface Model (DSM), i.e. a model that incorporates the topology of the terrain and the shape and heights of buildings. There are some on-line repositories that allows the user download this type of data (https://earthexplorer.usgs.gov/), but in some cases the DSM does not include bathymetry (underwater elevation) information, which can be useful to identify safe routes for boats and USVs.

Eventually, some features of the DSM may have to be ignored (e.g. small trees given the high resolution of the DSM), therefore, the user can edit and filter out the undesired features. We accomplish that using the QGIS(https://qgis.org/en/site/). open-source geographic information system.

Then, the resulting DSM is used as input to the HEC-RAS hydrological simulator. In order to simulate water flowing through the scenario, we have to define a 2D grid over the terrain and the location of the upper and lower river reaches. After that, the user defines the initial water level conditions and, to all river reaches, the user can associate the water volume (m^3) at each simulation time step. This configuration allows the variation of water flow over time. The final step is to run the simulation and export an HDF file, which includes the velocity field for each grid cell over time for the entire simulation of the water flow. This HDF file defined in a configuration file (YAML file) is used as input to the usv_water_current module.

For the simulation of winds, the user should create a configuration file (YAML extension), informing the name of filtered DSM, the wind direction and speed, grid resolution and the filename of an user-defined collision map. This map is a bitmap with the same resolution as the filtered DSM, allowing the user to manually add static obstacles which interfere with wind flow but which were not present in DSM at the time of its creation – e.g. buildings or a large ship. This way, the user can personalize the wind collision map using only an image editor, without the need to resort to DSM or QGIS manual edits.

The top view of the resulting flooding scenario can be seen in Fig. 4a. The water flow comes from the right-hand side of the image, at the location of the Dilvio river, in Porto Alegre, Brazil. The simulated area covers an area of 430 meters by 400 meters. Fig. 4b shows the direction of the water flow and the water velocity in the flooded scenario, the whiter the color, the deepest the channel. While the particles trajectories show the water current speed, so longest particles means that water current is fastest. One can see that the parts close to the original river bed have faster water currents than those nearby the buildings. Also, Fig. 4c shows the wind map of the same region, where heat map indicates the intensity of wind.

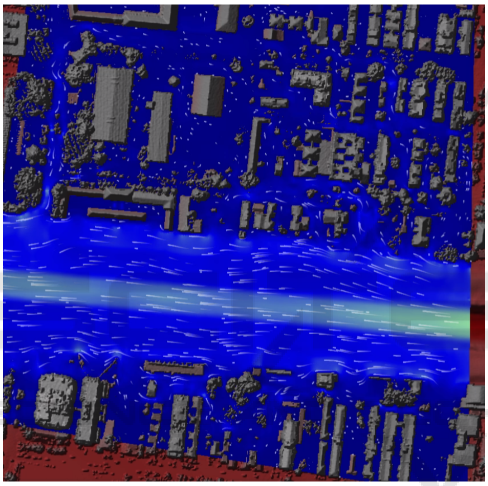
Fig. 5 shows another screenshot from the modeled scenario, but from a point of the view right behind the boat, showing the flooded buildings, trees, and infrastructure.

4 PRELIMINARY RESULTS

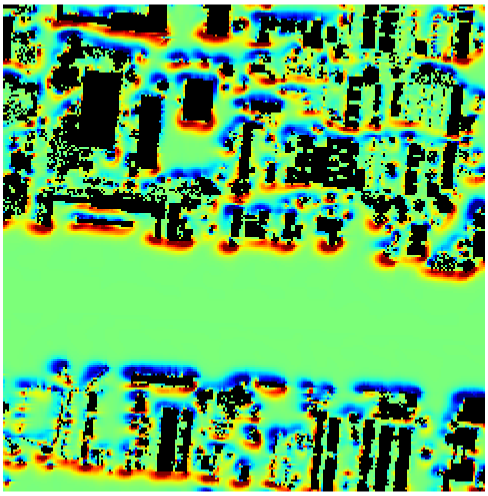
The implemented Guidance, Navigation and Control (GNC) strategy is based on a built-in heading PID controller, allowing them to reach desired positions in the environment. This control strategy computes the angle between the boat position and target(waypoint) position. With this angle, the position of actuators are updated to robot follow the desired orientation.

Fig. 6 shows the different trajectories when the

(a) Aerial photo from simulated scenario.



(b) flooded scenario and water current intensity



(c) the simulated wind current intensity map

Figure 4: Aerial view of the testing site in Google Maps (a), the water current (b) and wind intensity maps (c) for the flooded scenario.

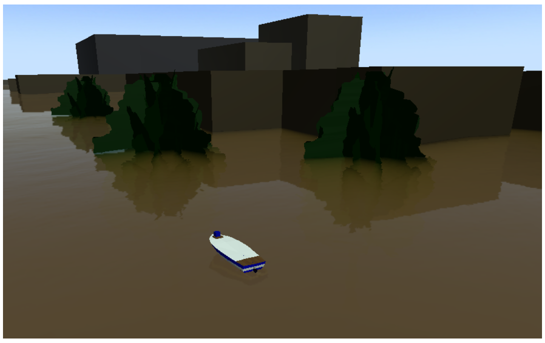


Figure 5: Simulation of the USV at a flooded site.

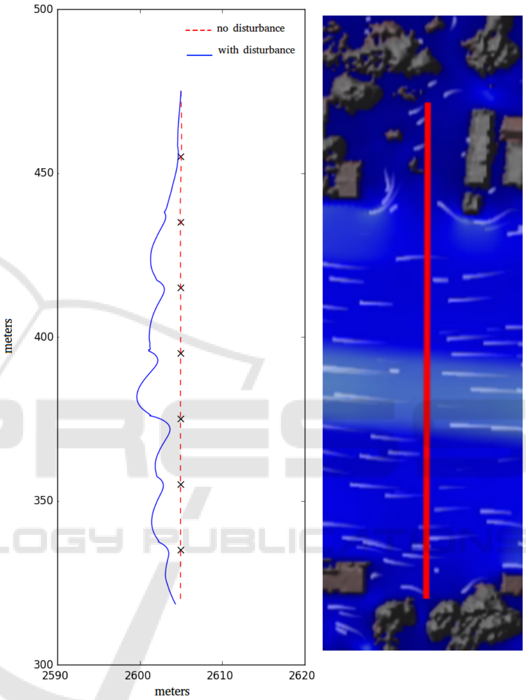


Figure 6: USV trajectories with (red) and without (blue) water & wind currents.

Table 2: Total travel time and distance to destination.

Disturbances	Time (s)	Distance (m)
no	141	157.0
yes	148	167.6

water/wind currents are on and off. Several equally spaced waypoints where inserted along the path of 160 meters. One can see that when the rudder boat is in the fastest water current part the trajectory, error increases considerably. On the other hand, the error reduces as the boat reaches the margins, where the water current is slower. However, the wind changes velocity and orientation near by the buildings.

Tab. 2 shows the time to perform the path and the total traveled distance.

5 DISCUSSION & FUTURE WORK

This paper presents a strategy to model realistic flooding scenarios in a robotic simulator in order to evaluate rescue USVs on harsh environments safely and cheaply when compared to an actual field test in a disaster site. Initial experiments show the importance of proper disturbance modeling and simulation in the control of USVs, which may even prevent the USV from achieving its goal or cause collisions — unacceptable for disaster response missions.

Beyond the presented flooding scenario, we plan to simulate several other types of disasters, such as, the effects of tsunamis or dam breaches. This would allow new GNC algorithms to be evaluated even in such harsh environment conditions.

However, there is much to be done to improve simulations in maritime environments. Current open-source simulators for USVs still lack many features. Proper simulation of sensors and communications problems with USVs in simulations are often limited to Gaussian noise. Instead, intermittent and unreliable services should be considered to validate the robustness of the whole system. These could be achieved by considering factors such as the simulation of wireless transmissions' shadows and GPS errors close to buildings and trees, as well as the effect of water disturbances & weather on sensors and communications. For instance, radio waves can be reflected by the water surface and waves, sporadicly interrupting wireless communications. The same is true for weather conditions and heavy rain, which affects cameras and the range of communications. Finally, underwater data transmission, should also consider physically correct signal attenuation and multi-path simulations. By doing that, the validation of unmanned systems in simulation environments will be more similar to that which UAV, UGV and USVs facing actual disaster missions.

In order to improve the realism of disaster locations, debris could be added to scene, so they could become floating obstacles which could even be carried by the water flow, colliding with USV and UUV. If needed, users could combine those objects together (by adding joints) and defining a limit force that would break their joints. This way, when the grouped object collides with another one with considerable strength, many parts or pieces can be detached generating even more debris deposited all over the city and into water channels, ports and bays. Then, the resulting scenario can be used to emulate post disaster assessment missions, where the amount and location of debris must be estimated by an heterogeneous team of unmanned systems. As a result, new strategies could be designed and properly tested & compared for area coverage, debris detection rate and their removal from affected areas in realistic simulation environments.

Future works may also include the integration of UAVs (affected by winds) and USVs (affected by winds, waves and water currents) collaborating in the same scenario where they look for strained people. We also plan to model landslides, bigger turbulent waves (tsunami like), bridge and oil platform collapses to assess, in simulations, heterogeneous teams of unmanned systems in such marine disaster environments.

ACKNOWLEDGEMENTS

This paper was partially funded by CAPES/Brazil, under project 88887.115590/2015-01, Pro-Alertas program.

The MDT and MDS maps belong to the municipality of Porto Alegre, Brazil. The maps were provided by the Secretary of Municipal Urbanism (Secretaria Municipal de Urbanismo - SMURB) to professor Regis Lahm.

REFERENCES

Balakirsky, S., Carpin, S., Kleiner, A., Lewis, M., Visser, A., Wang, J., and Ziparo, V. A. (2007). Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from robocup rescue. *Journal of Field Robotics*, 24(11-12):943–967.

Benjamin, M. R., Schmidt, H., Newman, P. M., and Leonard, J. J. (2010). Nested autonomy for unmanned marine vehicles with MOOS-IvP. *J. Field Robotics*, 27(6):834–875.

Carlson, J., Murphy, R. R., and Nelson, A. (2004). Follow-up analysis of mobile robot failures. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 5, pages 4987–4994.

Gerkey, B. P., Vaughan, R. T., and Howard, A. (2003). The Player/Stage project: Tools for multi-robot and distributed sensor systems. In *International Conference on Advanced Robotics (ICRA)*, pages 317–323.

Habib, M. K. and Baudoin, Y. (2010). Robot-assisted risky intervention, search, rescue and environmental surveillance. *International Journal of Advanced Robotic Systems*, 7(1):10.

Henriksen, E. H., Schjlberg, I., and Gjersvik, T. B. (2016). UW-MORSE: The underwater modular open robot simulation engine. In *IEEE/OES Autonomous Underwater Vehicles (AUV)*, pages 261–267.

Kermorgant, O. (2014). A dynamic simulator for underwater vehicle-manipulators. In Brugali, D., Broenink, J. F., Kroeger, T., and MacDonald, B. A., editors, *International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, volume 8810 of *Lecture Notes in Computer Science*, pages 25–36. Springer.

Khoury, H. M. and Kamat, V. R. (2009). Evaluation of position tracking technologies for user localization in indoor construction environments. *Automation in Construction*, 18(4):444 – 457.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2149–2154.

Mendona, R., Santana, P., Marques, F., Loureno, A., Silva, J., and Barata, J. (2013). Kelpie: A ROS-based multi-robot simulator for water surface and aerial vehicles. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 3645–3650.

Murphy, R. R. (2012). A decade of rescue robots. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5448–5449.

Murphy, R. R. (2014). *Disaster Robotics*. The MIT Press.

Murphy, R. R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., and Erkmen, A. M. (2008). *Search and Rescue Robotics*, pages 1151–1173. Springer Berlin Heidelberg, Berlin, Heidelberg.

Nourbakhsh, I. R., Sycara, K., Koes, M., Yong, M., Lewis, M., and Burion, S. (2005). Human-robot teaming for search and rescue. *IEEE Pervasive Computing*, 4(1):72–79.

Paravisi, M., Santos, D. H., Jorge, V. A. M., Gonçalves, L. M., and Amory, A. (2018). Unmanned surface vehicle simulator with environmental disturbances. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–7. submitted paper.

Prats, M., Prez, J., Fernndez, J. J., and Sanz, P. J. (2012). An open source tool for simulation and supervision of underwater intervention missions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2577–2582.

Qian, Y. H., D'Humires, D., and Lallemand, P. (1992). Lattice BGK models for navier-stokes equation. *EPL (Europhysics Letters)*, 17(6):479.

Rohmer, E., Singh, S. P. N., and Freese, M. (2013). V-REP: A versatile and scalable robot simulation framework. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1321–1326.

Sehgal, A. and Cernea, D. (2010). A multi-AUV missions simulation framework for the USARSim robotics simulator. In *Mediterranean Conference on Control Automation (MED)*, pages 1188–1193.

Takahashi, T. and Tadokoro, S. (2002). Working with robots in disasters. *IEEE Robotics Automation Magazine*, 9(3):34–39.

Torres-Torriti, M., Arredondo, T., and Castillo-Pizarro, P. (2016). Survey and comparative study of free simulation software for mobile robots. *Robotica*, 34(4):791–822.

Tosik, T. and Maehle, E. (2014). MARS: A simulation environment for marine robotics. In *2014 Oceans - St. John's*, pages 1–7.

W. Brunner, G. (1995). HEC-RAS river analysis system. hydraulic reference manual. version 1.0. Technical report, DTIC.

Wilde, G. A., Murphy, R. R., Shell, D. A., and Marianno, C. M. (2015). A man-packable unmanned surface vehicle for radiation localization and forensics. In *2015 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–6.

Wong, C., Yang, E., Yan, X. T., and Gu, D. (2017). An overview of robotics and autonomous systems for harsh environments. In *2017 23rd International Conference on Automation and Computing (ICAC)*, pages 1–6.