IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 December 2023, accepted 26 January 2024, date of publication 26 February 2024, date of current version 8 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3370443

TOPICAL REVIEW

Underwater Simulators Analysis for Digital Twinning

NICOLÒ CIUCCOLI[1], (Member, IEEE), LAURA SCREPANTI[1], (Member, IEEE), AND DAVID SCARADOZZI[1,2], (Member, IEEE)
[1]Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, 60131 Ancona, Italy
[2]ANcybernetics s.r.l., 60131 Ancona, Italy

Corresponding author: David Scaradozzi (d.scaradozzi@univpm.it)

This work was supported in part by European Erasmus+ programme-Cooperation for Innovation and the Exchange of Good Practices under Grant 2020-1-PL01-KA226-HE-096375.

ABSTRACT The underwater environment is among Earth's most challenging domains, where failures incur elevated risks and costs for both technology and human endeavors. As a consequence, forecasting the behavior of mechatronic systems through simulation has become increasingly important. Underwater Robotic Simulators (URSs) allow researchers and engineers to safely develop and assess submarine systems. The selection of an appropriate URS from the list of available tools is not trivial. Moreover, the integration of this software with the Digital Twin (DT) concept presents numerous advantages, particularly the ability to link the simulated environment with actual underwater vehicles. This connection is facilitated by performing validation and simulation tests using Software In the Loop (SIL), Model In the Loop, and Hardware In the Loop (HIL) techniques. This paper extensively reviews URSs in the context of both robots and unmanned vehicles in light of the DT paradigm. The article critically examines distinctions among existing URSs, offering valuable insights to aid researchers in selecting the most fitting tool for their specific applications. Additionally, the review explores the practical applications of the identified simulators, categorizing their usage across different fields to illuminate the preferences within the scientific community and showcase prominent case studies.

INDEX TERMS Underwater simulator, underwater vehicle, digital twin, manipulation, cooperation, cyber-physical systems.

I. INTRODUCTION

Understanding the behavior of systems prior to entering a complex scenario has become vital for many projects and applications. Simulations offer a way to forecast the behavior of robotic systems. The popularity of simulators has increased over time, thanks to their ability to offer an accelerated and secure avenue for the development, verification, and testing of robotic control algorithms and prototypes. Having a digital copy of the system in a reliable software tool result in understanding safely, faster, and at a lower cost its dynamics, its interactions with the surrounding environment and the budget associated with its adoption in a specific application. Despite the availability of numerous simulators to industry and researchers, selecting the most suitable one remains a non-trivial task. In various scenario, a diverse array of physics simulators proves suitable, and the introduction of the Digital Twin (DT) concept raised new interest in their development and utilization. While simulation generally refers to a digital model emulating the operations or processes within a system, the DT represents the digital counterpart of physical or non-physical processes, systems, or objects. Moreover, it integrates all data produced or associated with the process or system it mirrors. Thus, the DT enables the digital representation of a process, system or object within its ecosystem, reflecting the data transfer that occurs in the real world and reproducing it in real-time. Thus, the real-time digital representation provided by a DT allows for the execution of various simulations, including Software In the Loop (SIL), Model In the Loop, and Hardware In the

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

Loop (HIL), collectively referred to as xIL. It is crucial to note that not all simulators can be considered, a priori, as tools for DT; only those designed in accordance with the DT paradigm are suitable [1]. This paradigm delineates an architecture abstracted into three primary levels: physical, network, and computational. The physical entities constitute the physical layer, the network layer is the part that connects the physical entities with the virtual ones, and the simulation of the virtual entities represents the computational level, as in the classical paradigm of environment simulation. Not all known and used simulators in marine robotics are ready for this new design strategy. To guide the selection of an appropriate tool for the Digital Twinning of a marine system, this study delineates the essential characteristics that simulators must possess:

- Ability to simulate the agents (processes), including the iterations with the environment and its changes due to agents' movements;
- Ability to simulate sensors readings in the environment;
- Ability to simulate actuators' actions in the environment;
- Ability to model materials characteristics, Multiphysics interactions between rigid bodies and the possibility of introducing low-level control algorithms and strategies;
- Ability to interact with the real prototype and/or with other simulators in a network
- Ability to extract data from a historical database of real field interactions and replicate the behaviour in simulations.

The present paper aims to analyze the literature on underwater simulators, highlighting deficiencies, features, ongoing research and practical applications. The goal is to offer fundamental guidelines for selecting an underwater simulator tailored to the specific application needs of the reader. Despite the numerous options presented by the market and scientific literature [2], the field of underwater robotics simulators has not yet attained a level of maturity comparable to terrestrial, aerial [2] and space robotics [3] applications. Unlike these counterparts, which benefit from several simulators supporting specific tasks, marine applications involving vehicles at and below the water surface [4] encounter limited support. The scarcity of tools in marine simulation can be attributed partly to the difficulty of modelling and computing all the dynamics in such a complex environment. Nonetheless, the lack of a de-facto standard in underwater and marine robotics simulators may slow down the process of building reliable underwater and surface vehicles that can be promptly and safely deployed for diverse applications. To establish whether there is a de-facto standard in simulators designed for robotic vehicles in a marine environment, it is essential to recognize that the physics governing the systems operating at the water's surface differ from those operating beneath it. Although fundamental physical principles remain consistent above and below the surface of the water, they necessitates adaptation to account for two distinct conditions, resulting in two separate sets of applications. In the present paper, only simulators of underwater robotic vehicles are considered. The paper is organized as follows: Section II provides a concise explanation of robotics simulators, covering their prevalent usage in ground, aerial, and marine applications, also in connection with the development of a DT ecosystem. Building upon the definition established in Section II regarding underwater environment simulators, Section III delineates the methodology employed by the researchers in conducting the literature search. Section IV details the results of the literature search, presenting a comparative analysis of the properties of each mathematical model underpinning the simulators and the associated measurements. In Section V, the paper explores how the mathematical models are implemented in studies, highlighting sensor properties employed for measuring pertinent variables. Section VI showcases the practical application of simulators through real case studies. The concluding Section VII encapsulates the authors' final considerations based on the findings derived from the literature search.

II. OVERVIEW OF ROBOTIC SIMULATORS AND THEIR RELEVANCE TO THE DEVELOPMENT OF A DT

a Robotic simulator designed to replicate the behavior or structure of a physical machine. Simulation tools enable the digital recreation of behavior in a certain application independent of the software actually running on the machine. Nowadays, a simulator usually includes all or part of the following elements:

- A graphic engine for the 3D visualization of the robot and its surrounding environment;
- Sensor models, not necessarily with the possibility to accurately represent the physical features of the sensor;
- Actuator models, not necessarily with the possibility to accurately represent the physical features of the actuator;
- A physics engine that simulates the robot dynamics, not necessarily including the dynamic interactions with the environment;
- Application Programming Interfaces (APIs) for the introduction of new sensors, actuators, and control systems and, possibly, interfacing them with external environments,
- Already-existing control strategies of robots.

The most cited simulator in the research field is Gazebo [5], developed in 2002 by Andrew Howard and Nate Koenig at the University of Southern California. Gazebo uses OGRE3D as a rendering engine, can be compiled with four physical engines, and uses the Robot Operating System (ROS) middleware to maintain synchronization across different robots (either real or simulated). ROS also acts as an interface for external software with the possibility to simulate tests in xIL. Gazebo is used for the simulations of various types of robots: cooperative [6], aerial [7], and ground vehicles [8]. Other important simulators employed for simulating robots in different environments include Coppelia Sim [9] and Morse [10]. Coppelia Sim is a newer version of V-rep, which was born in 2013. It uses two types of rendering engines, one based on OpenGL and one based on POV-Ray, where OpenGL presents low-quality graphics but a higher frame rate than POV-Ray. V-rep also supports the same physics engines as Gazebo. Some applications of Coppelia Sim include the control of a mobile robot using a deep learning approach [11], the validation of a free real-time path planning collision algorithm of an aerial vehicle [12], and the development of algorithms of a humanoid robot for its collaboration with a human subject [13]. The Morse simulator was developed by the Laboratoire d'Analyse et d'Architecture des Systemes at the University of Toulouse. It uses Bullet Engine and Blender Game Engine as the physics and graphics engine, respectively. Researchers usually use Morse for the simulation of a complex robotic scenario with the ground and aerial vehicle [14], and for their control as well [15]. The simulation of underwater vehicles necessitates additional requirements due to the diverse range of vehicles (e.g. Autonomous Underwater Vehicle (AUV) or Remote Operated Vehicle (ROV)), each equipped with various functionalities (e.g. integrating a manipulator, as it is the case of Underwater Vehicles With Manipulator (UVMS)) and operating within the underwater environment. The peculiarities of the underwater environment mandate the inclusion of features beyond those typically associated with existing robotic simulators, such as:

- the physics engine must be able to model the interactions of the water particles with the robot structure to simulate the hydrostatic and hydrodynamic effects, such as buoyancy, added masses, viscous friction and thrusters; generally, at least part of these effects is taken into account and realized by adding a plugin to the basic simulator software;
- the simulator must be able to model the effects of the waves and currents which are present in the underwater environment;
- the simulator must be able to graphically represent the effects of the water, such as turbidity, lack of visibility or the presence of lights in vehicles;
- the implementation of underwater specific sensors, such as various types of sonar, Doppler Velocity Log (DVL), acoustic positioning systems, etc.

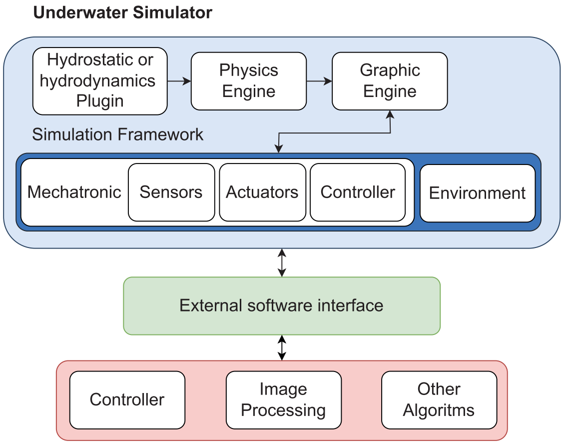


FIGURE 1. General architecture of underwater simulator.

The general architecture of an underwater simulator is outlined in figure 1. The simulation framework serves as the segment of the software connecting the graphics engine with the physics engine. It encompasses all elements developed within the simulator, including robots, sensors, actuators, and more. In many cases, this function is carried out by the middleware, serving not only as an external interface but connecting all the components inside the simulator. A robotic simulator represents an excellent ally for developing, testing and validating products and processes. Moreover, when designed in accordance with the essential characteristic of DT, it is possible to connect the virtual twin with the physical device in the real environment.This integration enables the utilization of a database containing field collected data for xiL validations. The concept of simulation has recently been related to DT because they both use digital models to replicate products and processes. Despite this connection, notable distinctions exist between the two approaches. Notably, a DT represents a virtual, real-time counterpart of a physical object or process. In this sense, representing a problem using a DTand matching it with the real devices allows the study of the entire infrastructure with the Cyber-Physical System (CPS) paradigm. The comprehensive nature of the DT enhances the simulation process, contrasting with the conventional simulation's focus on a singular process. Where a simulation generally refers to a single process, the DT refers to the whole environment in which a set of processes that can be said "to belong to the same family (CPS)". Moreover, in a DT sensor data are obtained in a real-time fashion enabling the exploration of a broader spectrum of issues compared to traditional simulations. The DT idea was born in 1991 by David Gelernter [16]. However, the concept was introduced and applied for the first time in the manufacturing industry by Michael Grieves, who defined it "the virtual digital expression equivalent to physical products" [17]. In robotics, particularly in the field of robotic vehicles, the most appropriate definition for DT is the one provided in 2010 by NASA: "DT is an integrated multiscale, multi-physics, probabilistic simulation of a system that uses the best available physical models, sensor updates, etc. to mirror the life of its flying twin" [18]. In recent years, many DT concepts and definitions have been proposed based on the scope of the different applications. Table 1 provides a summary of distinct concepts and perspectives on DT. A DT generates a virtual environment capable of conducting several simulations, leveraging real-time data and establishing a mutual flow of information between the twin and the sensors collecting data. This implies that, while a DT begins similarly to a simulation model, the introduction of real-time data enables, time by time, the twin to change its status. Through the continuous collection and analysis of data, the DT provides a more dynamic simulation, presenting varied information not attainable with a static simulation. Furthermore, whereas a simulation is theoretical because it replicates what could happen to a generic object or process, a DT simulates the agents or processes in real-time incorporating specific data that allow reproducing the actual agents or process status. This enriched infrastructure can also be improved with a Machine Learning supervisor either functioning as the DT or operating within the real environment.

III. MATERIALS AND METHODS

The literature review, constituting the primary focus of this study, was undertaken to examine academic simulators for underwater vehicles. The research question guiding the authors aimed to discern the most widely utilized academic simulators for underwater vehicles and identify which among them is conducive to the establishment of Digital Twins (DT) in an underwater environment. Publications were identified using an electronic search strategy of relevant ocean engineering academic field databases and a subsequent search based on the reference lists of identified papers. Relevant databases were IEEE Explorer, Science Direct and Scopus. Keywords for the search were: underwater simulators, Unmanned Underwater Vehicle, underwater vehicle, robotic intervention, and underwater vehicle manipulator systems. The inclusion criteria were:

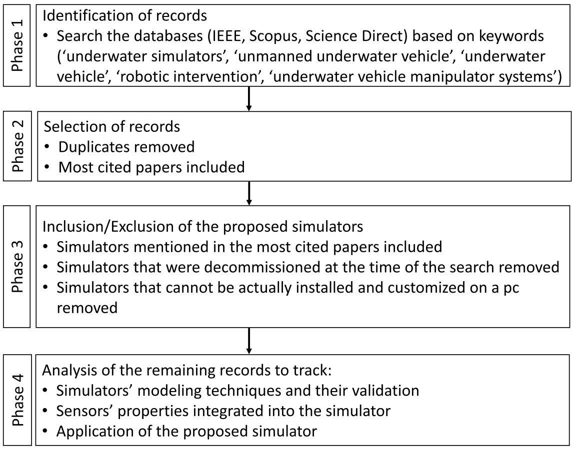
- Records describing a simulator that was developed in the last ten years,
- Records describing a simulator that was not decommissioned,
- Records describing a simulator that has an online repository from which to download and install the simulator on a computer

The exclusion criteria were:

- Records describing a simulator that was developed more than ten years ago,
- Records describing a simulator that was decommissioned,
- Records describing a simulator that has not an online repository from which to download and install the simulator on a computer

The whole process of the literature search is represented in figure 2. Within this context, a simulator is considered popular when it records at least 50 citations.

The popularity criterion was analyzed by counting all citations of the study as reported on Scopus. In figure 3 the search results after applying all the criteria are shown by the number of citations each simulator received. Figure 3 highlights the most mentioned simulators in the academic scientific literature during the last ten year. The term ‘‘academic’’ refers to open-source software used in at least one scientific publication. The total number of selected records for the present review is 127: 29 are journal papers, and 98 are conference papers. These results were then analyzed to determine how the simulators were used, which techniques made it possible to develop and/or validate them, and which application they were used for. Additionally, the present review will briefly mention the new simulators currently under development to provide a full picture of the existing scenario of underwater simulators and discuss them in light of the latest trends in research.



Identification of records
- Search the databases (IEEE, Scopus, Science Direct) based on keywords (‘underwater simulators’, ‘unmanned underwater vehicle’, ‘underwater vehicle’, ‘robotic intervention’, ‘underwater vehicle manipulator systems’)

Selection of records
- Duplicates removed
- Most cited papers included

Inclusion/Exclusion of the proposed simulators
- Simulators mentioned in the most cited papers included
- Simulators that were decommissioned at the time of the search removed
- Simulators that cannot be actually installed and customized on a pc removed

Analysis of the remaining records to track:
- Simulators’ modeling techniques and their validation
- Sensors’ properties integrated into the simulator
- Application of the proposed simulator

FIGURE 2. **The four phases of the review process.**

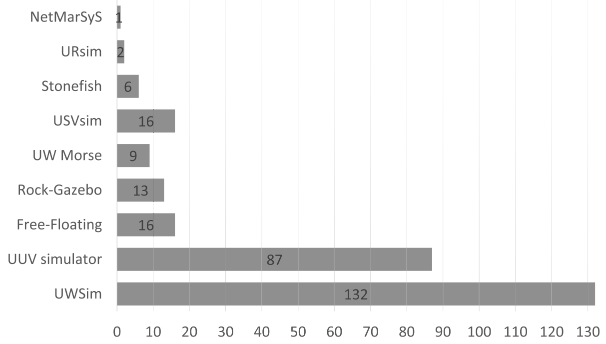


FIGURE 3. **Citations counted for reviewed underwater simulators.**

IV. SIMULATORS OF UNDERWATER VEHICLES

The results of the literature search described in section III show that the most cited simulators are UWsim [25] and UUV simulator [26] (figure 3). Other underwater simulators are represented as well. In particular, some of those that were born as extensions of Gazebo and Morse for considering the marine environment, i.e. Free-Floating [27], Rock-Gazebo [28] and UW Morse [29]. However, they implement fewer features than UWsim and UUV. In the following subsections, a summary of each simulator’s main features is provided to deliver an overview of the capabilities of these tools.

TABLE 1. Digital twin definitions from literature.

Authors	Year	Concept
Glaessgen and Stargel [19]	2012	A DT is a probabilistic, multiscale and multiphysics simulation of a system, which uses the best mathematical models and the best technologies available to replicate the life of the real twin
Rosen et Al. [20]	2014	DT is a type of life cycle management through model and simulation, which include the state and historical information of the aircraft when manufacturing. It is used to realize the high-fidelity modeling of the aircraft in the full life cycle
Grieves [21]	2014	A DT is made up of three parts: a physical product in real space, a virtual product in virtual space, the two-way communication of data and information that binds the two products together
Schluse et al. [22]	2016	DT is a virtual substitution for the real world. It contains a virtual presentation and communication capabilities, constituting smart objects as intelligent nodes, within the IoT
Tao et al. [23]	2018	A DT is made up of three parts: physical product, virtual product and a connection that binds them together. The characteristics that a Digital Twin must have are: replication in real time and the virtual space must maintain a high level of synchronization and fidelity to the real space
Wu et al. [24]	2020	DT is a digital replica of a physical entity, with the close connection between the two

A. UWsim

UWSim[1] [25] is the first simulator for underwater vehicles. It was developed in 2012 inside the Interactive and Robotic Systems Lab project at Jaume-I University. It utilizes respectively OpenScene-Graph(OSG) and Bullet Engine as graphics and physics engine. UWsim allows the simulation of underwater vehicles with a manipulators system (UVMS), particularly the GIRONA 500 I-AUV vehicle equipped with different manipulators, even if only the kinematic model is present. In UWsim, hydrodynamic forces on the manipulator, like multi-body interactions, are not modelled. The mathematical model of the vehicle and thrusters is also quite simplified, even if it implements sea currents and waves. In order to compensate for this lack, in [30] the possibility of simulating vehicles dynamics in Simulink and of the manipulators has been integrated through the Simurv library. UWSim uses ROS as an interface for external applications such as control systems or sensors, allowing it to conduct tests in SIL/HIL on vehicles that implement the same interfaces. All sensors are implemented as ROS nodes in line with many real vehicles, in particular IMU, DVL, Positioning systems, Cameras and different types of Sonar. It also allows the simulation of multiple vehicles and Unmanned Surface Vehicles, even if this procedure is quite tricky because objects in the scene must be created manually as all the ROS interfaces for each vehicle with scarce documentation. Furthermore, in [31], [32], and [33] the authors present UWSim as a tool for benchmarking underwater intervention and a simulation framework to compare different algorithms that share a common robotic platform and also to evaluate position controllers under the influence of sea currents. UWSim has a very realistic rendering that can be heavily exploited in 3D reconstruction applications [34]. A web server with a graphical interface has also been developed to let external users design their specific experiments on autonomous underwater interventions [35]. Specifically, the user can set the simulation parameters (for example, the object tracker algorithm, current visibility, PID controller coefficients and tracking algorithms) and acquire the dataset.

B. UUV SIMULATOR

The Unmanned Underwater Vehicle (UUV) Simulator[2] [26] is an open-source plugin that adapts Gazebo to the underwater environment. It was developed within the Smart and Networking Underwater Robots in Cooperation Meshes (SWARMs) project, and it let the user simulate only submarine vehicles. Even if it does not allow the simulation of surface vehicles, UUV Simulator lets the user develop and test underwater vehicles equipped with a manipulator. Since UUV Simulator is an extension of Gazebo, it exploits its graphics engine, even if it does not have a realistic rendering and cannot modify the water properties. On the one hand, this open-source plugin faithfully models all hydrostatic and hydrodynamic effects of vehicles, including thrusters and fins. On the other hand, it does not model the forces produced by umbilical cables in Remotely Operated Vehicles (ROVs) and buoyancy and hydrodynamic effects for the manipulator systems. Like UWSim, UUV Simulator uses ROS as an external interface to create new control modules and other features. Also, many sensors have been modelled within this plugin: IMU, DVL magnetometer, pressure sensors, cameras and sonar. Furthermore, various underwater environments, such as the sea and lakes, also present features like, for example, wrecks are also proposed.

C. FREE-FLOATING

Free-Floating plugin connects UWSim and Gazebo through ROS [27]. Kermorgant created it to exploit the ability of Gazebo to simulate UVMS and the potential of the UWSim graphical rendering. However, there are some limitations a user must consider when using this plugin. First, dealing with stability problems, it is impossible to include the added masses in the model. Second, UUV Simulator uses Gazebo to simulate the manipulators without the hydrodynamic effects.

D. ROCK-GAZEBO

Within the FlatFish project, it was developed the Rock-Gazebo package[3] [28] to integrate Robot Construction Kit

---

[1] http://www.irs.uji.es/uwsim/

[2] https://uuvsimulator.github.io/

[3] https://uuvsimulator.github.io/

(Rock) framework and Gazebo to create a platform for real-time HIL simulations. Rock-Gazebo uses OpenSceneGraph (OSG), the same as UWsim, as the graphics engine, particularly for modelling water's visual effects. However, this framework does not allow for simulation manipulators and surface vehicles and is not a multi-robot tool.

E. UW MORSE

UW Morse[4] [29] is an open-source expansion of Morse that adapts it to the underwater environment. It maintains all the properties of the original simulator, such as Blender Game Engine for graphic rendering, Bullet as a physical engine and an interface for the various middleware on which controller nodes or new sensors can be implemented to do SIL. The mathematical solver implements all water hydrostatic and hydrodynamic effects, including sea currents. In addition to the classic sensors for ground vehicles, typical sensors of the marine environment are modelled: acoustic positioning sensors (LBL, USBL), 360 ° Scanning Sonar, Echosounder Altimeter and pressure sensors. Manipulators, surface vehicles and multi-robot simulations have not yet been implemented in UW Morse.

F. StoneFish

Among the most recently developed simulators, Stonefish[5] [36] represents the most valid alternative to UWSim and UUV simulators for the features that have been implemented. The framework is mainly composed of a library containing the Bullet physics engine, the graphical interface, not created through a graphics engine but directly in OpenGL, and a ROS package for sensors, controller and external interface. The unique feature of this simulator is its simulation accuracy due to the calculation of the hydrostatic and hydrodynamic effects of the vehicle geometry. The solver solution is based on representing each rigid body as a polyhedron. The multi-body dynamics between vehicle and manipulator, like in Featherstone [37], and several sensors are supported. In addition, its realistic rendering models various graphic effects: underwater lighting absorption, scattering, air-light, shadow and ocean surface. The disadvantage of the StoneFish simulator is the high computational cost of the simulation and, therefore, the inability to simulate a large number of vehicles.

G. URSim

Unity ROS Simulator (URSim)[6] [38] is the first successful attempt among competitors to use the Unity 3D game engine and ROS as an interface for external nodes. Physics is modelled directly within unity in C#, including drag, and buoyancy, while the physical parameters of the vehicle can be entered through the Unity3D's Rigidbody component. IMU, cameras and Pressure Sensors are also modelled with the possibility of adding Gaussian Noise to the output data. This simulator is still developing, and other sensors, like acoustic communication and underwater current, are planned to be included in the next release.

H. OTHERS

Unmanned Surface Vehicle simulator (USVSim) [39] is the only simulator dedicated exclusively to simulating surface vehicles. It consists of a Gazebo plugin to simulate vehicles in a disaster scenario. This means that wind, currents, and waves are factors considered, especially in validating robust control algorithms with the presence of disturbances. USVSim incorporates the Foil Dynamics Plugin to simulate lift and drag during vehicle advance generated by foil, which is a very common actuator in the surface environment.

Networked Marine Systems Simulator (NetMarSyS) [40] is a platform for the 3D simulation of both submarine and surface vehicles individually and in cooperating applications. It comprises a Unity 3D Game Engine for visualization and a webserver socket that communicates through ROS with independent units. This architecture differs from the previous ones because it does not use a physics engine for simulations. However, the units directly simulate vehicles and sensors, making it less costly at a computational level, even if less precise. This platform is in the early stage of development, and more complex mechanical systems will be integrated using existing simulators.

A new simulator for submarine vehicles, particularly ROVs, is also described in [41]. The simulator aims to describe the vehicle behaviour during the reconnaissance tasks as closely as possible. The simulator is based on Choreonoid, an open-source integrated GUI software for robots, an environment that allows users to extend the interface with their functions. In particular, two plugins have been created, one to include all the hydrostatic and hydrodynamic forces/torques that the vehicles are subject to, and the other to acquire a more truthful image from the camera subject to noise and distortion.

Wang et al. [42] proposed a new general simulation platform for USVs autonomous learning to generate data and test control algorithms. This software comprises six modules: Water, Environment, Infrastructure, Vessel, Sensors, Data generation and analysis. Vehicle rendering and physics are realistic, but disturbances such as wind and waves are not included. The novelty introduced by the simulator in [42] is the ability to collect data from different sensors to train neural networks for subsequent validation within the platform. Another less cited but exciting application of a simulator to the underwater environment uses Unreal Engine and V-Rep simulators to validate specific components of submarine vehicles. Authors in [43] developed and validated a new high-fidelity model of Forward-Looking Sonar for autonomous submarine vehicles. Ganoni et al. [44] have developed a new simulator for ROVs capable of simulating the forces generated by the umbilical cable using an Unreal

---

[4] https://github.com/eirikhex/UW-MORSE

[5] https://stonefish.readthedocs.io/en/latest/

[6] https://github.com/srmauvsoftware/URSim

Engine. The authors did not rely on a physics engine, so it cannot simulate contact physics. Moreover, cables are modelled as a set of connected particles. Also, [45] proposed a new simulator for UVMS consisting of V-rep/Simulink. The double platform choice is due, as in UWSim, to the choice of modelling the hydrodynamic effects, the thruster model, the manipulator, and Guidance Navigation Control systems externally. On the other hand, V-rep is used in [46] to develop a framework called Autonomous Marine Surface Vessel Simulator (AMSVS) to simulate surface vehicles during the Maritime RobotX Challenge. As for the challenge, it was an essential requirement recreating an accurate 3D model of the environment, the sensors that the USVs are equipped with (IMU, GPS, Camera and Lidar), as well as the physics of the vehicle, including waves are so accurate.

Table 2 summarizes the main capabilities of the simulators and describes the extent to which the feature is present or well represented: “0” means that the simulator does not contain that feature, “1” means that it contains the feature only partially or there is no documentation about that feature, and “2” means that the feature is implemented and it is well documented.

V. MATHEMATICAL MODELLING CAPABILITIES IN IDENTIFIED SIMULATORS

underwater simulations are vital for many applications. The analysis of the existing literature highlighted two main categories of applications: model the behaviour and simulate sensors payload. The first category aims to verify the agents’ compliance with the desired one. This category includes all types of control, path planning, cooperation, etc. The second category includes applications that involve processing sensors’ measurements. In this respect, it is essential to have a reliable and accurate sensor model. Typical case studies in this category are Simulations Localization and Mapping (SLAM) algorithms, 3D Reconstruction, etc. The applications of this category use the simulator both as a data collection tool and validation tool in case the algorithms can be implemented in real-time. To achieve an acceptable performance of the DT infrastructure in all the applications, the virtual twin must include a mathematical model of the vehicle as faithful as possible to reality and a mathematical model of sensors and actuators. The following subparagraph will provide the basic mathematical model for the physics behind all the previously mentioned simulators, highlighting their similarities and differences. Moreover, an overview of what is needed to analyze and integrate the desired model of a sensor is provided.

A. MATHEMATICAL MODEL

Generally, rigid bodies, especially vehicles, are systems with 6 degrees of freedom (DOF). The position is described with respect to an inertial reference system and the attitude through the Euler angles by the variable $\eta = x, y, z, \phi, \theta, \psi$. Velocity with respect to the body frame is identified by $\nu = u, v, w, p, q, r$. The physics engines implemented in the simulator, in particular, Bullet [26], integrate the equation that describes the dynamics of any robot:

$$M_{RB}\dot{\nu} + C_{RB}(\nu)\nu + W(\eta) = \tau_g \tag{1}$$

where M_{RB} is the rigid-body mass matrix, C_{RB} is the rigid-body Coriolis matrix, $W(\eta)$ is the force of gravity vector and τ_g are the external forces and torque acting on the body. Generally, τ_g is generated by the actuators, but it can also include other factors, all of which can be calculated by external software. In the underwater environment also, some other effects, like, i.e. friction, need to be considered [47]:

$$M_{\mathrm{RB}}\nu + C_{\mathrm{RB}}(\nu)\nu + M_{\mathrm{A}}\dot{\nu_r} + C_{\mathrm{A}}(\nu_r)\nu_r + g(\eta) + D(\nu_r)\nu_r = \tau \tag{2}$$

where M_{A} and $C_{\mathrm{A}}(\nu_r)$ are respectively the added-mass matrix and Coriolis matrix including added mass, $g(\eta) = (W - B)(\eta)$ is restoring force and moment vector which includes the force of gravity and buoyancy force, $D(\nu_r)$ the dumping matrix and τ includes only the control force and torque. In this equation, the velocity vector includes the subscript r to highlight that the model considers the body’s velocity, excluding sea current. Generally, physics engines cannot calculate these contributions because they would require a very high computational cost caused by the computations required to represent the iterations between water particles and the vehicle. Therefore, to account for these effects in the simulation, they are included in 1 in the vector of external forces τ_g:

$$\tau_g = -M_{\mathrm{A}}\dot{\nu_r} - C_{\mathrm{A}}(\nu_r)\nu_r + B(\eta) - D(\nu_r)\nu_r + \tau \tag{3}$$

where $B(\eta)$ is the buoyancy force vector.

It can be seen from 3 that the acceleration is included within the vector of external forces, which was used to calculate the current acceleration. Thus, the choice of the integration method and the sample time can lead to instability. For this reason, in Free-Floating, these effects are not considered in the simulations. At the same time, in UUV Simulator, a low pass filter is included to obtain the previous acceleration, which is used in 3. In Rock-Gazebo a new method for considering added masses was presented [48]. In particular, the method consists in calculating the compensated effort $C = -M_{\mathrm{A}}\dot{\nu_r}$ as a function of the other forces applied to the vehicle and of the mass matrices; moreover, stability is guaranteed if $||M_{\mathrm{RB}}^{-1}M_{\mathrm{A}}||_2 < 1$.

B. ACTUATION

In the underwater environment, the thruster is the actuation present in almost all vehicles, so its modelling is a fundamental point for the validity of simulated vehicles. Since it is impossible for these simulators to simulate the iteration of the propeller with the water, all simulators require the knowledge of some parameters of the thruster to obtain the most realistic behaviour. The best way of modelling the thrusters is the one a propeller is identified with a function that converts the rotational velocity into thrust. The most

TABLE 2. Underwater robotics simulators' characteristics comparison [0 not present; 1 claimed; 2 present].

Simulator	Realistic Simulation	Thruster	Fins	Realistic Rendering	Water Property	Contact Dynamics	Waves	Water Current	Multi robot	Manipulator	USV
UWsim	2	2	0	2 (osgOcean)	2	2 (osgBullet)	2	2	2	Kinematics	1
UUV	2	2	2	0 (Gazebo)	0	2 (Gazebo)	0	2	2	No Buoyancy, Hydrodinamics	0
Rock-Gazebo	2	2	0	2 (OSG)	2	2 (Gazebo)	0	2	0	0	0
Free-Floating	No Added Mass	2	0	2 (UWsim)	2	2 (Gazebo)	0	0	0	1	0
UW Morse	2	2	0	2 (BGE)	0	2 (Bullet)	0	2	0	0	0
Stonefish	2	2	0	2 (OpenGL based)	2	2 (Bullet)	2	2	2	Featherston	0
URsim	2	2	0	2 (Unity)	2	0	2	0	0	0	0

commonly used models for propellers are the zero, first order system (Gain), Yoerger and Bessa model [49]. The thrust, instead, is usually modelled with the following quadratic form:

$$T(\Omega) = K\Omega|\Omega| \tag{4}$$

where K is the thrust coefficient which can be different according to the direction of rotation, and Ω is the rotation speed of the propeller. An essential component to add to this model is the so-called dead zone:

$$T(\Omega) = \begin{cases} K_L(\Omega|\Omega| - \delta_L) & \text{if } \Omega|\Omega| < \delta_L \\ K_R(\Omega|\Omega| - \delta_R) & \text{if } \Omega|\Omega| > \delta_R \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

UUV follows this philosophy exactly. Free-Floating and Rock Gazebo use those provided by the Gazebo plugin. Stonefish created its engine model, which does not include the dead zone and the propeller model among the coefficients but foresees the torque generated due to the propeller's rotation.

C. SENSORS

The fundamental sensors for underwater vehicles can be divided into two broad categories: sensors that measure vehicle properties and sensors that measure properties of the environment. Examples of the former are accelerometers, gyroscopes and compasses, the inertial measurement unit (IMU), DVL and acoustic positioning systems like Ultra Short Baseline (USBL). Examples of the latter category are cameras, sonars and lasers. For the sensors that measure vehicle properties, it is essential to model the following characteristics: Sampling rate, Bias, Non-linearity, White Noise, Dead-Zone, and Resolution. These sensors are necessary for all applications, from control and path planning to SLAM algorithms. On the opposite, for sensors that measure properties of the environment, like detection of objects or measurements of distances, it is necessary to focus on the fundamental properties of the measure involved. For the cameras, for instance, it is important to carefully consider the resolution, pixel noise and distortion because these factors deeply influence the quality of the images, therefore their closeness to reality. Finally, given the importance that sensors have in the physical and virtual environment and that a wide variety of sensors dramatically increases the possible fields of application, an analysis of the available sensors on each simulator was carried out, and the results are summarized in Table 3.

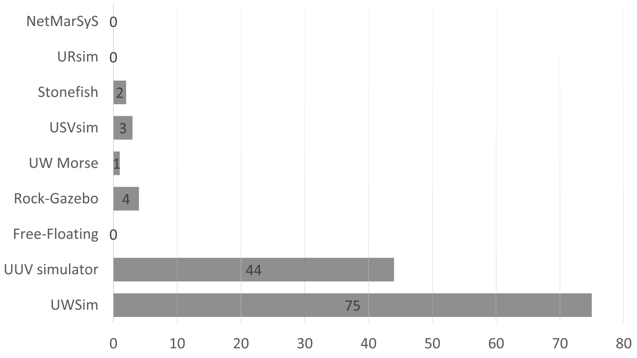


FIGURE 4. Use cases for each reviewed underwater simulator.

VI. ANALYSIS OF USE CASES AND APPLICATIONS ADDRESSED

This section shows the results of analysing the applications of the simulators that were presented previously. Firstly, the search results tested whether the simulator was used in a real case study and/or how many applications each simulator had. Results are shown in figure 4. As expected, the most cited simulators are also the most used. The main fields of application can be summarized as:

- *Control and/or simulation of Unmanned Underwater Vehicle (UUV)*, including control algorithms of AUV or ROV
- *Control of UVMS*, including control algorithms that also involve a manipulator
- *Cooperation*, including the cases of two or more vehicles cooperating to achieve a task
- *Path planning*, including algorithms for new path generation
- *SLAM*, including the applications dealing with Simultaneous Localization and Mapping
- *3D Reconstruction*, including the collection of camera or sonar images of an object and its subsequent 3D reconstruction

TABLE 3. Main sensors available in the simulators.

Simulator	IMU	Pressure Sensor	Positiong System	DVL	Sonar	Camera
UWsim	✓	✓	✓	✓	✓	✓
UUV	✓	✓	✓	✓	✓	✓
Rock-Gazebo	✓	✓	✓	✓	✓	✓
Free-Floating	✓	✓	✗	✗	✗	✓
UW Morse	✓	✓	✓	✓	✓	✓
Stonefish	✓	✓	✓	✓	✓	✓
URSim	✓	✓	✗	✗	✗	✗

- *New sensor simulation*, including new models to simulate sensors in the underwater environment, in particular cameras and sonars.

The applications of UWSim, UUV simulator and other simulators were analyzed considering their fit into these categories. Despite being the main fields of application for underwater simulators, these categories do not cover all the possible applications. Thus, also other applications are considered in figures 5, 6 and 7.

A. UWsim

Figure 4 shows that UWSim has a total of 75 examples of applications, which are distributed over the range of categories identified previously, as shown in the figure 5. The applications explored the most with UWSim are UVMS and Path Planning, followed by UUV control/simulation, cooperation and SLAM. Interestingly, in the context UUV control/simulation, the Sliding Mode technique was used for depth control in [50]. Also, the Sliding Mode controller and the vehicle model were implemented to improve the simulation fidelity in [51]; a Matlab-Simulink software was integrated with UWSim, acting only as a 3D viewer, in order to track the desired trajectory. Another exciting application described in [52] is controlling the vehicle's distance from the seabed. It can be calculated directly through distance sensors, like DVL, or by a vision system consisting of a camera whose objective frames two laser pointers. Thus, an algorithm identifies the laser points and measures their distance; the vehicle's distance to the seabed is calculated. Recently, the control of underwater vehicles has been increasingly exploiting predictive techniques.

Among them, Model Predictive Control (MPC) was used to assign predefined trajectories to UVMS. For example, in [24] and [53] MPC was used to track the path while keeping the distance from the seabed constant while minimizing the computational load. In [54] a robust finite horizon controller was developed for underactuated vehicles respecting constraints like obstacle avoidance, advancement speed and saturation of the motors under the action of disturbances such as waves and currents. Techniques based on Deep Reinforcement Learning have been developed to make vehicles equipped with Sonar or Cameras perform different tasks [55]. The main advantage of these techniques is that they are not model-based and do not need a localization algorithm. A typical validation scenario is the underwater pipe inspection task. A feature of UWSim that is much appreciated is the presence of the models of some vehicles equipped with manipulators. In particular, the GIRONA 500 I-AUV vehicle can be equipped with different DOF manipulators. In this case, the control algorithms are based on the kinematic model, but the specifications vary according to the available sensors. In [56], cameras and sonar estimate vehicle torsion due to uncontrolled speed. If disturbances are not negligible, the controller must be designed to be robust to external disturbances. In [57], an Extended Kalman Filter (EKF) is implemented to estimate the position and a Fast Tube MPC to control the end effector position. Control algorithms have also been developed for redundant manipulators subject to hard joint and Cartesian constraints [58], and for allowing movements to the vehicle while checking the position and orientation of the manipulator, [59]. The controllers have been developed for carrying out different types of tasks such as the grasping of objects in the water using measurements from vision, laser, and tactile systems [60], opening and closing of a valve [61], plugging/unplugging a connector recognized using stereo cameras [30], [62], [63], solving the underwater "search & recovery" problem in shallow water conditions, with the highest level of autonomy ever seen before [63], and welding broken pipes identified by a segmentation algorithm applied to the images obtained from stereo cameras [64]. If, in addition to the kinematic model, the user also wants to include the manipulator dynamics, UWsim can be integrated with Simurv by Matlab/Simulink [30].

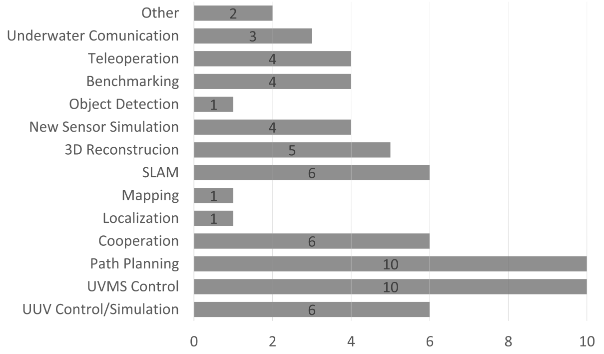


FIGURE 5. UWsim simulator citations for each application type.

A subject in which much research has been done in recent years is the cooperation between multiple vehicles. In the submarine field, this usually includes a control algorithm for the single vehicle, plus one for the cooperation of the two, with the possibility of equipping the AUVs with manipulators. A task where all these challenges are included is when two vehicles have to seize an object on the seabed and move it from one point to another. In detail, in [65] the controller of the single UVMS was designed, and in [66], [67], [68], and [69] the control and cooperation algorithm for the transport of a tube at a constant altitude and speed is developed. Shahab [70] proposes an algorithm that differs from the previous ones because it is based on the Non-linear MPC. Another multi-vehicle cooperation task is to follow a path while keeping the vehicles in formation [71], the leading vehicle is the one that is given the path to follow. In contrast, all the others must follow the leading vehicle, keeping their distance and avoiding collisions. In addition to the control issue, it is equally important to define which trajectory the vehicle must track. The part of the software responsible for generating the reference path is called Path Planner. It must generate the best possible path that allows the vehicle to perform its tasks while satisfying requirements and constraints. Authors in [72] aimed to identify the best existing path planner based on the problem. New path planners are also proposed and validated thanks to UWSim [73], [74], and [75]. In [76] an online 3D path planning algorithm was developed to respect the obstacle avoidance and distance from the seabed while trying to ensure the execution of the path without deviations and the safety of the vehicle when it has to pass near obstacles. In order to solve this problem, the AUV was equipped with a profiling sonar which also allowed the online mapping on a completely unknown environment. When the problem is to build a bathymetric map, the vehicle must run a path equipped with multibeam sonar so that the acquired images cover the entire area. An algorithm has been developed in [77] to solve this problem. When calculating the best path, it considers the desire to execute parallel paths or maximize the sonar image quality, avoiding turns in the target area. References [78] and [79] describe a framework to allow AUVs equipped with adequate sensors to explore unknown environments. The framework is composed of two main parts. The first calculates the collision-free path using sonar profiling, and, at the same time, it maps the surrounding environment incrementally. The second allows the reconstruction of various 3D representations (i.e., sparse, dense, meshed, textured) of the surveyed area using images gathered by a camera. The previous structure was validated in simulation and real environments using the Sparus-II Vehicle. It performed various tasks such as start-to-goal query in a virtual scenario of sea rocks, the breakwater blocks area inspection, and the exploration of a seamount to create a 2D occupation map [80]. In order to achieve a 3D reconstruction, it is necessary to perform a trajectory that allows gathering images from the entire object and to perform it at such a speed that there is a good percentage of overlap between two successive images. The reconstruction quality can be improved by equipping the vehicle with a localization algorithm that merges IMU, DVL and USBL/LBL and an online collision-free path planner [81]. 3D reconstruction can be obtained from Multibeam sonar [82], [83] doing one or more layers path [84]. A new 3D reconstruction technique where the object is inspected using the laser line/camera system installed in the robotic manipulator forearm is described in [85]. This technique scans the object with the laser while the robotic arm moves. The laser line captured in the image identifies the object's surface. The SLAM process enables a mobile robot to build a map of the environment and, simultaneously, to use this map to compute its location. This task is much more challenging underwater than in other environments because the visibility is reduced, and the communication and localization tasks must rely on acoustic devices. Therefore, vehicles need to integrate other onboard position sensors (IMU, depth sensor, etc.) and perception sensors (Mechanical Scanned Imaging Sonars MSIS, Forward-Looking Sonars FLS, Side-Scan Sonars SSS and cameras) in order to perform this task. Several SLAM techniques have been validated thanks to UWSim. Reference [86] presents an algorithm for Graph SLAM based on stereo vision, and EKF (Extended Kalman Filter) has been developed and tested on the SPARUS II vehicle. Silveira et al. developed a SLAM algorithm inspired by how dolphins navigate, called DolphinSLAM, using neural networks [87], [88]. The sea current was included in the training [89], and then the final DolphinSlam was compared to the EKF [90]. One more technique, the visual-pressure fusion-based SLAM technique, called ORB-SLAM, was developed in [91]. When the vehicle is equipped only with a camera and pressure sensor, the roll and pitch angle are observable only when the robot has a nonzero component of motion in the vertical direction, that is, during the robot immersion. The authors propose an online initiation method which integrates any two subsequent pressure and visual data. Sometimes sub-parts of the SLAM process may be sufficient to execute some operations. For example, localization is required in every autonomous vehicle to execute any task. An example is the creation of two EKF [92], one that merges data from IMU, DVL, visual tracker and depth gauge, and the second corrective that incorporates position from GPS or USBL, implemented and tested in UWSim. Another process that can be separated is mapping, i.e. positioning the vehicle within a predefined map or built online. In [93], a framework is proposed that incrementally maps the vehicle into the surrounding environment and simultaneously calculates a possible path that probabilistically guarantees the vehicle's safety in accordance with model/mapping uncertainties. Due to its realistic rendering UWSim is also used to validate new perceptive sensor models. For example, a new Side Scan Sonar, which approximates the Lambertian diffusion model, was developed by Gwon et al. [94]. For its validation,

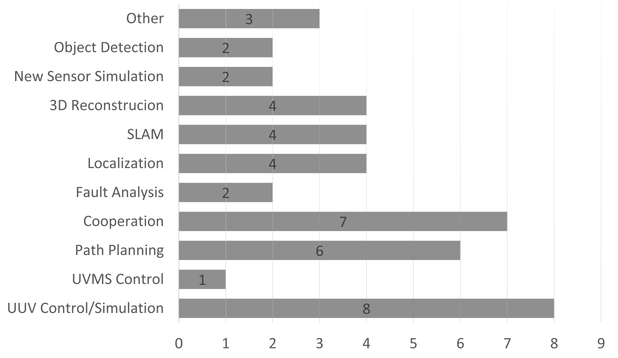


FIGURE 6. UUV simulator citations for each application type.

the inliers matching the images obtained in UWSim with the original and the new model were performed. Moreover, in underwater mining scenarios, multibeam sonar and 2D sonar imaging have been developed [95]. Other applications in which autonomous vehicles are widely used are the seabed inspection to search for a particular object or the inspection of underwater pipes for possible damage/loss. Authors in [96] developed an algorithm for real-time identification and tracing of underwater pipes using a multibeam sonar echosounder with noisy measurement and a false positive filtering system. An aspect not to be underestimated is the use of simulators for training operators in driving and managing ROVs, with or without manipulators, also called teleoperation. Usually, in the case of UVMS, two Joysticks are involved, one for the manipulator and the other for the vehicle, based, for example, on haptic force and torque [97]. Then, based on the precision required for the task, the pilot can select the control algorithm that allows him to make faster or more precise movements. Many attempts have been made to make the experience more immersive using a 3D viewer. In order to decrease the operator stress and, at the same time to increase the level of the experience of the user, a new interface was developed to drive the ROV via Leap Motion or joystick and to see the scene in a 3D viewer [98]. This interface reconstructs the 3D environment and communicates with UWSim through ROS. The SPAURS II vehicle was also piloted by an operator watching the 3D scene using an Oculus Rift viewer, and two Oculus Touch Controllers [99]. The platforms thus developed were then tested by a group of users who drove the UVMS while grasping and moving an object; however, several users still claim to prefer an original interface to the 3D one [100]. Although the teleoperation of submarine vehicles generally requires the umbilical cable, architecture was also developed to allow it to exploit wireless communication, mainly acoustic communication [101], and Radiofrequency [102]. In particular, a transmission protocol was developed and tested both in simulation and in HIL by creating special modules. In the underwater environment, the exchange of information is complicated because the only devices allowing the exchange of information at a distance of several meters are based on acoustic waves, which, however, have a low band. The need to have a large amount of information exchanged between several vehicles led to the development a new method/protocol for transporting information underwater for long missions [103]. It uses small transposing robots, called MUSSEL, which first collect data at close range from the robot stations using lights, then move to the next robot and transfer the data. A similar architecture was developed in [104], where transport robots must share the energy from surface vehicles using solar panels to deep-sea vehicles via an inductive system. Energy awareness is another well-known problem, especially for long-duration missions. A solution is proposed in [105], where a low-cost estimation model has been developed.

B. UUV SIMULATOR

UUV Simulator has a total of 45 citations. Their distribution across the various fields of application is shown in Figure 6. The main difference between UUV Simulator and UWSim lies is the lower use of UUV Simulator on UVMS. This may be explained because in the UUV Simulator, no standard vehicle is equipped with a manipulator and its implementation is not immediate. Moreover, the most frequent use of the UUV Simulator is the UUV control/simulation of submarine vehicles. The RexROV vehicle, integrated into the UUV Simulator, was used as a case study to compare the PID controller implemented within the platform with a sliding mode controller to evaluate its performance in the presence of disturbances [106]. On the opposite, authors in [107] used UUV Simulator to verify the behaviour of their newly designed mini ROV. Nowadays, neural networks have been applied to many different fields. Control systems make no exception and use simulators extensively as they let the user test the system many times and create the dataset for training and validation phases. Lopez et al. in [108] designed a Feed Forward Neural Network controller for obstacle avoidance during the pipeline inspection task. Deep Reinforcement Learning (DRL) controller was developed for different types of applications (i.e. to solve the BlueROV2 station Keeping [109] or tracking problem [110]). DRL controller was also compared with PID in the Start-end Point task in the presence of various ocean currents [111]. Regarding identifying the pose through vision systems, the tether has been a subject of study [112]. Identifying the curve formed by the cable can be used to estimate the position of an ROV during exploration. Suppose the tether connects a leader vehicle to a follower. In that case, it is advantageous to control its shape because it allows the leader ROV to explore having the minimum cable effect without the risk of entanglements. In contrast, the follower vehicle pursues it maintaining the desired shape. A control system for the follower has been proposed in [113]. The only application involving UVMS in UUV Simulator was the development of a mini-ROV for underwater manipulation [114]. The simulation environment was used to validate the first algorithms, whose aim was to identify and manipulate an underwater object using a camera. At the same time, it is connected to the main AUV, which locates it through a sonar. Robot's cooperation in UUV Simulator involved rather different tasks concerning UWsim. The exploration of a bounded area to obtain images from vision systems/sonar or water data, such as the salinity obtained by multiple robots, significantly reduces the exploration time and improves the robustness if compared to the traditional way, which uses a single robot that has a higher risk of failure. Griffith et al. have developed a new strategy for exploring unknown underwater environments based on Robotic Darwinian Particle Swarm Optimization (RDPSO) [115], in which each robot calculates online and autonomously the next target avoiding collisions and having limited communication with other members. A ROS high-level ROS modular architecture called Robot System Onboard Architecture (RSOA) for the teleoperation of surface and depth vehicles was designed in [116]. The framework comprises a Mission Management Tool (MMT) that generates missions, assigns tasks, etc., and a Middleware ROS that manages communication between MMT and vehicles via wi-fi and acoustics. A Mission Manager called Missions and Task Register MTRR has been developed in [117]. The novelty introduced is the use of decentralized hierarchical control for self-adaptive systems where the mission is not completely precharged at the beginning, but, thanks to the virtualization of the planning capabilities of the individual vehicles, the different tasks are assigned one by one once the status of those in progress is received. A method of searching for mines in a given area by multiple cooperative submarine vehicles to allow passage in a safe trajectory of a surface vehicle has also been the subject of study in [118]. Stateflow, a Matlab/Simulink package that models decision logic through state machines, was also used to develop a Model Driven Architecture for multi-vehicle coordination avoiding collisions which communicate through ROS with UUV Simulator [119]. In some applications, vehicles need the ability to communicate, but if Radio Frequency is used, the distance that these vehicles can operate is minimal. [120] shows a strategy that allows performing tasks to keep the distance between the vehicles constant. When a vehicle loses communication, one of the vehicles moves in such a way as to restore communication. Architecture Analysis and Design Language (AADL), a language for systems modelling and analysis, was also used to design a control system for vehicle coordination by analyzing faults and latencies [121]. The Osate software allowed the automatic generation of the code to integrate the control system components within ROS nodes. The Path Planner (PP) component calculates the vehicle path during the mission. It is a module that can be included in the Mission Manager, and the vehicle provided can calculate it online. The PP of off-board multi-robot for the support of actions during the execution of long-term missions was validated in the case of a surface vehicle, and several deep-sea vehicles [122]. While an online solution was presented for exploring unknown vehicle environments subject to kinematic constraints [123], onboard PP is specific to the mission, the available sensors to the presence of constraints. A new methodology has been introduced by Zacchini et al. [124] where the sensors on board guide the trajectory to explore an entire unknown area of known dimensions. In the pipeline inspection task, the path must be guided by vision systems or sonar [125]. Machine Learning based solutions have also been proposed [126], [127], and a toolchain that supports the architectural modelling of CPS with Learning enabled Components that made it possible to model the path planner through a neural network trained through many experiments in simulation. Despite an unrealistic rendering, UUV Simulator was used as a tool to test different 3D reconstruction algorithms. The RexRov vehicle equipped with MSIS was used for the exploration of a wreck, and its 3D reconstruction [128]. Moreover, to verify the structural conditions of offshore wind piles, an inspection was performed by an ROV equipped with a 2D Lidar and a localization system and 3D reconstructions of the pillars were obtained [129]. Also, Gazebo can be exploited to inspect objects that are partially submerged. Reconstruction results obtained from the Collaboration of UUV and USV inspecting the object via multibeam sonar and UAV via camera were presented in [130]. In contrast, inspection for multidomain reconstructions via ASV equipped with Multibeam Echosounder sonar for the seabed and 3D lidar for objects on the surface was carried out in [131]. Only four applications of SLAM and Localization were found. Generally, in these contexts, the vehicles are equipped with the same basic set of sensors, namely IMU, DVL, sonar or cameras. A localization system is provided in case the context requires high precision. The most common algorithm is the EKF which can have several revisions [132]. Another technique was developed in [133], which is based on the saliency of the submap of the images captured by Imaging Sonar. Creating an occupation map of an unknown area is another known problem in SLAM. It consists of creating a map of an unknown area, dividing it into a grid and indicating which elements there are obstacles. An algorithm that solves it has been proposed in [135] where a vehicle builds in a real-time fashion and updates the map during the exploration, merging measurement data from proprioceptive sensors and from a 1D laser line to replicate the profiling sonar. Localization using neural networks was also achieved in [136]. Simulators are handy for comparing different localization techniques starting from the same datasets. EKF, UKF and CDKF were compared from a dataset of an ROV subject to current disturbances [137]. Barker et al. in [138] and [139] developed a relative position estimation algorithm concerning a block of ice moving on the water surface whose one-point position is known by GPS. The vehicle, also equipped with a USBL, measures the relative distance between the vehicle and the ice employing a USBL and filters it considering the data from onboard sensors. A new side-scan sonar model was developed and implemented as a ROS node for UUV Simulator, and UWSim [140]. Another framework, developed for both simulators, greatly improved the underwater image simulation [141]. In addition to including a new model for images in different sea conditions, it allows the application of a filter that prepares the images for an easier processing phase. First attempts at underwater human-vehicle interaction were undertaken by employing UUV Simulator. Experimentation in human-vehicle interaction was carried out by simulation. An example is [142], where a diver was chased by a vehicle sensing its presence through the images taken from the stereo camera it was equipped with. Further advances in target search are described in [143] where a methodology to provide the sampling points that the robot must visit was developed. Interestingly, the methodology also considers the level of fidelity with which the robot must collect data. Another example of underwater human-robot interaction is reported in [144], where the Unity 3D interface displayed the environment on a viewer and gloves allowed the teleoperation of the ROV by interpreting hand gestures as commands and transmitting them to the ROV. Fault occurrence is an issue for every robotic system. The ability to know when a fault will occur, to establish which kind of fault it is or on which component it occurred, give time to take measures to complete the mission or to spare further damage to the robot. A neural network with a Learning enables component approach was presented in [145] for diagnosing faults. The neural network was trained by simulating various faults (e.g. on battery and motors) on the BlueROV2 vehicle, whose output goes to a module that chooses which countermeasures to adopt. Hartsell et al. instead developed a framework that calculates the risks of an autonomous vehicle during the exploration [146]. "Risk" means the probability that a hardware or software can fault or a change in environmental conditions. A low-cost platform to simulate wireless communication between surface vehicles was created starting from UUV Simulator [147]. It consists of a network module that connects all the USVs in which routing strategies have been implemented to exchange messages with each other with the shortest possible delay. Authors in [148] modified the simulator to validate Lipschitz-continuous Models with bounded confidence of which vehicles, especially submarines, are typical case studies.

C. OTHERS

All the other simulators had shallow usage. Seven uses are divided into categories as shown in figure 7.

Rock-Gazebo showed only three applications within the selected literature. The first "the simulation of the new submarine vehicle Flatfish" [149]. New high-fidelity models for real-time simulation of mechanical scanning imaging sonar and forward-looking sonar have been proposed in [150]. The same authors have improved the simulation speed in the presence of multiple sonar devices thanks to the combination of rasterization and raytracing [151]. Stonefish showed two uses. Bhat et al. [152] validated in simulation the SAM AUV vehicle defined as a Cyber-Physical System (CPS) composed of several modules to carry out multi-agent research and detection missions before carrying out field tests. Within the Twinbot project, the mission of two Girona500 vehicles with Manipulators to take a pipe and transport it by cooperating from one point to another has been replicated [153]. Stonefish has made it possible to obtain a more reliable simulation, especially of the manipulator and to obtain measurements from a force-torque sensor mounted on the end effector wrist. UW Morse was used in [154] where safety envelopes and traffic rules were validated to avoid collisions with known static objects. Lastly, the USV sim was used in two papers. The Lutra-pop vehicle was simulated with different atmospheric conditions (wind, waves, current, buoyancy) to validate the energy consumption model in different conditions [155]. Moreover, authors in [156] developed a Q-Learning approach for generating the path of surface vehicles for long-term missions. The algorithm allows the ASV to pass from a starting point, avoiding obstacles and dangerous situations while taking actions based on the wind direction. At the end of the present analysis, the figure 8 shows all considered simulators and the use cases found in the analyzed bibliography.

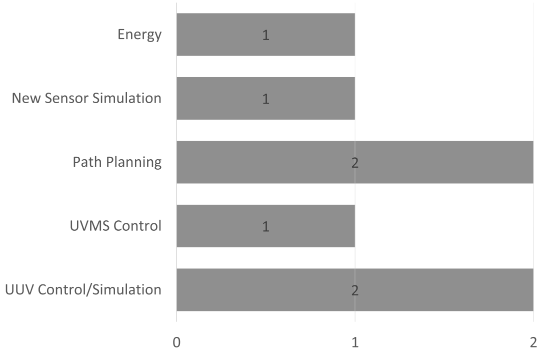
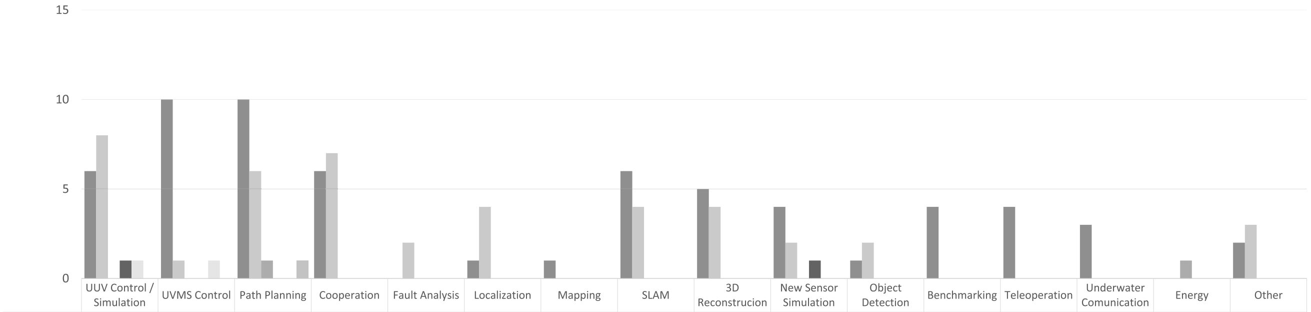


FIGURE 7. All other simulators citations for each application type.

VII. CONCLUSION

The past decade has witnessed the emergence of numerous simulators tailored for the underwater environment, facilitating the development of many algorithms and technologies. The primary benefits associated with the utilization of these tools include the reduction of costs and risks inherent in real-field missions, particularly during the developmental phase. Additionally, these simulators afford the opportunity to test algorithms in various modes such as Software In the Loop (SIL), Model In the Loop, and Hardware In the Loop (HIL) collectively referred to as xIL. DT have become both the present and the future for submarine robotics tools. The capability to establish a connection between the real device and the virtual counterpart enables more reliable simulation and validation of system components. Currently, the primary challenges revolve around the accurate integration of hydrodynamic forces into simulation without



	UUV Control / Simulation	UUVMS Control	Path Planning	Cooperation	Fault Analysis	Localization	Mapping	SLAM	3D Reconstrucion	New Sensor Simulation	Object Detection	Benchmarking	Teleoperation	Underwater Comunication	Energy	Other
■ Uwsim	6	10	10	6	0	1	1	6	5	4	1	4	4	3	0	2
■ UUVsim	8	1	6	7	2	4	0	4	4	2	2	0	0	0	0	3
■ USVsim	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
■ Rock-Gazebo	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
■ Stonefish	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
■ UW Morse	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

FIGURE 8. All simulators citations for each application type.

compromising the realism, weight and stability of the simulation. In future developments, it is anticipated that physics engines will be designed to internally integrate hydrodynamic forces. Another viable prospect involves modeling intricate phenomena through neural networks and incorporating advanced deep learning algorithms. Further advancements involve enhancing the robot's interaction with the surrounding marine life and refining the accuracy of visualizing the underwater environment. Within the realm of research, there is a focus on simulating challenges in underwater communication. This entails incorporating factors like signal degradation and addressing issues arising from the thermocline, particularly in the context of acoustic communication, and also encompassing optical communication challenges. Exploring methodologies for simulating and evaluating the fault tolerance and robustness of underwater robotic systems necessitates the generation of scenarios that challenge the robots with unforeseen adversities and malfunctions, thereby assessing their capacity to respond effectively.

This study presented a comprehensive review of academic simulators for submarine environments and agents. Initially, the general structure of all the academic simulators currently featured in the State-Of-The-Art (SOTA) is presented. Simulators developed within the last decade that are not yet dismissed are discussed, with UWsim and UUVsim emerging as the most frequently used and cited. Each simulator exhibits distinct strengths and weaknesses; UUVsim demonstrates high simulation fidelity but lacks realistic rendering, in contrast to UWsim. At the same time, Stonefish incorporates the best of these two simulators but imposes a substantial computational load on the processor.

Thinking about the future of these simulators, UWsim and UUV simulator are anticipated to persist due to their multifaceted features reliance on ROS, a middleware pivotal in building the software architecture of many ROVs/AUVs. Furthermore, authors of UWsim have introduced UWsim-NET [157] a novel tool facilitating accurate simulation of acoustic waves transmission in water. This innovation allows

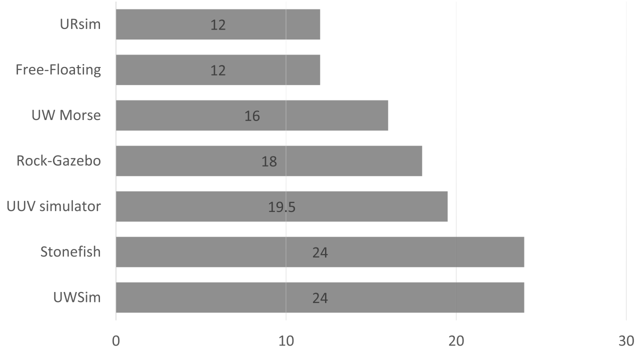


FIGURE 9. Underwater simulator final score.

the implementation and simulation of acoustic positioning and communication systems, a unique feature compared to all the other simulators. Conversely, Free Floating, UW morse and URsim, despite being available online, arere presently underutilized and no longer undergoing development, making their obsolescence in the near future. Stonefish on the other hand holds significant potential, especially in simulating multibody vehicles in the water, yet its future remains uncertain due to inadequate documentation and limited awareness in the community.

From the detailed analysis of the applications of underwater simulators presented in this study, it appears evident that researchers have developed several algorithms with very different purposes. the wide-ranging applications of these simulators include the development of control algorithms for submarine vehicles, manipulators and cooperating robots. Additionally, they have been instrumental in real-time and non-real-time path planning for the data collection and SLAM. The paper showed insights into creating DTs of robotic submarine systems using available simulators, outlining essential characteristics to guide new researchers in their development endeavors. Figure 9 displays a graph illustrating the assigned scores to each simulator based on the criteria outlined in Tables 2 and 3. The scoring system involved assigning scores of 0, 1, and 2 to each simulator, following the guidelines provided in Table 2. For the manipulator category, scores were determined based on the presence of both kinematics and dynamics modules (score of 2), the implementation of both modules with incomplete dynamics (score of 1.5, exemplified by UUV simulator and Stonefish), or the implementation of only kinematics equations (score of 1, exemplified by UWsim). Results from Table 2 were considered, assigning a score of 1 if a sensor was present and 0 otherwise. The cumulative scores were utilized to rank each simulator, as depicted in Figure 9, showcasing the outcomes of this scoring approach. According to this methodology, UWsim, Stonefish, and UUV emerged as the top three simulators for digital twinning, occupying the highest ranks. Nonetheless, Stonefish exhibits fewer documented applications compared to UWsim and UUV, which appear to offer robust resources, extensive applications, and well-documented functionalities. In the future these aspects could impede the adoption of Stonefish while potentially facilitating the widespread use of UWsim and UUV.

REFERENCES

[1] C. Semeraro, M. Lezoche, H. Panetto, and M. Dassisti, "Digital twin paradigm: A systematic literature review," *Comput. Ind.*, vol. 130, Sep. 2021, Art. no. 103469, doi: 10.1016/j.compind.2021.103469.

[2] J. Collins, S. Chand, A. Vanderkop, and D. Howard, "A review of physics simulators for robotic applications," *IEEE Access*, vol. 9, pp. 51416–51431, 2021.

[3] E. Papadopoulos, F. Aghili, O. Ma, and R. Lampariello, "Robotic manipulation and capture in space: A survey," *Frontiers Robot. AI*, vol. 8, Jul. 2021, Art. no. 686723.

[4] B. Bingham, C. Aguero, M. McCarrin, J. Klamo, J. Malia, K. Allen, T. Lum, M. Rawson, and R. Waqar, "Toward maritime robotic simulation in Gazebo," in *Proc. OCEANS*, 2019, pp. 1–10, doi: 10.23919/OCEANS40490.2019.8962724.

[5] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2004, pp. 2149–2154.

[6] M. A. Hopkins, D. W. Hong, and A. Leonessa, "Humanoid locomotion on uneven terrain using the time-varying divergent component of motion," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Nov. 2014, pp. 266–272.

[7] S. Hrabar, "3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 807–814.

[8] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, "Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2261–2268, Jul. 2018.

[9] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 1321–1326.

[10] G. Echeverria, N. Lassabe, A. Degroote, and S. Lemaignan, "Modular open robots simulation engine: MORSE," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 46–51.

[11] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 31–36.

[12] J. L. Sanchez-Lopez, M. Wang, M. A. Olivares-Mendez, M. Molina, and H. Voos, "A real-time 3D path planning solution for collision-free navigation of multirotor aerial robots in dynamic environments," *J. Intell. Robotic Syst.*, vol. 93, nos. 1–2, pp. 33–53, Feb. 2019.

[13] G. J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, and J. Peters, "Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks," *Auto. Robots*, vol. 41, no. 3, pp. 593–612, Mar. 2017, doi: 10.1007/s10514-016-9556-2.

[14] G. Echeverria, S. Lemaignan, A. Degroote, S. Lacroix, M. Karg, P. Koch, C. Lesire, and S. Stinckwich, "Simulating complex robotic scenarios with MORSE," in *Simulation, Modeling, and Programming for Autonomous Robots*, I. Noda, N. Ando, D. Brugali, and J. J. Kuffner, Eds. Berlin, Germany: Springer, 2012, pp. 197–208.

[15] C. M. Filho, D. F. Wolf, V. Grassi, and F. S. Osório, "Longitudinal and lateral control for autonomous ground vehicles," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 588–593.

[16] D. Gelernter, *Mirror Worlds: Or the Day Software Puts the Universe in a Shoebox…How it Will Happen and What it Will Mean*. Oxford, U.K.: Oxford Univ. Press, 1993. [Online]. Available: https://books.google.it/books?id=jh2U379fq18C

[17] M. W. Grieves, S. Flumerfelt, K. G. Schwartz, D. Mavris, and S. Briceno, *Complex Systems Engineering: Theory and Practice*. American Institute of Aeronautics and Astronautics, 2019, pp. 175–200.

[18] M. Shafto, M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne, and L. Wang. (2012). *Simulation, Information Technology and Processing Roadmap. National Aeronautics and Space Administration*. National Aeronautics and Space Administration. [Online]. Available: https://www.nasa.gov/sites/default/files/501321main_TA11-ID_rev4_NRC-wTASR.pdf

[19] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and US air force vehicles," in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dyn. Materials Conf.*, Apr. 2012, p. 1818.

[20] R. Rosen, G. von Wichert, G. Lo, and K. D. Bettenhausen, "About the importance of autonomy and digital twins for the future of manufacturing," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 567–572, 2015, doi: 10.1016/j.ifacol.2015.06.141.

[21] M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," *White Paper*, vol. 1, no. 2014, pp. 1–7, 2014.

[22] M. Schluse and J. Rossmann, "From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems," in *Proc. IEEE Int. Symp. Syst. Eng. (ISSE)*, Oct. 2016, pp. 1–6, doi: 10.1109/SysEng.2016.7753162.

[23] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with big data," *Int. J. Adv. Manuf. Technol.*, vol. 94, nos. 9–12, pp. 3563–3576, Feb. 2018.

[24] B. Wu, W. D. Widanage, S. Yang, and X. Liu, "Battery digital twins: Perspectives on the fusion of models, data and artificial intelligence for smart battery management systems," *Energy AI*, vol. 1, Aug. 2020, Art. no. 100016, doi: 10.1016/j.egyai.2020.100016.

[25] M. Prats, J. Perez, J. J. Fernandez, and P. J. Sanz, "An open source tool for simulation and supervision of underwater intervention missions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 2577–2582.

[26] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach, "UUV simulator: A Gazebo-based package for underwater intervention and multi-robot simulation," in *Proc. OCEANS MTS/IEEE Monterey*, Sep. 2016, pp. 1–8, doi: 10.1109/OCEANS.2016.7761080.

[27] O. Kermorgant, "A dynamic simulator for underwater vehicle-manipulators," in *Simulation, Modeling, and Programming for Autonomous Robots*, vol. 8810. Berlin, Germany: Springer, Oct. 2014, pp. 25–36.

[28] T. Watanabe, G. Neves, R. Cerqueira, T. Trocoli, M. Reis, S. Joyeux, and J. Albiez, "The Rock-Gazebo integration and a real-time AUV simulation," in *Proc. 12th Latin Amer. Robot. Symp., 3rd Brazilian Symp. Robot. (LARS-SBR)*, 2015, pp. 132–138.

[29] E. H. Henriksen, I. Schjølberg, and T. B. Gjersvik, "UW MORSE: The underwater modular open robot simulation engine," in *Proc. IEEE/OES Auto. Underwater Vehicles (AUV)*, Nov. 2016, pp. 261–267.

[30] J. J. Fernandez, J. Perez, A. Peñalver, J. Sales, D. Fornas, and P. J. Sanz, "Benchmarking using UWSim, Simurv and ROS: An autonomous free floating dredging intervention case study," in *Proc. OCEANS*, 2015, pp. 1–7, doi: 10.1109/OCEANS-Genova.2015.7271514.

[31] P. J. Sanz, J. Pérez, J. Sales, A. Peñalver, J. J. Fernández, D. Fornas, R. Marín, and J. C. Garcia, "A benchmarking perspective of underwater intervention systems," *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 8–13, 2015.

[32] J. Pérez, J. Sales, A. Peñalver, J. J. Fernández, D. Fornas, J. C. García, R. Mar ̄ın, and P. J. Sanz, "Benchmarking water turbidity effect on tracking algorithms," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11191–11196, Jul. 2017, doi: 10.1016/j.ifacol.2017.08.1243.

[33] J. Pérez, J. Sales, R. Marín, and P. J. Sanz, ''Online tool for benchmarking of simulated intervention autonomous underwater vehicles: Evaluating position controllers in changing underwater currents,'' in *Proc. 2nd Int. Conf. Artif. Intell., Model. Simul.*, 2014, pp. 246–251.

[34] J. Perez, J. Sales, A. Penalver, D. Fornas, J. J. Fernandez, J. C. Garcia, P. J. Sanz, R. Marin, and M. Prats, ''Exploring 3-D reconstruction techniques: A benchmarking tool for underwater robotics,'' *IEEE Robot. Autom. Mag.*, vol. 22, no. 3, pp. 85–95, Sep. 2015.

[35] J. Pérez, J. Sales, R. Marín, and P. J. Sanz, ''Web-based configuration tool for benchmarking of simulated intervention autonomous underwater vehicles,'' in *Proc. IEEE Int. Conf. Auto. Robot Syst. Competitions (ICARSC)*, May 2014, pp. 279–284.

[36] P. Cieslak, ''Stonefish: An advanced open-source simulation tool designed for marine robotics, with a ROS interface,'' in *Proc. OCEANS*, Jun. 2019, pp. 1–6, doi: 10.1109/OCEANSE.2019.8867434.

[37] R. Featherstone, *Rigid Body Dynamics Algorithms*. Berlin, Germany: Springer, 2008.

[38] P. Katara, M. Khanna, H. Nagar, and A. Panaiyappan, ''Open source simulator for unmanned underwater vehicles using ROS and Unity3D,'' in *Proc. IEEE Underwater Technol. (UT)*, Apr. 2019, pp. 1–7, doi: 10.1109/UT.2019.8734309.

[39] M. Paravisi, D. H. Santos, V. Jorge, G. Heck, L. Gonçalves, and A. Amory, ''Unmanned surface vehicle simulator with realistic environmental disturbances,'' *Sensors*, vol. 19, no. 5, p. 1068, Mar. 2019, doi: 10.3390/s19051068.

[40] S. Garg, J. Quintas, J. Cruz, and A. M. Pascoal, ''NetMarSyS—A tool for the simulation and visualization of distributed autonomous marine robotic systems,'' in *Proc. IEEE/OES Auto. Underwater Vehicles Symp. (AUV)*, Oct. 2020, pp. 1–5, doi: 10.1109/AUV50043.2020.9267922.

[41] K. Suzuki and K. Kawabata, ''Development of a simulator for underwater reconnaissance tasks by utilizing remotely operated robots,'' in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Jan. 2020, pp. 1100–1106.

[42] W. Wang, H. Zhang, Y. Li, Z. Zhang, X. Luo, and S. Xie, ''USVs-sim: A general simulation platform for unmanned surface vessels autonomous learning,'' *Concurrency Comput., Pract. Exper.*, vol. 34, no. 3, Feb. 2022, Art. no. e6567, doi: 10.1002/cpe.6567.

[43] C. Morency, D. J. Stilwell, and S. Hess, ''Development of a simulation environment for evaluation of a forward looking sonar system for small auvs,'' in *Proc. OCEANS*, 2019, pp. 1–9, doi: 10.23919/OCEANS40490.2019.8962650.

[44] O. Ganoni, R. Mukundan, and R. Green, ''A generalized simulation framework for tethered remotely operated vehicles in realistic underwater environments,'' *Drones*, vol. 3, no. 1, p. 34, Dec. 2018, doi: 10.3390/drones3010001.

[45] M. Razzanelli, S. Casini, M. Innocenti, and L. Pollini, ''Development of a hybrid simulator for underwater vehicles with manipulators,'' *IEEE J. Ocean. Eng.*, vol. 45, no. 4, pp. 1235–1251, Oct. 2020.

[46] P. Smith and M. Dunbabin, ''High-fidelity autonomous surface vehicle simulator for the maritime RobotX challenge,'' *IEEE J. Ocean. Eng.*, vol. 44, no. 2, pp. 310–319, Apr. 2019.

[47] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Hoboken, NJ, USA: Wiley, 2011.

[48] J. Britto, A. Conceiçáo, S. Joyeux, and J. Albiez, ''Improvements in dynamics simulation for underwater vehicles deployed in Gazebo,'' in *Proc. OCEANS*, 2017, pp. 1–6.

[49] D. R. Yoerger, J. G. Cooke, and J.-J. E. Slotine, ''The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design,'' *IEEE J. Ocean. Eng.*, vol. 15, no. 3, pp. 167–178, Jul. 1990.

[50] M. Cai, Y. Wang, S. Wang, R. Wang, and M. Tan, ''ROS-based depth control for hybrid-driven underwater vehicle-manipulator system,'' in *Proc. Chin. Control Conf. (CCC)*, Jul. 2019, pp. 4576–4580.

[51] Q. Tang, D. Chen, Y. Li, and L. Tang, ''Three-dimensional visual simulation for trajectory tracking of autonomous underwater vehicle based on UWSim,'' in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2019, pp. 1175–1180.

[52] P. M. Rodrigues, N. A. Cruz, and A. M. Pinto, ''Altitude control of an underwater vehicle based on computer vision,'' in *Proc. OCEANS*, 2018, pp. 1–9, doi: 10.1109/OCEANS.2018.8604525.

[53] M. Cai, Y. Wang, S. Wang, R. Wang, L. Cheng, and M. Tan, ''Prediction-based seabed terrain following control for an underwater vehicle-manipulator system,'' *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 8, pp. 4751–4760, Aug. 2021.

[54] S. Heshmati-Alamdari, A. Nikou, and D. V. Dimarogonas, ''Robust trajectory tracking control for underactuated autonomous underwater vehicles in uncertain environments,'' *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 3, pp. 1288–1301, Jul. 2021.

[55] H. Wu, S. Song, Y. Hsu, K. You, and C. Wu, ''End-to-end sensorimotor control problems of AUVs with deep reinforcement learning,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 5869–5874.

[56] O. Kermorgant, Y. Pétillot, and M. Dunnigan, ''A global control scheme for free-floating vehicle-manipulators,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 5015–5020.

[57] Y. Dai, S. Yu, Y. Yan, and X. Yu, ''An EKF-based fast tube MPC scheme for moving target tracking of a redundant underwater vehicle-manipulator system,'' *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 6, pp. 2803–2814, Dec. 2019.

[58] A. Peñalver, J. J. Fernández, A. Soriano, and P. J. Sanz, ''A multi-task priority framework for redundant robots with multiple kinematic chains under hard joint and Cartesian constraints,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 2410–2417.

[59] A. Carrera, M. Carreras, P. Kormushev, N. Palomeras, and S. Nagappa, ''Towards valve turning with an AUV using learning by demonstration,'' in *Proc. OCEANS*, Jun. 2013, pp. 1–7, doi: 10.1109/OCEANS-Bergen.2013.6608097.

[60] P. Sanz, A. Penalver, J. Sales, D. Fornas, J. Fernandez, J. Perez, and J. Bernabe, ''GRASPER: A multisensory based manipulation system for underwater operations,'' in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2013, pp. 4036–4041.

[61] A. Peñalver, J. Pérez, J. J. Fernández, J. Sales, P. J. Sanz, J. C. García, D. Fornas, and R. Marín, ''Autonomous intervention on an underwater panel mockup by using visually-guided manipulation techniques,'' *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 5151–5156, 2014, doi: 10.3182/20140824-6-za-1003.02545.

[62] P. J. Sanz, M. Vincze, and D. Fornas, ''Increasing underwater manipulation autonomy using segmentation and visual tracking,'' in *Proc. OCEANS*, 2017, pp. 1–5, doi: 10.1109/OCEANSE.2017.8084762.

[63] P. J. Sanz, P. Ridao, G. Oliver, G. Casalino, Y. Petillot, C. Silvestre, C. Melchiorri, and A. Turetta, ''TRIDENT—An European project targeted to increase the autonomy levels for underwater intervention missions,'' in *Proc. OCEANS*, 2013, pp. 1–10, doi: 10.1109/IEEECONF38699.2020.9389466.

[64] V. Garg, T. Sharma, T. Singh, and V. Rastogi, ''Crack detection and localization with real-time path tracking using stereo vision for autonomous underwater welding,'' in *Proc. Global Oceans*, Oct. 2020, pp. 1–8, doi: 10.1109/IEEECONF38699.2020.9389466.

[65] E. Simetti and G. Casalino, ''Manipulation and transportation with cooperative underwater vehicle manipulator systems,'' *IEEE J. Ocean. Eng.*, vol. 42, no. 4, pp. 782–799, Oct. 2017.

[66] G. Casalino, E. Simetti, N. Manerikar, A. Sperinde, S. Torelli, and F. Wanderlingh, ''Cooperative underwater manipulation systems: Control developments within the MARIS project,'' *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 1–7, 2015, doi: 10.1016/j.ifacol.2015.06.001.

[67] E. Simetti, G. Casalino, N. Manerikar, A. Sperindé, S. Torelli, and F. Wanderlingh, ''Cooperation between autonomous underwater vehicle manipulations systems with minimal information exchange,'' in *Proc. OCEANS*, May 2015, pp. 1–6, doi: 10.1109/OCEANS-Genova.2015.7271700.

[68] N. Manerikar, G. Casalino, E. Simetti, S. Torelli, and A. Sperindé, ''On autonomous cooperative underwater floating manipulation systems,'' in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 523–528.

[69] N. Manerikar, G. Casalino, E. Simetti, S. Torelli, and A. Sperindé, ''On cooperation between autonomous underwater floating manipulation systems,'' in *Proc. IEEE Underwater Technol. (UT)*, Feb. 2015, pp. 1–6, doi: 10.1109/UT.2015.7108310.

[70] S. Heshmati-Alamdari, G. C. Karras, and K. J. Kyriakopoulos, ''A distributed predictive control approach for cooperative manipulation of multiple underwater vehicle manipulator systems,'' 2019, *arXiv:1906.09563*.

[71] C. P. Bechlioulis, F. Giagkas, G. C. Karras, and K. J. Kyriakopoulos, ''Robust formation control for multiple underwater vehicles,'' *Frontiers Robot. AI*, vol. 6, p. 90, Sep. 2019, doi: 10.3389/frobt.2019.00090.

[72] D. Youakim and P. Ridao, ''Motion planning survey for autonomous mobile manipulators underwater manipulator case study,'' *Robot. Auto. Syst.*, vol. 107, pp. 20–44, Sep. 2018, doi: 10.1016/j.robot.2018.05.006.

[73] J. D. Hernández, G. Vallicrosa, E. Vidal, È. Pairet, M. Carreras, and P. Ridao, "On-line 3D path planning for close-proximity surveying with AUVs," *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 50–55, 2015, doi: 10.1016/j.ifacol.2015.06.009.

[74] M. Carreras, J. D. Hernández, E. Vidal, N. Palomeras, and P. Ridao, "Online motion planning for underwater inspection," in *Proc. IEEE/OES Auto. Underwater Vehicles (AUV)*, Nov. 2016, pp. 336–341.

[75] J. D. Hernáandez, M. Moll, E. Vidal, M. Carreras, and L. E. Kavraki, "Planning feasible and safe paths online for autonomous underwater vehicles in unknown environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 1313–1320.

[76] J. D. Hernández, E. Vidal, G. Vallicrosa, E. Galceran, and M. Carreras, "Online path planning for autonomous underwater vehicles in unknown environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2015, pp. 1152–1157.

[77] E. Galceran, S. Nagappa, M. Carreras, P. Ridao, and A. Palomer, "Uncertainty-driven survey path planning for bathymetric mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 6006–6012.

[78] J. David Hernández, E. Vidal, G. Vallicrosa, É. Pairet, and M. Carreras, "Simultaneous mapping and planning for autonomous underwater vehicles in unknown environments," in *Proc. OCEANS*, 2015, pp. 1–6, doi: 10.1109/OCEANS-Genova.2015.7271469.

[79] J. Hernández, K. Istenic, N. Gracias, N. Palomeras, R. Campos, E. Vidal, R. García, and M. Carreras, "Autonomous underwater navigation and optical mapping in unknown natural environments," *Sensors*, vol. 16, no. 8, p. 1174, Jul. 2016, doi: 10.3390/s16081174.

[80] M. Carreras, J. D. Hernández, E. Vidal, N. Palomeras, D. Ribas, and P. Ridao, "Sparus II AUV—A hovering vehicle for seabed inspection," *IEEE J. Ocean. Eng.*, vol. 43, no. 2, pp. 344–355, Apr. 2018.

[81] Z. Shen, J. P. Wilson, and S. Gupta, "An online coverage path planning algorithm for curvature-constrained AUVs," in *Proc. MTS/IEEE OCEANS*, Oct. 2019, pp. 1–5, doi: 10.23919/OCEANS40490.2019.8962629.

[82] Z. Shen, J. Song, K. Mittal, and S. Gupta, "An autonomous integrated system for 3-D underwater terrain map reconstruction," in *Proc. OCEANS*, Oct. 2016, pp. 1–6, doi: 10.1109/OCEANS.2016.7761067.

[83] T. Guerneve and Y. Petillot, "Underwater 3D reconstruction using BlueView imaging sonar," in *Proc. OCEANS*, May 2015, pp. 1–7, doi: 10.1109/OCEANS-Genova.2015.7271575.

[84] Z. Shen, J. Song, K. Mittal, and S. Gupta, "Autonomous 3-D mapping and safe-path planning for underwater terrain reconstruction using multi-level coverage trees," in *Proc. OCEANS*, Anchorage, AK, USA, Sep. 2017, pp. 1–6.

[85] A. Penalver, J. J. Fernandez, J. Sales, and P. J. Sanz, "Multi-view underwater 3D reconstruction using a stripe laser light and an eye-in-hand camera," in *Proc. OCEANS*, 2015, pp. 1–6, doi: 10.1109/OCEANS-Genova.2015.7271497.

[86] P. L. N. Carrasco, F. Bonin-Font, M. M. Campos, and G. O. Codina, "Stereo-vision graph-SLAM for robust navigation of the AUV SPARUS II," *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 200–205, 2015, doi: 10.1016/j.ifacol.2015.06.033.

[87] L. Silveira, F. Guth, P. Drews-Jr., P. Ballester, M. Machado, F. Codevilla, N. Duarte-Filho, and S. Botelho, "An open-source bio-inspired solution to underwater SLAM," *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 212–217, 2015, doi: 10.1016/j.ifacol.2015.06.035.

[88] G. B. Zaffari, M. M. dos Santos, A. C. Duarte, D. De A. Fernandes, and S. S. D. C. Botelho, "Exploring the DolphinSLAM's parameters," in *Proc. OCEANS*, Apr. 2016, pp. 1–5, doi: 10.1109/OCEANSAP.2016.7485531.

[89] G. B. Zaffari, M. M. Dos Santos, P. L. J. Drews, and S. S. C. Botelho, "Effects of water currents in a continuous attractor neural network for SLAM applications," in *Proc. 13th Latin Amer. Robot. Symp., 5th Brazilian Robot. Symp. (LARS/SBR)*, Oct. 2016, pp. 328–333.

[90] A. C. Duarte, G. B. Zaffari, R. T. S. da Rosa, L. M. Longaray, P. Drews, and S. S. C. Botelho, "Towards comparison of underwater SLAM methods: An open dataset collection," in *Proc. OCEANS MTS/IEEE Monterey*, Sep. 2016, pp. 1–5, doi: 10.1109/OCEANS.2016.7761315.

[91] C. Hu, S. Zhu, Y. Liang, Z. Mu, and W. Song, "Visual-pressure fusion for underwater robot localization with online initialization," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 8426–8433, Oct. 2021.

[92] E. Guerrero-Font, M. Massot-Campos, P. L. Negre, F. Bonin-Font, and G. O. Codina, "An USBL-aided multisensor navigation system for field AUVs," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. for Intell. Syst. (MFI)*, Sep. 2016, pp. 430–435.

[93] É. Pairet, J. D. Hernández, M. Lahijanian, and M. Carreras, "Uncertainty-based online mapping and motion planning for marine robotics guidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 2367–2374.

[94] D.-H. Gwon, J. Kim, M. H. Kim, H. G. Park, T. Y. Kim, and A. Kim, "Development of a side scan sonar module for the underwater simulator," in *Proc. 14th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, 2017, pp. 662–665.

[95] D. Sytnyk, R. Pereira, D. Pedrosa, J. Rodrigues, A. Martins, A. Dias, J. Almeida, and E. Silva, "Simulation environment for underground flooded mines robotic exploration," in *Proc. OCEANS*, 2017, pp. 1–6, doi: 10.1109/OCEANSE.2017.8084907.

[96] V. Bharti, D. Lane, and S. Wang, "Robust subsea pipeline tracking with noisy multibeam echosounder," in *Proc. IEEE/OES Auto. Underwater Vehicle Workshop (AUV)*, Nov. 2018, pp. 1–5, doi: 10.1109/AUV.2018.8729803.

[97] J. Ma, Y. Wang, R. Wang, and S. Wang, "Remote operation with haptic force and virtual proxy for an underwater vehicle-manipulator system," in *Proc. IEEE 10th Data Driven Control Learn. Syst. Conf. (DDCLS)*, Oct. 2021, pp. 304–309.

[98] J. C. Garcia, B. Patrao, J. Perez, J. Seabra, P. Menezes, J. Dias, and P. J. Sanz, "Towards an immersive and natural gesture controlled interface for intervention underwater robots," in *Proc. OCEANS*, 2015, pp. 1–5, doi: 10.1109/OCEANS-Genova.2015.7271501.

[99] F. Bonin-Font, M. Massot Campos, and A. B. Burguera, "ARSEA: A virtual reality subsea exploration assistant," *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 26–31, 2018.

[100] M. de la Cruz, G. Casañ, P. Sanz, and R. Marín, "Preliminary work on a virtual reality interface for the guidance of underwater robots," *Robotics*, vol. 9, no. 4, p. 81, Oct. 2020, doi: 10.3390/robotics9040081.

[101] D. Centelles, E. Moscoso, G. Vallicrosa, N. Palomeras, J. Sales, J. V. Marti, R. Marin, P. Ridao, and P. J. Sanz, "Wireless HROV control with compressed visual feedback over an acoustic link," in *Proc. OCEANS*, 2017, pp. 1–7, doi: 10.1109/OCEANSE.2017.8084979.

[102] D. Centelles, A. Soriano, R. Marin, and P. J. Sanz, "Wireless HROV control with compressed visual feedback using acoustic and RF links," *J. Intell. Robotic Syst.*, vol. 99, nos. 3–4, pp. 713–728, Sep. 2020.

[103] P. Mazdin, B. Arbanas, T. Haus, S. Bogdan, T. Petrovic, and N. Miskovic, "Trust consensus protocol for heterogeneous underwater robotic systems," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 341–346, 2016, doi: 10.1016/j.ifacol.2016.10.427.

[104] B. Arbanas, S. Boljuncic, T. Petrovic, and S. Bogdan, "Decentralized energy sharing protocol using TÆMS framework and coalition-based metaheuristic for heterogeneous robotic systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5914–5919, Jul. 2017, doi: 10.1016/j.ifacol.2017.08.1377.

[105] V. De Carolis, D. M. Lane, and K. E. Brown, "Low-cost energy measurement and estimation for autonomous underwater vehicles," in *Proc. OCEANS*, 2014, pp. 1–5, doi: 10.1109/OCEANS-TAIPEI.2014.6964490.

[106] M. M. M. Manhaes, S. A. Scherer, L. R. Douat, M. Voss, and T. Rauschenbach, "Use of simulation-based performance metrics on the evaluation of dynamic positioning controllers," in *Proc. OCEANS*, Jun. 2017, pp. 1–8.

[107] S. Ray, R. Bhowal, and P. Patel, "An overview of the design and development of a 6 DOF remotely operated vehicle for underwater structural inspection," in *Proc. Int. Conf. Commun., Control Inf. Sci. (ICCISc)*, vol. 1, Jun. 2021, pp. 1–6, doi: 10.1109/ICCISc52257.2021.9484879.

[108] D. M. Lopez, P. Musau, N. Hamilton, H.-D. Tran, and T. T. Jonhson, "Case study: Safety verification of an unmanned underwater vehicle," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2020, pp. 189–195.

[109] K. B. Knudsen, M. C. Nielsen, and I. Schjølberg, "Deep learning for station keeping of AUVs," in *Proc. OCEANS*, Oct. 2019, pp. 1–6, doi: 10.23919/OCEANS40490.2019.8962598.

[110] Q. Zhang, J. Lin, Q. Sha, B. He, and G. Li, "Deep interactive reinforcement learning for path following of autonomous underwater vehicle," *IEEE Access*, vol. 8, pp. 24258–24268, 2020.

[111] Y. Sola, T. Chaffre, G. Le Chenadec, K. Sammut, and B. Clement, "Evaluation of a deep-reinforcement-learning-based controller for the control of an autonomous underwater vehicle," in *Proc. Global Oceans*, Oct. 2020, pp. 1–7, doi: 10.1109/IEEECONF38699.2020.9389415.

[112] M. Laranjeira, C. Dune, and V. Hugel, "Embedded visual detection and shape identification of underwater umbilical for vehicle positioning," in *Proc. OCEANS*, Jun. 2019, pp. 1–9, doi: 10.1109/OCEANSE.2019.8867548.

[113] M. Laranjeira, C. Dune, and V. Hugel, "Catenary-based visual servoing for tether shape control between underwater vehicles," *Ocean Eng.*, vol. 200, Mar. 2020, Art. no. 107018, doi: 10.1016/j.oceaneng.2020.107018.

[114] J. Kim, T. Kim, J. Kim, S. Rho, Y.-W. Song, and S.-C. Yu, "Simulation and feasibility test of mini-ROVs with AUV for the manipulation purpose," in *Proc. OCEANS*, 2019, pp. 1–6, doi: 10.23919/OCEANS40490.2019.8962810.

[115] N. D. G. Sànchez, P. A. Vargas, and M. S. Couceiro, "A Darwinian swarm robotics strategy applied to underwater exploration," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–6, doi: 10.1109/CEC.2018.8477738.

[116] M. Barbier, E. Bensana, and X. Pucel, "A generic and modular architecture for maritime autonomous vehicles," in *Proc. IEEE/OES Auto. Underwater Vehicle Workshop (AUV)*, Nov. 2018, pp. 1–6, doi: 10.1109/AUV.2018.8729765.

[117] N. L. Martínez, J.-F. Martínez-Ortega, J. Rodríguez-Molina, and Z. Zhai, "Proposal of an automated mission manager for cooperative autonomous underwater vehicles," *Appl. Sci.*, vol. 10, no. 3, p. 855, Jan. 2020, doi: 10.3390/app10030855.

[118] J. Fransman, J. Sijs, H. Dol, E. Theunissen, and B. De Schutter, "Distributed constraint optimization for autonomous multi AUV mine counter-measures," in *Proc. OCEANS*, 2018, pp. 1–7, doi: 10.1109/OCEANS.2018.8604924.

[119] T. Soriano, H. A. Pham, and V. H. Ngo, "Analysis of coordination modes for multi-UUV based on model driven architecture," in *Proc. 12th France-Japan 10th Europe-Asia Congr. Mechatronics*, Sep. 2018, pp. 189–194.

[120] N. Li, M. Li, D. Huang, and W. Yi, "Multi-UUV task allocation with limited communication range based on hoplites framework," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 428, no. 1, Sep. 2018, Art. no. 012047, doi: 10.1088/1757-899X/428/1/012047.

[121] H. A. Pham, T. Soriano, and V. H. Ngo, "Applying AADL to realize embedded control systems for coordination of multiple low-cost underwater drones," in *Proc. OCEANS*, Jun. 2019, pp. 1–7, doi: 10.1109/OCEANSE.2019.8867198.

[122] Y. Carreno, R. P. A. Petrick, and Y. Petillot, "Multi-agent strategy for marine applications via temporal planning," in *Proc. IEEE 2nd Int. Conf. Artif. Intell. Knowl. Eng. (AIKE)*, Jun. 2019, pp. 243–250.

[123] J. Scharff Willners, D. Gonzalez-Adell, J. D. Hernández, È. Pairet, and Y. Petillot, "Online 3-dimensional path planning with kinematic constraints in unknown environments using hybrid A* with tree pruning," *Sensors*, vol. 21, no. 4, p. 1152, Feb. 2021.

[124] L. Zacchini, A. Ridolfi, and B. Allotta, "Receding-horizon sampling-based sensor-driven coverage planning strategy for AUV seabed inspections," in *Proc. IEEE/OES Auto. Underwater Vehicles Symp. (AUV)*, Sep. 2020, pp. 1–6, doi: 10.1109/AUV50043.2020.9267903.

[125] W. Akram and A. Casavola, "A visual control scheme for AUV underwater pipeline tracking," in *Proc. IEEE Int. Conf. Auto. Syst. (ICAS)*, Aug. 2021, pp. 1–5, doi: 10.1109/ICAS49788.2021.9551173.

[126] C. Hartsell, N. Mahadevan, S. Ramakrishna, A. Dubey, T. Bapty, T. Johnson, X. Koutsoukos, J. Sztipanovits, and G. Karsai, "CPS design with learning-enabled components: A case study," in *Proc. 30th Int. Workshop Rapid Syst. Prototyping*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 57–63, doi: 10.1145/3339985.3358491.

[127] C. Hartsell, N. Mahadevan, S. Ramakrishna, A. Dubey, T. Bapty, T. Johnson, X. Koutsoukos, J. Sztipanovits, and G. Karsai, "Model-based design for CPS with learning-enabled components," in *Proc. Workshop Design Autom. CPS IoT*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–9, doi: 10.1145/3313151.3313166.

[128] B. Justo, M. M. dos Santos, P. L. J. Drews, J. Arigony, and A. W. Vieira, "3D surfaces reconstruction and volume changes in underwater environments using MSIS sonar," in *Proc. Latin Amer. Robot. Symp. (LARS), Brazilian Symp. Robot. (SBR), Workshop Robot. Educ. (WRE)*, 2019, pp. 115–120.

[129] D. Kleiser and P. Woock, "Towards automated structural health monitoring for offshore wind piles," in *Proc. Global Oceans*, Oct. 2020, pp. 1–5, doi: 10.1109/IEEECONF38699.2020.9389437.

[130] J. Ross, J. Lindsay, E. Gregson, A. Moore, J. Patel, and M. Seto, "Collaboration of multi-domain marine robots towards above and below-water characterization of floating targets," in *Proc. IEEE Int. Symp. Robotic Sensors Environments (ROSE)*, Oct. 2019, pp. 1–7, doi: 10.1109/ROSE.2019.8790419.

[131] D. F. Campos, A. Matos, and A. M. Pinto, "Multi-domain mapping for offshore asset inspection using an autonomous surface vehicle," in *Proc. IEEE Int. Conf. Auto. Robot Syst. Competitions (ICARSC)*, Apr. 2020, pp. 221–226.

[132] S. Suresh, P. Sodhi, J. G. Mangelson, D. Wettergreen, and M. Kaess, "Active SLAM using 3D submap saliency for underwater volumetric exploration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3132–3138.

[133] Y. Xu, R. Zheng, M. Liu, and S. Zhang, "Keyframe-based imaging sonar localization and navigation using elastic windowed optimization," in *Proc. Global Oceans*, 2020, pp. 1–7, doi: 10.1109/IEEECONF38699.2020.9389045.

[134] P. Sodhi, B.-J. Ho, and M. Kaess, "Online and consistent occupancy grid mapping for planning in unknown environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 7879–7886.

[135] B.-J. Ho, P. Sodhi, P. Teixeira, M. Hsiao, T. Kusnur, and M. Kaess, "Virtual occupancy grid map for submap-based pose graph SLAM and planning in 3D environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 2175–2182.

[136] H. F. Chame, M. M. dos Santos, and S. S. D. C. Botelho, "Neural network for black-box fusion of underwater robot localization under unmodeled noise," *Robot. Auto. Syst.*, vol. 110, pp. 57–72, Dec. 2018, doi: 10.1016/j.robot.2018.08.013.

[137] R. T. S. da Rosa, G. B. Zaffari, P. J. D. O. Evald, P. L. J. Drews, and S. D. C. Botelho, "Towards comparison of Kalman filter methods for localisation in underwater environments," in *Proc. Latin Amer. Robot. Symp. (LARS), Brazilian Symp. Robot. (SBR)*, 2017, pp. 1–6, doi: 10.1109/SBR-LARS-R.2017.8215339.

[138] L. D. L. Barker and L. L. Whitcomb, "A preliminary study of ice-relative underwater vehicle navigation beneath moving sea ice," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7484–7491.

[139] L. D. L. Barker and L. L. Whitcomb, "Performance analysis of ice-relative upward-looking Doppler navigation of underwater vehicles beneath moving sea ice," *J. Mar. Sci. Eng.*, vol. 9, no. 2, p. 174, Feb. 2021, doi: 10.3390/jmse9020174.

[140] N. Bore and J. Folkesson, "Modeling and simulation of sidescan using conditional generative adversarial network," *IEEE J. Ocean. Eng.*, vol. 46, no. 1, pp. 195–205, Jan. 2021.

[141] O. Álvarez-Tuñón, A. Jardón, and C. Balaguer, "Generation and processing of simulated underwater images for infrastructure visual inspection with UUVs," *Sensors*, vol. 19, no. 24, p. 5497, Dec. 2019.

[142] L. G. Prasad Antervedi, Z. Chen, H. Anand, R. Martin, R. Arrowsmith, and J. Das, "Terrain-relative diver following with autonomous underwater vehicle for coral reef mapping," in *Proc. IEEE 17th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2021, pp. 2307–2312.

[143] L. Wei, X. Tan, and V. Srivastava, "Expedited multi-target search with guaranteed performance via multi-fidelity Gaussian processes," 2020, *arXiv:2005.08434*.

[144] D. Nad, C. Walker, I. Kvasić, D. O. Antillon, N. Misković, I. Anderson, and I. Lonč ar, "Towards advancing diver-robot interaction capabilities," *IFAC-PapersOnLine*, vol. 52, no. 21, pp. 199–204, 2019, doi: 10.1016/j.ifacol.2019.12.307.

[145] D. Stojcsics, D. Boursinos, N. Mahadevan, X. Koutsoukos, and G. Karsai, "Fault-adaptive autonomy in systems with learning-enabled components," *Sensors*, vol. 21, no. 18, p. 6089, Sep. 2021.

[146] C. Hartsell, S. Ramakrishna, A. Dubey, D. Stojcsics, N. Mahadevan, and G. Karsai, "ReSonAte: A runtime risk assessment framework for autonomous systems," 2021, *arXiv:2102.09419*.

[147] D. Sousa, M. Luís, S. Sargento, and A. Pereira, "An aquatic mobile sensing USV swarm with a link quality-based delay tolerant network," *Sensors*, vol. 18, no. 10, p. 3440, Oct. 2018.

[148] T. J. Carpenter, R. Ivanov, I. Lee, and J. Weimer, "ModelGuard: Runtime validation of lipschitz-continuous models," *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 37–42, 2021, doi: 10.1016/j.ifacol.2021.08.471.

[149] J. Albiez, S. Joyeux, C. Gaudig, J. Hilljegerdes, S. Kroffke, C. Schoo, S. Arnold, G. Mimoso, P. Alcantara, R. Saback, J. Britto, D. Cesar, G. Neves, T. Watanabe, P. Merz Paranhos, M. Reis, and F. Kirchner, "FlatFish—A compact subsea-resident inspection AUV," in *Proc. OCEANS*, 2015, pp. 1–8, doi: 10.23919/OCEANS.2015.7404442.

[150] R. Cerqueira, T. Trocoli, G. Neves, S. Joyeux, J. Albiez, and L. Oliveira, "A novel GPU-based sonar simulator for real-time applications," *Comput. Graph.*, vol. 68, pp. 66–76, Nov. 2017, doi: 10.1016/j.cag.2017.08.008.

[151] R. Cerqueira, T. Trocoli, J. Albiez, and L. Oliveira, "A rasterized ray-tracer pipeline for real-time, multi-device sonar simulation," *Graph. Models*, vol. 111, Sep. 2020, Art. no. 101086, doi: 10.1016/j.gmod.2020.101086.

[152] S. Bhat, I. Torroba, Ö. Özkahraman, N. Bore, C. I. Sprague, Y. Xie, I. Stenius, J. Severholt, C. Ljung, J. Folkesson, and P. Ogren, "A cyber-physical system for hydrobatic AUVs: System integration and field demonstration," in *Proc. IEEE/OES Auto. Underwater Vehicles Symp. (AUV)*, Sep. 2020, pp. 1–8, doi: 10.1109/AUV50043.2020.9267947.

[153] R. Pi, P. Cieslak, P. Ridao, and P. J. Sanz, "TWINBOT: Autonomous underwater cooperative transportation," *IEEE Access*, vol. 9, pp. 37668–37684, 2021.

[154] J. Hegde, E. H. Henriksen, I. B. Utne, and I. Schjølberg, "Development of safety envelopes and subsea traffic rules for autonomous remotely operated vehicles," *J. Loss Prevention Process Industries*, vol. 60, pp. 145–158, Jul. 2019, doi: 10.1016/j.jlp.2019.03.006.

[155] W. Touzout, Y. Benmoussa, D. Benazzouz, E. Moreac, and J.-P. Diguet, "Unmanned surface vehicle energy consumption modelling under various realistic disturbances integrated into simulation environment," *Ocean Eng.*, vol. 222, Feb. 2021, Art. no. 108560, doi: 10.1016/j.oceaneng.2020.108560.

[156] A. G. S. S. Junior, D. H. D. Santos, A. P. F. D. Negreiros, J. M. D. S. Silva, and L. M. G. Gonçalves, "High-level path planning for an autonomous sailboat robot using Q-learning," *Sensors*, vol. 20, no. 6, p. 1550, Mar. 2020, doi: 10.3390/s20061550.

[157] D. Centelles, A. Soriano, J. V. Martí, R. Marin, and P. J. Sanz, "Uwsimnet: An open-source framework for experimentation in communications for underwater robotics," in *Proc. OCEANS*, 2019, pp. 1–8, doi: 10.1109/OCEANSE.2019.8867460.

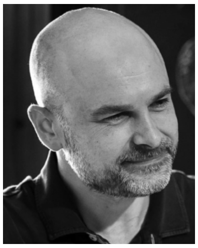
NICOLÒ CIUCCOLI (Member, IEEE) received the M.S. degree in computer and automation engineering and the Ph.D. degree in automation, information and management engineering from Università Politecnica delle Marche, Ancona, Italy, in 2015 and 2019, respectively.

He is currently a Postdoctoral Researcher with the Department of Information Engineering, Università Politecnica delle Marche. He participated in the Student Autonomous Underwater Vehicles Challenge–Europe (SAUC-E) with the NATO Undersea Research Centre (NURC), La Spezia, Italy, in 2016. He worked on navigation algorithms, vision algorithms for industrial systems, and submarine robotics. His main research interests include navigation, guidance, and control of autonomous vehicles, health sensors for divers, industrial systems, automation, and modeling of wastewater treatment plants. His bio-inspired project won the Innovation Award from NURC.

LAURA SCREPANTI (Member, IEEE) received the B.S. degree in biomedical engineering, the M.Sc. degree in electronic engineering, and the Ph.D. degree in information engineering (working on modeling learning from students engaged in activities of educational robotics) from Università Politecnica delle Marche, Ancona, Italy, in 2011, 2014, and 2020, respectively.

From 2015 to 2016, she was a Research Assistant with the Department of Biomedical Sciences and Public Health for the development of a mechatronic infrastructure gathering biometric signals from SCUBA divers during their underwater activity. She has been an Adjunct Professor of systems modeling and identification and a Teacher of real-time systems with the Fondazione ITS Nuove Tecnologie per il Made in Italy. She is currently a Postdoctoral Researcher with the Department of Information Engineering, Università Politecnica delle Marche. She is also an expert on educational robotics and innovative STEAM methodologies to fight against gender stereotypes for the National Institute of Documentation, Innovation and Research in Education (INDIRE). Her research interests include educational robotics, systems modeling and identification, research methodologies, and techniques.

Dr. Screpanti is a member of the IEEE Education Society and the IEEE Robotics and Automation Society.



DAVID SCARADOZZI (Member, IEEE) received the M.Eng. and Ph.D. degrees from Università Politecnica delle Marche (UNIVPM), Ancona, Italy, in 2001 and 2005, respectively.

He is currently an Assistant Professor with the Department of Information Engineering, UNIVPM. He is the author of about 115 publications in refereed international journals, books, and conferences. His research interests include control and optimization of dynamical systems, robotics and automation (motion and interaction control problems in distributed agents; rapid prototyping and mechatronics), educational robotics, with special interests devoted to all aspects regarding the study and development of new robotic tools and lesson plans for teaching e-STrEM subjects, in formal and non-formal education, and underwater robotics and marine technologies, focusing on tools for 3D documentation of sea operative surveys for marine protected areas and archaeological sites. He is a member of Italian Interuniversity Center for the Marine Environment (ISME) where he cooperates with Italian Navy and NATO.

• • •

Open Access funding provided by 'Università Politecnica delle Marche' within the CRUI CARE Agreement