Towards Automated Structural Health Monitoring for Offshore Wind Piles

Dominik Kleiser
Systems of Measurement, Control and Diagnosis (MRD)
Fraunhofer IOSB
Karlsruhe, Germany
dominik.kleiser@iosb.fraunhofer.de

Philipp Woock
Systems of Measurement, Control and Diagnosis (MRD)
Fraunhofer IOSB
Karlsruhe, Germany
https://orcid.org/0000-0002-1212-302X

***Abstract*—Simulation plays an important role in the development, testing and evaluation of new robotic applications, reducing implementation time, cost and risk. In this paper we show a digital twin simulation model of an inspection ROV which is capable of performing structural health monitoring by automated creation of a map of an offshore wind monopile. The data is compared to a known reference model. The digital twin simulation model is extended by a physical sensor data input device to bridge the gap between simulation and testing in water.**

***Index Terms*—Structural health monitoring, monopile, offshore wind, ROV, simulation, digital twin**

I. Introduction

Today an ever increasing number of offshore wind turbines poses a high demand for inspection tasks to ensure proper functioning of the wind energy plant as well as satisfying legislative requirements for safe operation. Scouring and damages like fatigue fractures have to be detected in a very early stage to plan an optimal maintenance schedule [1]. Employing human divers for inspection tasks is time-consuming and often hazardous, which is why the time slots where inspections can be performed are rather scarce. Using ROVs for such tasks has been done, but only in a fully human-operated way. We propose to deploy ROVs which perform the task of mapping autonomously. Mapping is not restricted to purely geometric data but could also include, e. g., corrosion measurements (given that a suitable payload like a stabber with probe is attached to the vehicle) [2]. That paves the way for fully autonomous inspection as soon as autonomous surface vehicles are available to deploy such automated ROVs.

II. Related Work

A. Marine Robotics Simulation

An overview of the current technology for marine robotics simulation is provided by [3]. Most of these simulations build upon the foundation of the Robot Operating System (ROS) [4] and Gazebo [5] as the core simulation engine. Gazebo is a general purpose robot simulator originally developed in 2002 at the University of Southern California and is currently maintained by the Open Source Robotics Foundation. Since then due to its highly modular approach Gazebo has become

This work is funded by the German Federal Ministry for Economic Affairs and Energy

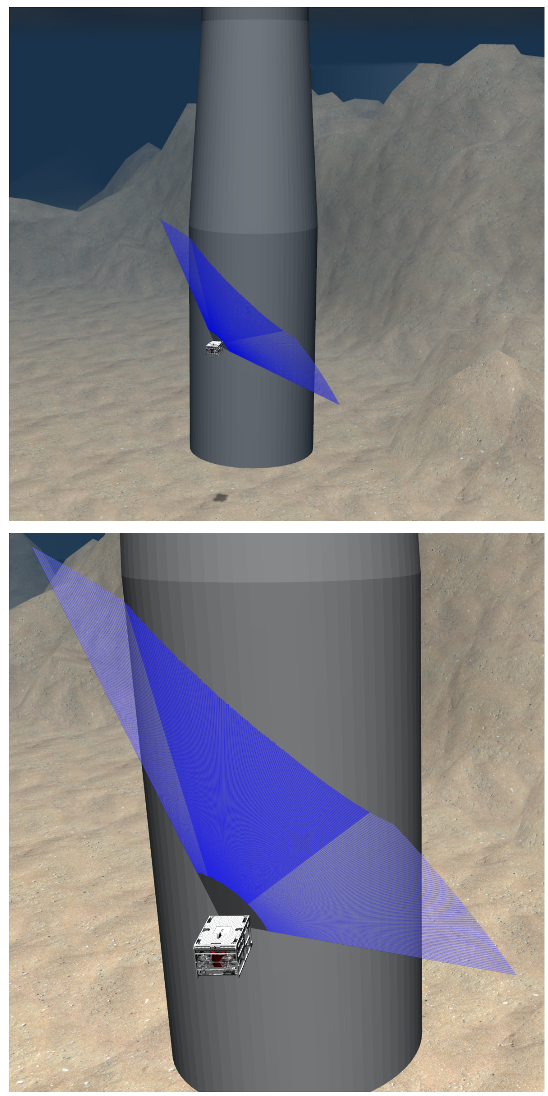

Fig. 1. ROS Gazebo simulation environment showing ROV and monopile. Scanning area depicted in blue. Darker blue denotes hitting the target. Overview (left) and close-up (right)

a mature open source project that is developed by the global robotics community for a wide variety of applications. Gazebo supports the use of multiple physics engines and has an extensive library of common robot sensors, such as camera, laser, sonar, GPS, and IMU, as well as standard noise models that can be parameterized as needed. Interfaces allow the users to interact programmatically with the simulation environment and add custom plugins. In this paper, we make use of the Unmanned Underwater Vehicle Simulator (UUVSim) [6], [7], a set of Gazebo plugins and ROS nodes under development, that are used for supporting and testing the simulation of unnamed underwater vehicles (UUV).

B. Autonomous inspection of underwater structures

The literature on autonomous inspection of underwater structures is scarce. A possible reason for this is that UUV localization, navigation and control are still major research issues due to the fact that a number of navigation aids, such as GPS and similar, are not available underwater [8]. For an extensive overview over the literature of UUV localization and mapping approaches we refer the reader to [9]. Preliminary work in the field of autonomous inspection was carried out in the early 2000s in [10], but seems not to be widely applied. More recent publications [11], [12] seem to focus on the development of ROVs and automatic image acquisition, without comparing the data to ground truth models.

III. Solution

Our solution to the problem consists of a digital twin simulation which employs the ROS, UUV Simulator and data fusion technology. The digital twin can also be connected to a physical sensor unit as a hybrid solution which shows the flexible amount of simulation conducted. The physical sensor unit provides real-world data within a dry lab environment. This enables testing the algorithms on real data. Our solution has the advantage of building upon open frameworks which constitutes a very low entry threshold and allows a user to flexibly adapt to its individual system configuration.

A. ROS and UUVSim

The ROS environment allows to define messages between components and if the messages are defined identically and have the same semantics, it doesn't matter for the system whether the algorithm interfaces to a real system or a simulator. Therefore, the basis for the simulation is UUVSim running in a ROS environment which allows for easy transfer to a real robot. Additionally, ROS provides robot middleware tasks like timestamping, message passing between components, recording and replay. By using the record/replay functionality, any algorithm can be examined thoroughly with different parameterization while the input remains identical between runs.

B. Simulation environment

The simulation environment consists of two parts: A pure simulation and an external sensor. In the pure simulation, it is simulated how a ROV scans an offshore monopile using a multibeam echosounder (MBES). The external sensor for the lab consists of an optical 2D-LiDAR and is optional, being the middle ground before moving to a real ROV. This is due to the similar type of measurement a MBES and a 2D-LiDAR provide. This enables us to have a smooth transition between pure simulation and real robotic inputs.

C. Sensors

After including the model of our ROV into UUVSim we needed to establish the sensory inputs. As the final ROV sports a BlueView MB2250-45, we added a multibeam sonar with 256 beams (range 1–10 m, 90° FOV, 40 Hz) as sensor. As a basis we used the UUVSim multibeam sonar which essentially is a laser scanner (2D-LiDAR) simulation. Due to the similar measurement principle this is appropriate, however, we added gaussian noise to the simulated measurements to account for the much more noisy characteristic of a sonar compared to a laser scanner. The sonar is mounted facing forward on a 45° angle to the vertical in order to have a wide swath while going both sideways and up and down. We don't expect diagonal motion in an inspection task except for short periods while taking a turn. The virtual sensor is modeled such that the specs of the virtual sensor match the real sonar as close as possible. The range of the sonar is restricted to 10 meters, approximately matching the range of the BlueView 2.25 MHz system. Furthermore, a DVL sensor provides velocity values relative to the seafloor. Most real-world systems are equipped with a DVL so the presence of a DVL is a fair assumption to make.

D. Mission

The mission plan for inspection is basically a vertical lawn mower pattern covering a 120° sector each pass. In the simulation we disturbed the path to be a bit wavy to account for platform motion due to underwater currents. Using the described inspection path should avoid to wrap the umbilical around the monopile while at the same time being energy-efficient.

E. External sensor

The system is extended by an external sensor unit comprising an inertial measurement unit (IMU) and a 2D LiDAR. That way, the inputs to the simulation can be redirected to stem from the external sources rather than the software simulation. The external sensor hardware is depicted in Fig. 3. Using this setup, example scans can be performed quickly in the lab without explicitly programming given trajectories. For instance, Fig. 4 shows how the sensor carrier is used to scan a pipe as a mock-up for a monopile. This facilitates a quick development loopback with dry runs in the lab instead of performing laborious wet tests.

IV. Mapping

A. Data fusion

In order to create a 3D point cloud of the sonar scans we make use of an IMU to track the position and orientation of

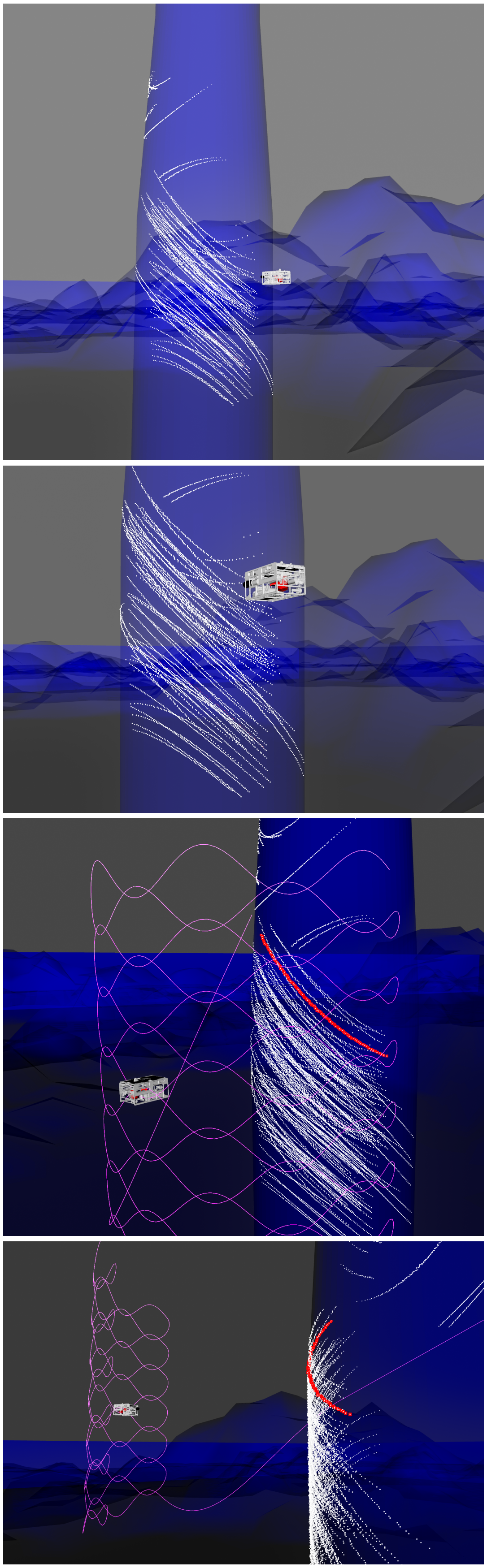

Fig. 2. Synthetic sensor data acquisition. AUV is untextured, the DVL sensor inside the ROV is colored red. Top row: Monopile ground truth and synthetic sonar measurements. Bottom row: Mission path is additionally shown in magenta and current measurement highlighted in red.

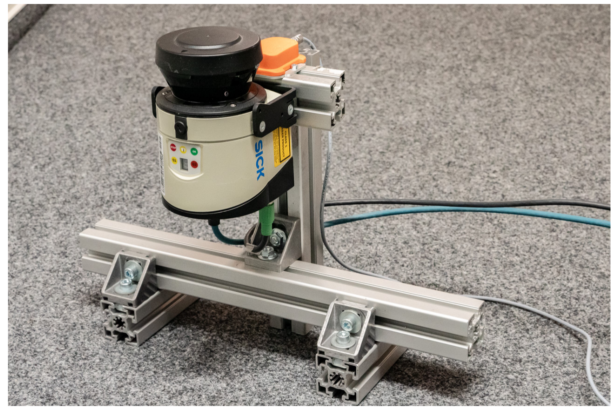

Fig. 3. External hand-held sensor setup with IMU and optical laser scanner.

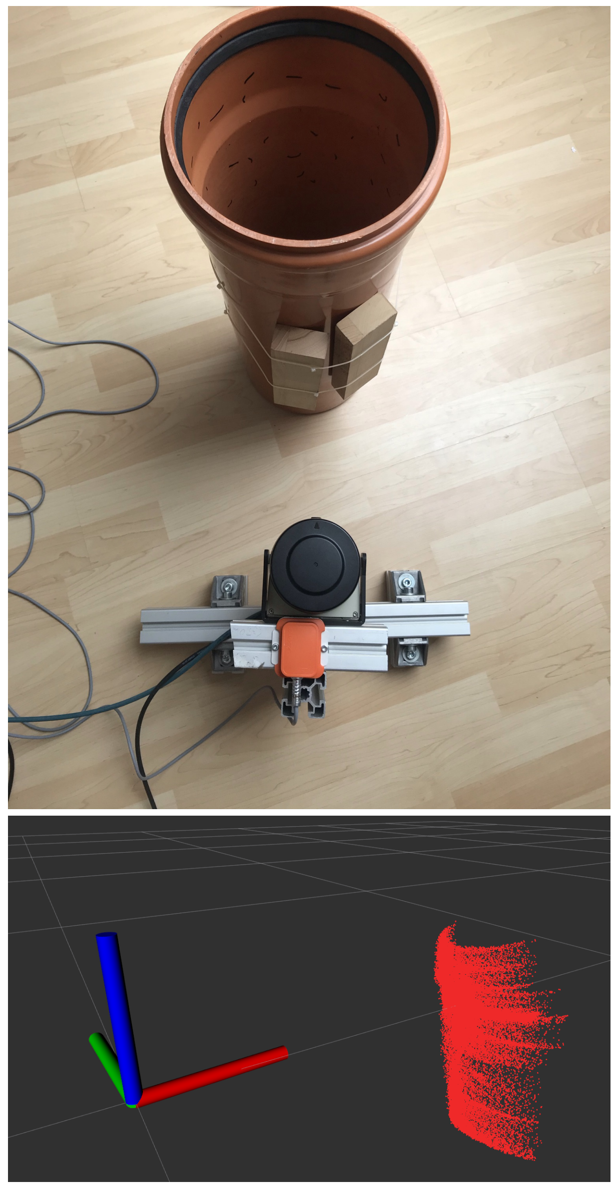

Fig. 4. Physical scan of a pipe with our hand-held sensor carrier. Top: Hand-held sensor carrier in front of the pipe. Bottom: Visualization of the scanned pipe after fusing with the IMU data.

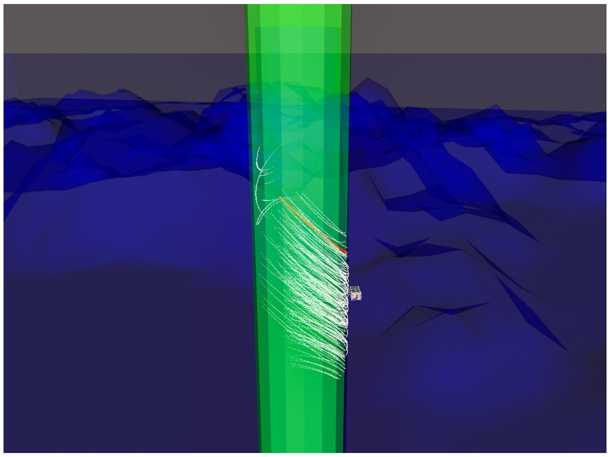

Fig. 5. Cylinder surface best approximating the data is depicted in green. The angular shape of the green cylinder comes from the low vertex resolution of the visualization tool rviz when visualizing parameterized cylinders.

the sonar sensor via dead reckoning. Inertial sensors offer good signals with high rate during fast motions but are prone to accumulated drift due to double integration during estimation of position [13]. Therefore, we combine the accelerometer and gyroscope signals with an additional DVL sensor providing depth and velocity measurements. The sensor fusion is done by employing an Extended Kalman Filter using an existing ROS package based on the work described in [14].

B. Parameterized cylinder fitting

As a monopile resembles a cylinder, and cylindrical structures are also common as parts of jacket constructions, we employed geometric cylinder fitting. This is a solution when there is no CAD data of the structure available. We fit a mathematical cylinder formula to the generated 3D point cloud to obtain a cylinder representation of the monopile. We apply MLESAC [15], a more robust variation of the RANSAC algorithm, by estimating 6 cylinder coefficients, i.e., a point on its axis and its axis direction. The cylinder radius is set to a given value. If the axis direction is not within an expected angle the estimation is rejected. The procedure is executed repeatedly with a frequency of 0.5 Hz to obtain a continuously improving estimation as more sonar data arrives at the processing unit. To evaluate the performance of the algorithm we compare the fitted model with the ground truth world model in RViz (see Fig. 5).

C. CAD model fitting

As the cylinder is a quick but only partially accurate way of modeling a monopile, we used a full 3D CAD model of the actual monopile under inspection instead of fitting a purely geometric cylinder to the data. That way, the model error is reduced and a more accurate map is obtained. The CAD model was created from a technical drawing of a monopile in SolidWorks. Then, from a triangle mesh of the CAD drawing, a dense point cloud was sampled resulting in a model (truth) point cloud of approximately 182,000 points (see Fig. 6).

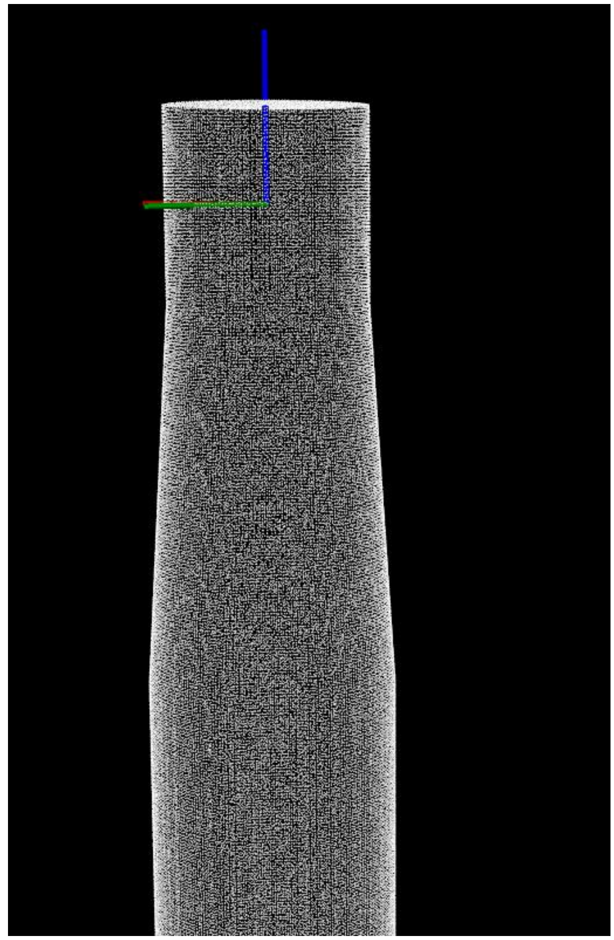

Fig. 6. Sampled CAD model of a monopile.

In order to obtain a transformation between the ground truth point cloud and the measured points we employ an iterative nonlinear optimizer based on the Levenberg-Marquardt algorithm with is a more robust extension of the iterative closed point (ICP) technique. The method is described in [16]. As before, we repeat the procedure with a frequency of 0.5 Hz to update the estimated transformation periodically.

Fig. 6 presents the result of the CAD fitting procedure. The method is more flexible than the RANSAC procedure described above. As we now fit a generic CAD model the procedure can be applied to arbitrary underwater foundation structures when 3D CAD data is available.

V. RESULTS

Using that simulation environment, a very large part of the development can be performed in the lab and the expensive wet tests can be reduced to a minimum as the very same software components are used, only the sensor input sources are different.

VI. CONCLUSION AND OUTLOOK

In this paper, we developed a twin simulation model for underwater inspection tasks based on the UUV Simulator.

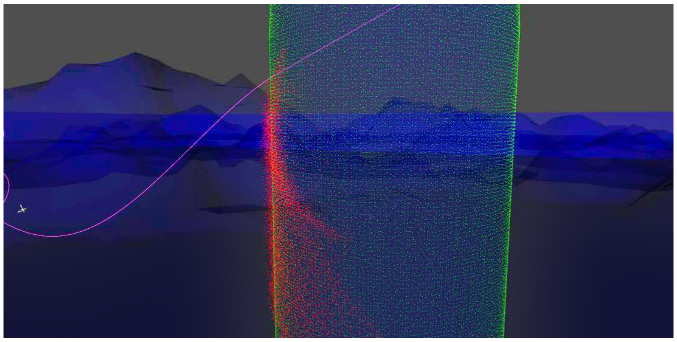

Fig. 7. Point cloud sampled from the monopile CAD model (green) fitted to the measured point cloud (red).

Moreover, we have created a hand-held data input device integrating real physical sensor data into the simulation framework. In the near future, we are planning to conduct field experiments in order to validate the system and the models that we use in a real-world inspection scenario. Future work will concentrate on adding defects such as cracks or dents to the simulation to test the robustness of our mapping approach. Also, we plan to improve the visualization of deviations between the the actual measured 3D point cloud and the ground truth model by creating 2D difference maps. Using an imaging sensor seems a natural extension to the concept. Given a calibration between camera and sonar, camera intensity values can be used to colorize the sonar measurements which may give an operator greater insight than camera images alone. Furthermore, we plan to make use of the UUVSim integrated current modeling capabilities to more realistically mirror the environmental conditions in the mission target area.

REFERENCES

[1] L. Ziegler and M. Muskulus, "Lifetime extension of offshore wind monopiles: Assessment process and relevance of fatigue crack inspection," in *12th EAWE PhD Seminar, DTU Lyngby, Denmark*, 2016.

[2] T. Mathiesen, A. Black, F. Grønvold, and P. Alle, "Monitoring and inspection options for evaluating corrosion in offshore wind foundations," *NACE Corrosion-2016, paper no C-2016*, vol. 7702, 2016.

[3] D. Cook, A. Vardy, and R. Lewis, "A survey of auv and robot simulators for multi-vehicle operations," in *2014 IEEE/OES Autonomous Underwater Vehicles (AUV)*, 2014, pp. 1–8.

[4] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system." [Online]. Available: https://www.ros.org

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

[6] 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 *OCEANS 2016 MTS/IEEE Monterey*. IEEE, sep 2016. [Online]. Available: https://doi.org/10.1109%2Foceans.2016.7761080

[7] "UUV Simulator." [Online]. Available: https://github.com/uuvsimulator/uuv_simulator

[8] R. P. Vio, R. Cristi, and K. B. Smith, "Uuv localization using acoustic communications, networking, and a priori knowledge of the ocean current," in *OCEANS 2017 - Aberdeen*, 2017, pp. 1–7.

[9] L. Paull, S. Saeedi, M. Seto, and H. Li, "Auv navigation and localization: A review," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2013.

[10] G. L. Foresti, "Visual inspection of sea bottom structures by an autonomous underwater vehicle," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 31, no. 5, pp. 691–705, 2001.

[11] J. Choi, Y. Lee, T. Kim, J. Jung, and H.-T. Choi, "Development of a rov for visual inspection of harbor structures," in *2017 IEEE Underwater Technology (UT)*. IEEE, 2017, pp. 1–4.

[12] D. McLeod and J. Jacobson, "Autonomous UUV inspection – Revolutionizing undersea inspection," in *OCEANS'11 MTS/IEEE KONA*, 2011, pp. 1–4.

[13] D. Vissiere, A. Martin, and N. Petit, "Using distributed magnetometers to increase IMU-based velocity estimation into perturbed area," in *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 4924–4931.

[14] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, July 2014.

[15] P. H. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer vision and image understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[16] A. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image and Vision Computing*, vol. 21, pp. 1145–1153, 04 2002.