From market-ready ROVs to low-cost AUVs

Jonatan Scharff Willners[b], Ignacio Carlucho[a], Tomasz Łuczyński[b], Sean Katagiri[b], Chandler Lemoine[a], Joshua Roe[b], Dylan Stephens[a], Shida Xu[b], Yaniel Carreno[b], Èric Pairet[b], Corina Barbalata[a], Yvan Petillot[b], Sen Wang[b]
[a] *Department of Mechanical Engineering, Louisiana State University*, Baton Rouge, USA
{icarlucho, clemo17, dstep12, cbarbalata}@lsu.edu
[b] *Institute of Sensors, Signals and Systems, Heriot-Watt University*, Edinburgh, UK
{j.scharff_willners, t.luczynski, s.katagiri, joshua.roe, sx2000, y.carreno, eric.pairet, y.r.petillot, s.wang}@hw.ac.uk

***Abstract*—Autonomous Underwater Vehicles (AUVs) are becoming increasingly important for different types of industrial applications. The generally high cost of AUVs restricts the access to them and therefore advances in research and technological development. However, recent advances have led to lower cost commercially available Remotely Operated Vehicles (ROVs), which present a platform that can be enhanced to enable a high degree of autonomy, similar to that of a high-end AUV. In this article, we present how a low-cost commercial-off-the-shelf ROV can be used as a foundation for developing versatile and affordable AUVs. We introduce the required hardware modifications to obtain a system capable of autonomous operations as well as the necessary software modules. Additionally, we present a set of use cases exhibiting the versatility of the developed platform for intervention and mapping tasks.**

***Index Terms*—Autonomous Underwater Vehicle, Remotely Operated Vehicle, Hardware, Underwater Manipulation, Autonomy, Robot Operating System**

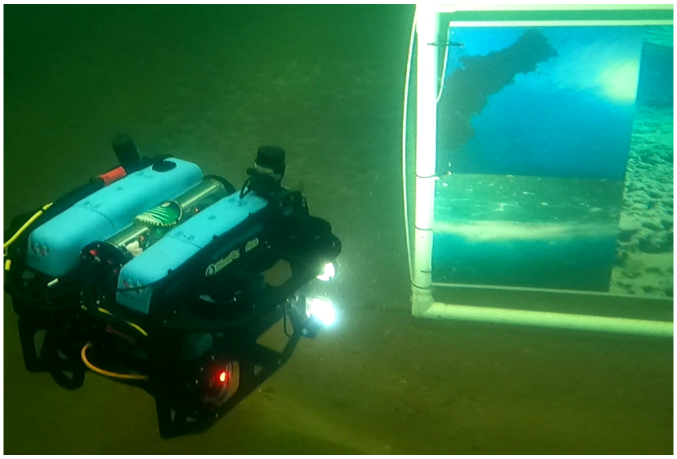


Fig. 1: The developed AUV based on the BlueRov2 ROV

I. Introduction

ROVs were first used for recovering torpedoes in the 1950s. Since then the technology has been adopted by a multitude of industries as their main approach for subsea operations. These robots have been previously utilized for tasks as diverse as, drilling for mineral and rock sampling [1], video survey of oil and gas platforms [2] and measuring the roughness of the seabed [3]. However, continuous operation of ROVs can be costly as they require constant monitoring from an operator, who is connected with a tether to the ROV from a support ship [4]. Operating such a system can be extremely costly, with ships potentially being over £50M, and with reported costs of £30K per day [5]. Additionally, this imposes a risk to operators who need to perform the launch or recovery of the ROV on the deck, specially under rough weather.

Nevertheless, in the search for reducing cost and mitigating the risk to operators, research has been focusing on autonomous alternatives, such as AUVs. These vehicles, do not require constant monitoring from operators, and can perform different types of operations such as surveying [6] or pipeline tracking [7] on its own. Autonomous operations can therefore save a lot of time, as well as ease the deployment, which reduces the operative cost. It does however, depend on advanced and reliable software, which increases the development cost.

Previous research has shown different types of AUVs configuration and detailed how they can be constructed. Nessie-V, a 6 Degree of Freedom (DoF) research AUV, was developed by Ocean System Laboratory (OSL) at Heriot-Watt University to be used in research focused on inspection and autonomous missions [8]. Authors in [9] detail Girona 500, an AUV with reconfigurable payload and propulsion system. In [10], Sparus II AUV is described which is a 5 DoF vehicle. Both Girona 500 and Sparus II are powered by Component Oriented Layer-based Architecture for Autonomy (COLA2) [11], and commercially available with a base cost of around $50,000. The modularity of these robots makes them suitable for multiple purposes including mapping and exploration. In [12] an AUV for acoustic imaging surveys, the Ictiobot-40, was developed by the INTELYMEC group. However, the development of these systems from ground up is extremely time-consuming and therefore not feasible for a large number of scientists.

Today, we see a shift, where cheaper ROVs are entering the market being affordable for hobbyists, small companies, and marine scientists. In this sense, the introduction of

This work was supported by the EPSRC funded ORCA Hub (EP/R026173/1) and EU H2020 Programme under EUMarineRobots project (grant ID 731103).

cheaper ROV and sensors enables the possibility to combine commercial-off-the-shelf (COTS) products to create an AUV at a lower cost and shorter time than building one from scratch. However, as expected with a lower price, there are trade-offs in terms of quality and capability. Most noticeable are the limited depth ratings, on-board sensors and the effect of the thrusters. The new affordable models are rarely rated for more than 100m depth, they lack navigational sensors to reliably estimate their position, and due to their small size cannot handle well heavy sea-currents or waves. However, new COTS ROVs can be a good starting point in which an autonomous system can be built on top of, leading to savings both in time and cost. The cheaper ROVs can be improved with additional sensors for navigation, and improved software. Additionally, most of the applications of AUV are for usage in sheltered areas. Hence, the depth rating is sufficient and they are not exposed to the same amount of external disturbances as a work-class ROV at open sea.

In this article we present how modern COTS products can be combined to turn a ROV into an AUV for a fraction of the price when compared with a hovering capable AUV. Our design is based on the BlueRov2 heavy configuration, shown in Fig. 1, a low-cost ROV capable of working in depth of up to 100 m [13]. We provide in this article a full description of the hardware modifications required for converting the ROV to a fully autonomous AUV, as well as the required software modules. Furthermore, we present real-time results for two different use cases i) Stereo vision with slam ii) Underwater payload manipulation. These results showcase the versatility of our proposed AUV platform for performing a variable number of underwater missions, while costing a fraction of the price.

The rest of the article is organized as follows. Section II presents the base platform used. Section III presents the hardware changes on the ROV, while Section IV presents the required software modifications. Section V introduces two use cases of the AUV, and finally Section VI presents some conclusions.

II. BLUEROV2 PLATFORM

The BlueRov2 is a COTS ROV produced by BlueRobotics [13]. This small ROV has a depth rating of 100 meters, which provides excellent capabilities for exploring underwater environments. The small footprint of the vehicle and its weight under 12 kg make it easy to transport and deploy. Furthermore, the BlueRov2 comes equipped with a tether of up to 300m, which allows operators to control the vehicle remotely by means of a dedicated software package. The ROV is internally powered, using a li-ion battery of 14.8V and 18Ah, providing ample energy autonomy.

Different versions of the vehicle are commercially available, with a starting price of £2.5K. We based our design on the BlueRov2 heavy configuration, which comes with 8 thrusters, arranged 4 in vertical position and 4 horizontally. This thruster configuration allows for a full 6 DoF control. Additionally, the BlueRov2 is equipped with a set of sensors that serve as feedback for basic control and exploration. The BlueRov2 has an Inertial Measurement Unit (IMU) (including accelerometer, gyroscope and compass), a depth sensor and a single-beam echosounder. Furthermore, it also has a front-facing tilting low-light camera, that can be enhanced with forward lights for underwater environments. Moreover, the BlueRov2 also has current and voltage sensing capabilities, as well as leak detection circuits, extremely useful for secure underwater operations. With regards to the computing capabilities, the BlueRov2 has two computers on-board, the Pixhawk flight controller [14] and a Raspberry Pi called a companion.

The BlueRov2 provides an exceptional starting platform for developing an autonomous solution. Based on the current system, in the following sections we will show what are the required hardware and software modifications necessary for transforming the COTS ROV into a fully functioning AUV.

III. HARDWARE

Commercially available ROVs often have a very limited set of sensors, usually able to measure orientation and depth and cameras to aid the operator. To enable the platform to operate reliably in an autonomous fashion, additional hardware is often desired. In this section, we describe the necessary hardware modifications done on the BlueROV2 vehicle, to obtain autonomous capabilities. We divide these modifications in four subsystems: i) Navigation sensors, ii) Perception sensors, iii) Data processing and decision-making system, iv) Communication system.

A. Navigation Sensors

There are a number of additional sensors that can support the quality of localisation, *e.g.*, Ultra Short Baseline (USBL) can provide an absolute pose information by acoustic communication with an external transponder, IMU can provide the orientation of the vehicle as well as angular velocity readings, and Doppler Velocity Log (DVL) allows for fairly accurate linear velocity estimates. Inertial sensors and DVL are sometimes combined in a single unit to provide Dead Reckoning (DR). The precision of the data provided by sensors used in DR are usually related to the price. Designing a full localisation system at a low budget requires therefore a careful consideration as to which sensors should be selected.

In the interest of minimizing the cost, no USBL was added to the systems used in our research. The ROV used comes equipped with an IMU, including a magnetometer, and a depth sensor. This gives absolute measurement for depth and orientation but no reliable information about movement in the x-y plane. To enable a full 6 DoF pose estimation we equipped the vehicle with a DVL. We utilize the DVL A50, from Waterlinked, due to its small footprint and accuracy on low seabeads. Additionally, this DVL can be integrated with the ArduSub Companion. The DVL was mounted to the lower section of the AUV using a custom made 3D printed piece. The piece was engineered to effectively mount and stabilize the DVL while not affecting the beams of the transducer. With this set of sensors on board the ROV, sensor fusion can then be used for pose estimation. More details on this will be presented in Section IV.

B. Perception Sensors

Underwater perception is usually performed either by acoustics (sonar) or optical (camera) sensors. While cameras are widely available and the data is easy to understand for humans, the visibility in underwater environments is often limited due to water conditions [15], thus cameras are mainly effective at short distance (below a few metres). On the other hand, sonars are not affected by the visibility in the water and are therefore a more reliable choice when the visibility is limited. However, sonars are expensive and can be hard to interpret [16]. Furthermore, the resolution of the data generated by sonars is also much lower, compared to cameras. Additionally, multiple cameras can be combined into stereo or multi-view setup, allowing for real-time 3D reconstruction, which can be a very valuable source of information. We equipped the platform developed in this paper with a set of stereo cameras with a custom-made sensor. Additionally, we include in the enclosures the required computational capabilities for processing the visual information. When designing a stereo camera system, there are a number of factors to consider. Parameters like sensor type, lens, interface, and baseline selection can be modelled to match the requirements specific to the application [17]. However, many factors may change in the development process, so two design principles are highlighted in the paragraphs below.

First, we present the design of a single housing to incorporate all cameras. This allows for easy hardware integration, camera synchronization, and testing various hardware configurations. Processing units can also be easily added to process the images on-board. Fig. 2 (right) shows our vision system designed according to this approach. This requires a dedicated underwater housing, which is difficult to design and produce, as well as much more expensive than typical cylinder-shaped enclosures. Furthermore, the relative pose of the cameras is fixed and cannot be changed. This approach is very useful during development but loses most of its advantages in the context of more mature systems, produced in larger numbers.

The second approach assumes that each camera is enclosed in a separate housing, and the processing units are also kept separately. This simplifies the construction of the enclosures and the components can be easily rearranged as necessary. Fig. 2 (left) shows the system implemented in the AUV, following these principles. The downside of this approach is that all components must be connected with underwater cables. This may become challenging as the cable and connectors need to rely on the protocol for data (and power) transfer, which might not always be commercially available.

C. Data Processing and Decision-Making System

To enable robust and reliable autonomy, the system needs to process large amounts of data, with as low latency as possible. This is a challenging task that grows with each software sub-module, especially in the context of algorithms responsible for data from the perception systems, which need to process a large amount of data at a high rate. Therefore, the selection of the embedded computers may heavily influence the performance of the system. Simpler tasks, such as running navigational sensors and handling the communication can be done on cheaper units (*e.g.*, Raspberry Pi). Image data can be processed efficiently on FPGA, but these systems are not very flexible and increases the time for development, and are therefore better suited to finished and stable software. Therefore, since we are mostly working on the development level, we utilized embedded computers from the Nvidia Jetson family. Thanks to the included GPU, this boards are extremely effective for testing and implementing different vision algorithm in real-time. Additionally, with some care during the mounting stage the aluminium water tight enclosures from the BlueRobotics family can be utilised to provide efficient cooling to the processing units.

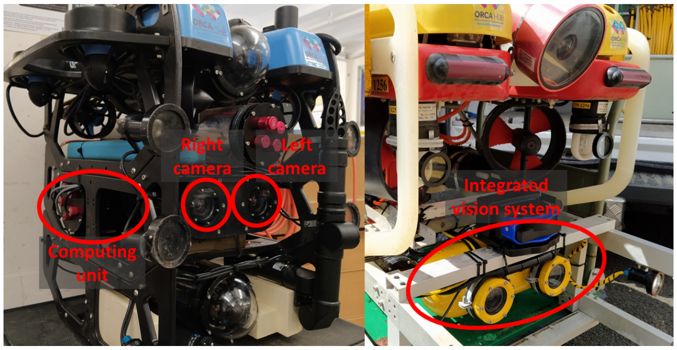


Fig. 2: Two vision system designed according to different principles: left - a modular design with each component in smaller separate enclosure; right - an integrated system with the cameras and all additional electronics are enclosed in a single housing.

D. Communication System

Ideally, an AUV should work independently, with little or no communication with the operator. However, during the development process to ensure safety of operations, a constant communication link with the vehicle is required. This can be achieved in a few ways. Typically, the development process is conducted with a tether attached. This allows for continuous observation of the systems and streaming data to the surface, where more powerful processing units can be used. This allows the development of algorithms on the surface, before deciding on how much on-board processing power is needed on the embedded computers. The tether can also be moved to a communication buoy able to transfer data to the operator wirelessly. The vehicle can then move more freely and all the processing can be performed on-board, while at the same time, high-speed connections are available for data monitoring and debugging.

However, to fully become an autonomous vehicle the tether should preferably be removed completely as this can be a limitation for navigation, especially in structured and cluttered environments where the tether can get entangled. Without a tether, the only reliable source of communication is by using acoustic communication. There have been recent progress in optical communication which can provide a high data transfer rate but this has a limited range, can be highly affected by additional light sources (*e.g.*, the sun), and requires a direct line of sight. Acoustic communication is usually low bandwidth but capable of transmission for many kilometres. In the developed platform, we removed the fixed tether connection present in the ROV, and replaced it with an underwater connector from suburban marine [18]. This connector utilizes the same penetrator size as the default Bluerov2, which allows for seamless integration, but can be easily connected or disconnected as required. The advantage of such a connector is that it allows a plug and play characteristic to the AUV. In this way, the vehicle can perform tasks autonomously by disconnecting the tether completely, but if necessary, the tether can be easily connected and used, for example, in conjunction with a communication buoy to transmit mission information. This creates a versatile research platform that allows for rapid testing of algorithms and alternative communication techniques.

IV. Software

In this section, we present the core software requirements necessary for obtaining an AUV capable of following predefined waypoints. The developed system consists of the interpreter node, the pose estimation system, the control architecture and the waypoint pilot. This general architecture will be referred as the navigation stack, upon which more intelligent and sophisticated autonomous behaviours and approaches can be built upon. Application examples of the navigation stack are presented in Section V.

A. *Interpreter Node*

Commercially available ROVs usually come with a dedicated computer to supply the user with an interface for control and observation capabilities, *e.g.*, by sending velocity or thrust commands for controlling the movements of the ROV or displaying sensor data as depth, heading and camera images. This computer is defined as a frontseat driver and usually has limited control capabilities. To enable more complex autonomous behaviours, a backseat driver can be used. The backseat driver is able to use the data from the ROV's sensors and command the ROV through an interpreter node, which enables user-defined software to take control over the ROV. The backseat driver can be deployed either as software on the frontseat's dedicated computer or as a separate computer connected to the frontseat. The architecture of the frontseat and the backseat can be seen Fig. 3.

Robot Operating System (ROS) is being used in many robotics applications as the de facto standard for handling internal communication, offering an easy approach to a modular software system [19]. Guided by this, we designed our system to leverage ROS communications. ROS allows for modular design and easy integration of additional modules, given that the communication layer is fast enough to transport the data as needed. All our vehicles are equipped with 1 GB network on board, which proved to be sufficient for performing the communicating and control tasks.

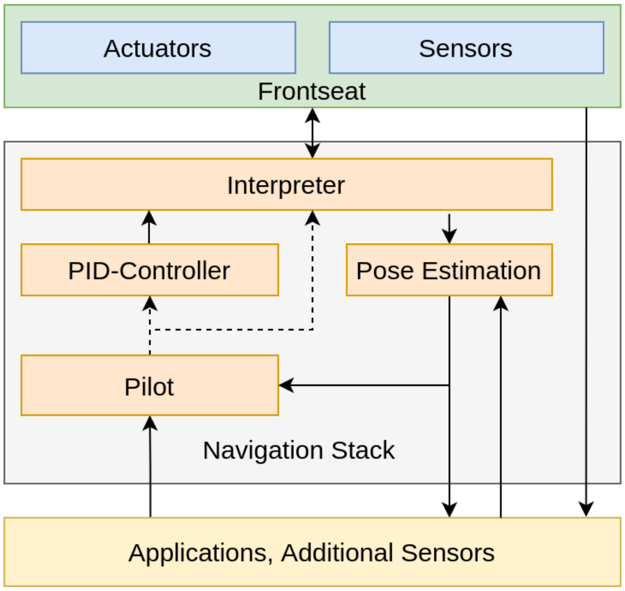


Fig. 3: A navigation stack is used to connect the frontseat to the backseat to enable autonomous operation. If the frontseat can only be controlled using velocity/force commands a PID-controller is used for waypoint navigation.

Since the Bluerov2 includes a frontseat computer, we include an interpreter node that can be used to handle the communication between the front and backseat. Furthermore, this allows the rest of the system to be platform agnostic and easily transferable between robots, allowing for faster deployment of algorithms in different platforms.

B. *Pose Estimation System*

To enable autonomous navigation, the robot needs the ability to estimate its own position. For terrestrial robots, the access to GPS gives the possibility to measure an absolute position in the world at a reasonably high frequency. However, electromagnetic signals are absorbed in water. Hence, the same technology cannot be applied for an AUV while submerged. Instead DR can be used to estimate the current position based on the internal sensors. To enable a full estimation in 6 DoF the vehicle utilizes the IMU for orientation, a pressure sensor for depth and the DVL for velocity estimation. In our navigation stack we use the Robot Localization Package from ROS which fuses the sensor data through an Extended Kalman Filter (EKF) [20] to perform DR. DR is however based on the integration of data containing potential noise and biases, hence the error and uncertainty can therefore grow without bound. An alternative to DR is to use natural features as references for estimating the pose using *e.g.*, visual [21] or acoustic [22] simultaneous localisation and mapping (SLAM). A third approach is to use artificial landmarks such as acoustic beacons to communicate with the AUV to perform Long Baseline

(LBL), Moving-LBL (MLBL) [23], [24], or USBL [25]. The different approaches can also be combined for improved pose estimation [26], [27].

In our implementation, we use additional odometry from a visual SLAM node fused with the DR generated by the frontseat, to improve the position estimate, further described in section V-A. However, if the frontseat is able of performing position control based on its DR and we have another pose estimation system, the two will drift apart over time. As we assume that the SLAM odometry is more reliable over time, due to the fusion of more data, we continuously need to provide a transformation between the backseat estimation and the frontseat estimation. This enables the backseat to operate in one coordinate frame, while the frontseat operates in one based on the DR.

C. Control System

To enable autonomous navigation between waypoints, a position and orientation controller is needed. We employ a generic navigation stack able to cope with two type of controllers in the frontseat: 1) position control and 2) velocity/force control (through *e.g.*, a joystick). However, if the vehicle's frontseat is not endowed with this capability, the position can be controlled from the backseat using *e.g.*, a cascaded PID-controller [28].

D. Waypoint Pilot

A waypoint pilot is used to keep track of which waypoints, and in what order to visit them. The pilot serves as the interface that all high-level planning algorithms will use to control the robot. This node is able to improve the control of the robot indirectly by interpolating a path between waypoints as well as querying a path planner to find a collision-free path between two waypoints. As most hovering capable ROVs and AUVs are controllable freely in 3D, a geometric planner such as informed Rapidly-exploring Random Tree* (I-RRT*) [29], implemented using Open Motion Planning Library (OMPL) [30] can ensure collision-free trajectories within the known environment. If the vehicle needs to incorporate the kinematic constraints of the robot, planners as presented in [31]–[33] can be used.

V. Experimental use cases

A. Stereo Visual SLAM and Autonomous Inspection

We include preliminary results that show how mapping can be performed with the developed system. The vehicle has been used with stereo vision ORB-SLAM [34] with online extrinsic calibration of a DVL, to incorporate the sensor measurements (velocity, depth and orientation) into the visual pose estimation. The approach was tested with ground truth as seen in Fig. 4a. The SLAM system can be used by the navigation stack to improve the pose estimation by taking external features as reference in addition to DR.

1) Autonomous Robust Inspection: An integrated SLAM with active relocalisation for map-merging/loop-closure was deployed to test a robust underwater SLAM system [35]. The approach combines task-planning [36] and viewpoint generation with the SLAM system to endow the system with a map-merging procedure when visual tracking is lost. This is based on random sampling in a region around key-frames in the prior map, and simulating the sensor to find a location with a high probability to relocalise based on previously seen features.

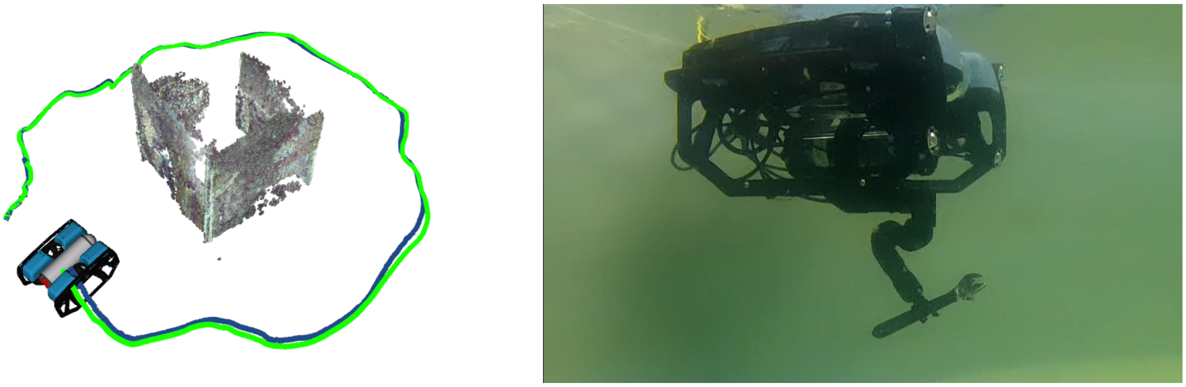
a) High-Level Task Planning: AI planning solutions have shown promising results while solving complex missions in the underwater domain [37], including environment's inspection. For the use case we present in this paper, the high-level task planner [36] generates a *plan*—sequence of actions that leads the robot from the initial to the final state where all goals are achieved—that allows the mapping of a structure amongst other actions. The plan actions are dispatched to the low-level system, including the hardware and software components previously discussed in this paper, for execution. Our system combines goal-based mission planning, based on a temporal planner [38], and a knowledge-based framework to achieve plans for dynamic problems. Therefore, this framework can adapt the initial plan to maintain robot operability when unexpected changes (not considered in the initial plan) occur. The knowledge-based framework encloses the Situational Evaluation and Awareness (SEA) component. SEA is a failure solver which acts when failures occur, driving the robot from the failure state by proposing alternative behaviours or updating its knowledge (that translate on the generation of alternative plans). SEA evaluates the characteristics of failures to complete the global goals (consider in the initial plan) while introducing local goals that enable recovery from failure states. The possible mission failures embedded in SEA consider the tolerance analysis, anomaly and fault detection provided for past experiments using different approaches [37]. SEA creates a bridge between plan reasoning and execution to support system robustness when running for long periods. Combining the high-level planning approach and the low-level system, including a viewpoint planning module, allows implementing a robust structure mapping while mission survivability is guaranteed. When poor mapping is notified* by the low-level system, SEA advises the generation of a new plan that includes local goals for AUV relocalisation concerning the initial map. These local goals represent points provided by the viewpoint planning module, where it is highly expected that the AUV can merge the actual map with the original.

B. Payload Manipulation

Furthermore, we also include results for another use case that shows how the AUV can be integrated with an underwater manipulator to perform intervention missions. The use case uses the Reach Alpha 5 [39] underwater manipulator also

For ArduSub: https://www.ardusub.com/developers/dvl-integration.html

*The AUV losses track of the features it was using for localisation leading to the failure notification. In this situation, the robot will find new features and will start building a new map which is not optimal when mapping a structure.



(a) The AUV performing SLAM. The green line shows ground truth (measured from an underwater motion tracking system) and the blue line shows the SLAM pose estimate.

(b) The AUV performing an intervention mission in an underwater environment using the Reach Alpha 5 manipulator.

Fig. 4: The developed AUV has multiple use cases such as autonomous mapping and underwater intervention.

integrated with ROS, for manipulating unknown payloads in underwater environments.

In this test case, we focus on the low-level control of an underwater manipulator, mounted on an autonomous underwater vehicle, that was required to manipulate different payloads of unknown shape and size. Due to the highly nonlinear dynamics and the unknown hydrodynamics effects of the payloads, the performance of the control system degrade quite rapidly. To mitigate these effects, we developed a data-driven model predictive controller, based on neural networks. By utilizing a neural network, we were able to obtain a more accurate model, that inherently takes into consideration the environmental disturbances. Moreover, to account for the variations caused by the payloads, we enhance the control system with an online adaptive tuning strategy based on adaptive interaction theory. This adaptive mechanism takes into consideration the prediction window used in the optimal controller, allowing to take predictive tuning actions to improve the overall performance. We named this algorith the Adaptive Neural Network Model Predictive Control (AdaNNMPC) [40]. In Fig. 5 we show the results obtained when utilizing the AdaNNMPC algorithm for a case study in which the arm is manipulating an unknown object, in this case a wrench, as shown in Fig. 4b. The reference position $(\mathbf{r}_t)$ in joint space is set as $\mathbf{r}_t = [2.5, 2.0, 1.6, 2.0]$ radians. It can be seen how the manipulator is able to reach the desired reference position in a short time, and with minimal overshoot, compensating for the variations caused by the payload.

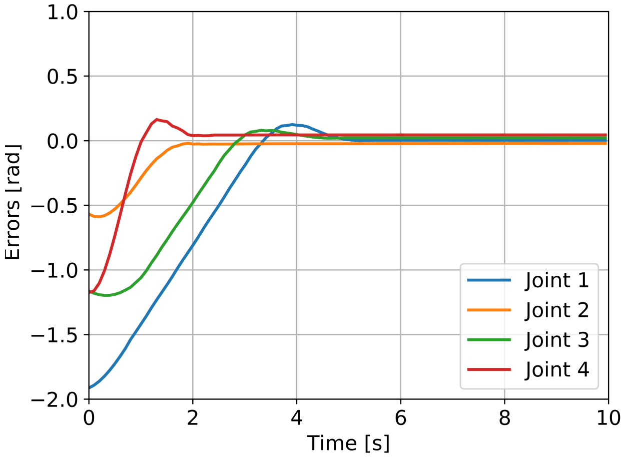


Fig. 5: AdaNNMPC algorithm with arm holding a wrench with $\mathbf{r}_t = [2.5, 2.0, 1.6, 2.0]$ radians

VI. Conclusion

In this article, we have presented how a COTS ROV can be turned into an AUV for fraction of the price, compared to commercially available AUVs. Likewise, the time saved by starting from an operating base platform compared to building it from scratch is large, and the economical benefit makes it the preferred approach for the majority of use cases. The paper presents the hardware changes as well as the software stack that constructs the core for an autonomous vehicle, capable of navigating without human intervention as well as providing a base on which to build more complex autonomous behaviours. The vehicle used as an example is a BlueRov2, a low-cost 6 DoF vehicle. However, the software, which is building on top of ROS to enable autonomous capabilities, and parts of the vehicle has additionally been deployed on a Saab Seaeye Falcon** ROVs, demonstrating the generality of the proposed approach. We show a set of in-field experiments for mapping, planning, and interaction such as autonomous inspections, and manipulation, which has been carried out by the platform. The experiments shows the complex tasks the AUV is capable at performing but for a fraction of the price of commercially available hovering capable AUVs.

**https://www.saabseaeye.com/solutions/underwater-vehicles/falcon

REFERENCES

[1] Martin Ludvigsen, Fredrik Søreide, Kurt Aasly, Steinar Ellefmo, Michael Zylstra, and Marshall Pardey. Rov based drilling for deep sea mining exploration. In *OCEANS 2017 - Aberdeen*, pages 1–6, 2017.

[2] D.L. McLean, M.D. Taylor, A. Giraldo Ospina, and J.C. Partridge. An assessment of fish and marine growth associated with an oil and gas platform jacket using an augmented remotely operated vehicle. *Continental Shelf Research*, 179:66–84, 2019.

[3] Marcia J. Isakson, Paul Abkowitz, Michael Rukavina, Zel Hurwitz, and Isaac Metcalf. Measuring seafloor roughness using an rov mounted laser profiling system. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–4, 2016.

[4] David A. Robb, Jonatan Scharff Willners, Nicolas Valeyrie, Francisco J.Chiyah Garcia, Atanas Laskov, Xingkun Liu, Pedro Patron, Helen Hastie, and Yvan R. Petillot. A Natural Language Interface with Relayed Acoustic Communications for Improved Command and Control of AUVs. *AUV 2018 - 2018 IEEE/OES Autonomous Underwater Vehicle Workshop, Proceedings*, 2018.

[5] Jonathan Teague, Michael J. Allen, and Tom B. Scott. The potential of low-cost rov for use in deep-sea mineral, ore prospecting and monitoring. *Ocean Engineering*, 147:333–339, 2018.

[6] Toshihiro Maki, Yoshiki Sato, Takumi Matsuda, Reyes-Tatsuru Shiroku, and Takashi Sakamaki. Auv tri-ton 2: An intelligent platform for detailed survey of hydrothermal vent fields. In *2014 IEEE/OES Autonomous Underwater Vehicles (AUV)*, pages 1–5, 2014.

[7] Y.R. Petillot, S.R. Reed, and J.M. Bell. Real time auv pipeline detection and tracking using side scan sonar and multi-beam echo-sounder. In *OCEANS '02 MTS/IEEE*, volume 1, pages 217–222 vol.1, 2002.

[8] N Valeyrie, F Maurelli, P Patron, J Cartwright, B Davis, and Y Petillot. Nessie v turbo: a new hover and power slide capable torpedo shaped auv for survey, inspection and intervention. In *AUVSI North America 2010 Conference*, pages 8–83, 2010.

[9] David Ribas, Pere Ridao, Lluís Magí, Narcís Palomeras, and Marc Carreras. The Girona 500, a multipurpose autonomous underwater vehicle. *OCEANS 2011 IEEE - Spain*, pages 6–10, 2011.

[10] Marc Carreras, Carles Candela, David Ribas, Narcís Palomeras, Lluís Magí, Angelos Mallios, Eduard Vidal, Èric Vidal, and Pere Ridao. Testing sparus ii auv, an open platform for industrial, scientific and academic applications. *Instrumentation viewpoint*, (18):54–55, 2015.

[11] Narcis Palomeras, Andres El-Fakdi, Marc Carreras, and Pere Ridao. COLA2: A control architecture for AUVs. *IEEE Journal of Oceanic Engineering*, 37(4):695–716, 2012.

[12] G. G. Acosta, B. V. Menna, I. Carlucho, M. D. Paula, S. A. Villar, H. J. Curti, A. F. Rozenfeld, R. J. de la Vega, A. Isasmendi, R. C. Leegstra, and L. M. Arrien. Ictiobot-40 a low cost auv platform for acoustic imaging surveying. In *OCEANS 2019 - Marseille*, pages 1–10, 2019.

[13] Bluerov2 - affordable and capable underwater rov. *URL https://bluerobotics.com/store/rov/bluerov2/*, Nov 2020.

[14] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys. Pixhawk: A system for autonomous flight using onboard computer vision. In *2011 IEEE International Conference on Robotics and Automation*, pages 2992–2997, 2011.

[15] Chenggang Dai, Mingxing Lin, Xiaojian Wu, Zhen Wang, and Zhiguang Guan. Single underwater image restoration by decomposing curves of attenuating color. *Optics & Laser Technology*, 123:105947, 2020.

[16] Gerardo G. Acosta and Sebastián A. Villar. Accumulated ca–cfar process in 2-d for online object detection from sidescan sonar data. *IEEE Journal of Oceanic Engineering*, 40(3):558–569, 2015.

[17] Tomasz Łuczyński, Piotr Łuczyński, Lukas Pehle, Manfred Wirsum, and Andreas Birk. Model based design of a stereo vision system for intelligent deep-sea operations. *Measurement: Journal of the International Measurement Confederation*, 144:298–310, 2019.

[18] suburbanmarine - underwater connectors. *URL https://www.suburbanmarine.io/*, July 2021.

[19] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.

[20] Thomas Moore and Daniel Stouch. A generalized extended Kalman filter implementation for the robot operating system. *Advances in Intelligent Systems and Computing*, 302:335–348, 2016.

[21] Bharat Joshi, Nikolaos Vitzilaios, Ioannis Rekleitis, Sharmin Rahman, Michail Kalaitzakis, Brennan Cain, James Johnson, Marios Xanthidis, Nare Karapetyan, Alan Hernandez, and Alberto Quattrini Li. Experimental Comparison of Open Source Visual-Inertial-Based State Estimation Algorithms in the Underwater Domain. *IEEE International Conference on Intelligent Robots and Systems*, pages 7227–7233, 2019.

[22] Eric Westman, Akshay Hinduja, and Michael Kaess. Feature-based SLAM for imaging sonar with under-constrained landmarks. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3629–3636, 2018.

[23] Maurice F. Fallon, Georgios Papadopoulos, and John J. Leonard. Cooperative AUV navigation using a single surface craft. *Springer Tracts in Advanced Robotics*, 62:331–340, 2010.

[24] Jonatan Scharff Willners, Lachlan Toohey, and Yvan Petillot. Sampling-Based Path Planning for Cooperative Autonomous Maritime Vehicles to Reduce Uncertainty in Range-Only Localization. *IEEE Robotics and Automation Letters*, 4(4):3987–3994, 2019.

[25] Pere Ridao, David Ribas, Emili Hernàndez, and Alex Rusu. USBL/DVL navigation through delayed position fixes. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2344–2349, 2011.

[26] Georgios Salavasidis, Andrea Munafò, Catherine A. Harris, Thomas Prampart, Robert Templeton, Micheal Smart, Daniel T. Roper, Miles Pebody, Stephen D. McPhail, Eric Rogers, and Alexander B. Phillips. Terrain-aided navigation for long-endurance and deep-rated autonomous underwater vehicles. *Journal of Field Robotics*, 36(2):447–474, 2019.

[27] Elizabeth Vargas, Raluca Scona, Jonatan Scharff Willners, Tomasz Luczynski, Yu Cao, Sen Wang, and Yvan R Petillot. Robust Underwater Visual SLAM Fusing Acoustic Sensing. *ICRA2021*, 2021.

[28] Corina Barbalata, Valerio De Carolis, Matthew W. Dunnigan, Yvan Pétillot, and David Lane. An adaptive controller for autonomous underwater vehicles. *IEEE International Conference on Intelligent Robots and Systems*, 2015-December:1658–1663, 2015.

[29] Jonathan D. Gammell, Siddhartha S. Srinivasa, and Timothy D. Barfoot. Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. *IEEE International Conference on Intelligent Robots and Systems*, pages 2997–3004, 2014.

[30] Ioan A. Sucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, dec 2012.

[31] Juan David Hernández, Eduard Vidal, Mark Moll, Narcís Palomeras, Marc Carreras, and Lydia E. Kavraki. Online motion planning for unexplored underwater environments using autonomous underwater vehicles. *Journal of Field Robotics*, 36(2):370–396, 2018.

[32] Èric Pairet, Juan David Hernández, Yvan Petillot, and Morteza Lahijanian. Online mapping and motion planning under uncertainty for safe navigation in unknown environments. *arXiv preprint arXiv:2004.12317*, 2020.

[33] Jonatan Scharff Willners, Daniel Gonzalez-Adell, Juan David Hernández, Èric Pairet, and Yvan Petillot. Online 3-Dimensional Path Planning with Kinematic Constraints in Unknown Environments Using Hybrid A* with Tree Pruning. *Sensors*, 21(4):1152, 2021.

[34] Shida Xu, Tomasz Luczynski, Jonatan Scharff Willners, Ziyang Hong, Kaicheng Zhang, Yvan Petillot, and Sen Wang. Underwater Visual Acoustic SLAM with Online Extrinsic Calibration. *Under review for IEEE IROS 2021*, 2021.

[35] Jonatan Scharff Willners, Yaniel Carreno, Shida Xu, Tomasz Łuczyński, Sean Katagiri, Joshua Roe, Èric Pairet, Yvan Petillot, and Sen Wang. Robust underwater slam using autonomous relocalisation. *IFAC Conference on Control Applications in Marine Systems*, 2021.

[36] Yaniel Carreno, Jonatan Scharff Willners, Yvan R. Petillot, and Ronald P. A. Petrick. Situation-Aware Task Planning for Robust AUV Exploration in Extreme Environments. In *Proceedings of the IJCAI Workshop on Robust and Reliable Autonomy in the Wild*, 2021.

[37] Fletcher Thompson and Damien Guihen. Review of mission planning for autonomous marine vehicle fleets. *Journal of Field Robotics*, 36(2):333–354, 2019.

[38] J Benton, Amanda Jane Coles, and Andrew Coles. Temporal planning with preferences and time-dependent continuous costs. In *ICAPS*, 2012.

[39] Blueprint Lab. Reach alpha 5 (2020). *URL https://blueprintlab.com/products/reach-alpha/*, 2020.

[40] Ignacio Carlucho, Dylan W. Stephens, and Corina Barbalata. An adaptive data-driven controller for underwater manipulators with variable payload. *Applied Ocean Research*, 113:102726, 2021.