# Underwater drone for marine ecosystem applications: An alternative propulsion and control system

Lachlan Paulsen

***Abstract*—New Zealand's marine ecosystems are an immensely precious resource to our country both culturally and economically. However, the size of New Zealand's Economic Exclusion Zone (EEZ) is the 9th largest in the world, making surveying these environments time and resource-intensive. The use of Autonomous Underwater Vehicles (AUV) can be used to supplement manual dive surveys with automated data gathering, however the wash and noise from conventional propeller-based propulsion both pose a threat to sensitive marine ecosystems. This project aimed to explore alternative propulsion methods by developing a prototype ROV that uses a novel, propellor-less propulsion system. The propulsion system developed is a dual-layer water jet network, which has the flow from a central water pump selectively redirected by a network of solenoid valves out of this jet network to produce thrust in different directions. The evaluation was conducted by placing the completed ROV prototype into a storage container filled with water. The ROV is capable of 3 Degrees of Freedom (DoF), but can only achieve translational speeds of 1.7 cm/s and a complete yaw rotation in one minute. The chassis used to house the propulsion system is not hydrodynamic and allows water to leak inside it. Ideally, this ROV design can inform the development of improved versions of this ROV concept.**

***Index Terms*—Article submission, IEEE, IEEEtran, journal, LaTeX, paper, template, typesetting.**

## I. Introduction

### A. Motivation

NEW ZEALAND'S coastal marine environments are a precious natural resource. Due to their unique geographical location and makeup, they are home to 65,000 species, 44% of which aren't found anywhere else in the world [1]. They contribute greatly to the country's economy, with seafood exports in 2022 netting almost NZD$2B [2]. However, our marine biodiversity has several threats, such as fishing, mining, chemical pollution, invasive species, climate change, and ocean acidification [3] [4] [5]. Per the United Nations' sustainability goal 14, these marine resources need to be conserved for future generations [6]. A logistical obstacle to this is the size of our Economic Exclusion Zone, which extends from 12 to 200 nautical miles offshore and spans approximately 430 million ha [7]. Autonomous Underwater Vehicles (AUVs) and Remotely Operated Vehicles (ROVs) can be used to augment manual observation of marine environments, making analyses of large bodies of water more feasible [8] [9].

This project was supervised by Alan Brent, Daniel Burmester

### B. Problem Statement

The need to safeguard New Zealand's coastal waters for future generations necessitates effective monitoring and protection measures. However, conventional propeller propulsion systems risk damaging marine ecosystems in this context. The wash, or swirling jet of water created by propeller rotation, can upheave seabed sediment and uproot marine plants [10]. The noise emitted from propeller propulsion has a direct physical and physiological impact on marine wildlife [11]. Hence, there is a need to explore alternative propulsion technologies for these autonomous or remote craft.

### C. Project Deliverables

This project aims to assess the feasibility of an alternative propulsion system for monitoring coastal marine environments by delivering a prototype Remote Operated Underwater Vehicle (ROV) and a basic manual control system for evaluation. The prototype ROV will be treated as a "proof-of-concept" for the novel propulsion system and will be considered a success if its design can be used to inform future, full-scale prototypes.

### D. Specifications

*ROV prototype specifications*

The ROV must have:

1) A novel propulsion system that
   a) Can propel the ROV through the water at a minimum speed of 0.1 m/s
   b) Can allow the ROV to move in the water with a minimum of four Degrees Of Freedom (DOF). This includes translational motion forward, backward, left, and right as well as rotation along the yaw and pitch axes.
2) A chassis that
   a) Is watertight and provides a safe, dry area for the enclosed electronics and power supply.
   b) Has a re-sealable lid to provide access to contents of the chassis after initial assembly.
   c) Can be partially disassembled to allow the removal of the propulsion system for maintenance.
   d) Is positively buoyant, allowing for the addition of fixed or variable ballast to achieve neutral buoyancy.

   e) Is big enough to contain the propulsion system and control electronics
   f) Has port holes that integrate with the propulsion system and are sealable
3) A power supply that:
   a) Can keep the propulsion system in constant operation for at least one hour.

Fig. 1. Physical implementation of ICHTUS ROV

*Control System Specifications*

The Control system needs

1) to function in a standalone mode, without reliance on external resources such as internet access or pre-existing wireless networks, ensuring its operability in any location.
2) to be operable from a Windows, Mac, or Linux device.
3) to be operable from a minimum range of x m from the ROV.

## II. Background Research

### A. Literature Review

To aid the selection of an approach the development of this prototype could follow, existing underwater submersibles that use propellor-less propulsion were investigated. Through comparison to this specific project's constraints and requirements, promising methods were identified and incorporated into the design of the prototype.

*1) Biomimetic fish ICHTUS (Yang et. Al.):* As an alternative to creating underwater vehicles that have artificial shapes, many have been developed that mimic how aquatic life traverses their marine environment. This craft can blend in with the coastal environment and observe sea creatures undisturbed in their natural habitat. One such craft is the ICHTHUS robotic fish [12]. A photo of this can be seen in Fig. 1. It is comprised of three sections of a fish-shaped chassis actuated by three connecting servo motors. It propels itself through the water via rapid undulation of its body and tail. The design boasts high propulsion efficiency and agile maneuverability in a form that can move through seaweed easily. Unfortunately, because of its design, the drone relies on translational motion to be able to change the direction it is facing. If a camera were mounted on the front of the fish, it would be difficult to quickly track creatures in motion without first traveling in an ark. Another drawback of this method of propulsion is that it is limited to three degrees of freedom (Yaw, elevation, and surge), which does not meet our requirement of four DOFs. The modularity of this design is also noteworthy. The robotic fish was made from parts that were easy to disassemble and replace. This makes repair very efficient, meaning it would be able to be returned to the field quickly after an unforeseen failure.

*2) Omni egg (Fittery et al.):* The Omni-egg is a novel ROV that has a smooth, appendage-free chassis [13]. A photo of this can be seen in Fig. 2. It was designed to operate in underwater areas in cluttered and fragile environments. It

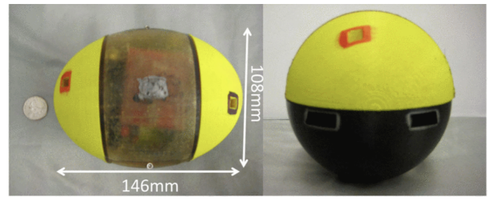


Fig. 2. Physical implementation of Omni egg ROV

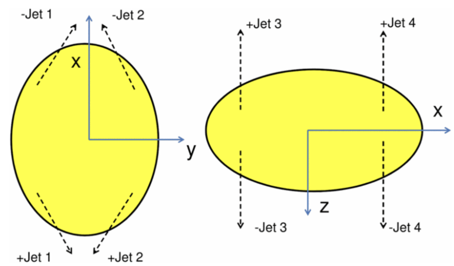


Fig. 3. Physical implementation of Omni egg ROV

features a centrifugal pump/valve network completely incorporated into the inside of the spheroidal chassis. Its unique shape and propulsion configuration grants it the ability to turn at high speeds and maneuver in up to 5 degrees of freedom (surge, sway, heave, yaw, and pitch). The craft was also designed to be neutrally buoyant, meaning that its thrusters are sufficient to cause the craft to dive and surface. The layout of the propulsion is worthy of further study. Fig. 3 shows the layout of the propulsion system which utilizes bi-directional centrifugal pumps and these jets are placed at orthogonal angles to each other. Jets 1 and 2 are at a 30-degree inward angle to give the craft more control of its rotation on the XY plane. These jets are used to stabilize yaw sway dynamics to maintain a heading when traveling in a straight line. 3

A downside of this design is that it was unable to maintain travel along a straight heading without the application of closed-loop feedback control. The open loop control of the prototype resulted in the craft constantly spinning due to the Munk Moment instability. The reproduction of such control is beyond the scope of this project. Another concern with the design is its use of expensive M400s centrifugal pumps produced by TCS micropumps. Per unit, these pumps cost an equivalent of $200 NZD, which is well and truly outside the budget of this project [14]. The general design of the propulsion layout could still be followed with cheaper off-the-shelf pumps.

### B. Tools and Methodologies

The tools used to complete this project include:

1) The VS-code extension Platform, which provides a streamlined and easy-to-use environment for developing C++ Arduino projects.
2) The electronics schematic and PCB layout editor KiCad, which is an all-in-one application for creating PCBs from idea to product.

# III. Design

## A. Design Intro

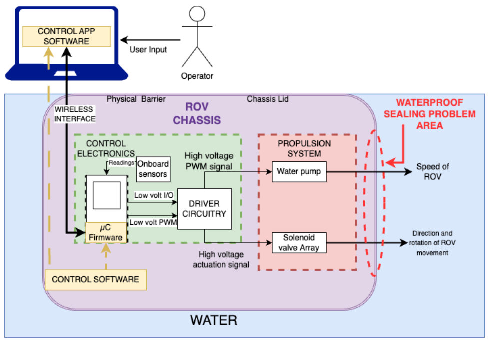


Fig. 4. High-Level architecture of ROV prototype

Fig. 4 shows the high-level design of the technical solution. It can be divided into four key areas: the ROV CHASSIS, the PROPULSION SYSTEM, the CONTROL ELECTRONICS, and the CONTROL SOFTWARE.

The ROV CHASSIS is the physical exterior of the ROV drone. It provides the physical structure of the craft, houses the CONTROL ELECTRONICS and the PROPULSION SYSTEM, and protects them from the underwater environment.

The ROV's PROPULSION SYSTEM is the physical mechanism that generates the thrust force required to propel the ROV through the water and was the central focus of the project. It needed to be designed in such a way that it enabled the ROV to move through the water with a minimum of four Degrees of Freedom (DoF) to fulfill specification, see Fig. 1b. The propulsion system design chosen uses a network of solenoid valves to channel the water output from a central pump to jet nozzles on the exterior of the chassis.

The CONTROL ELECTRONICS is the circuitry that physically controls the propulsion system. It provides the high voltage signals ( 12V) needed to actuate the pump and solenoid nozzles in the propulsion system. They include a microcontroller, which generates electronic signals to actuate the pump and solenoid nozzles to carry out drone movements specified by command instructions received from a control app running on the operator's laptop.

The control electronics also include an array of DRIVER CIRCUITS, because the microcontroller can't create signals with sufficient current or voltage to drive the propulsion system elements directly. These circuits feature a 12V MOSFET switch, which creates or destroys a connection between a propulsion element and a 12V supply line, depending on the signal the microcontroller supplies it with.

The CONTROL SOFTWARE is the programming element of the ROV control system. It is made up of a control app, which runs on the operator's laptop, and the firmware, which runs on the ROV's microcontroller. These two components work together to translate the operator's keyboard input into the desired changes in the ROV's motion.

Since the main objective of the prototype ROV is to assess the viability of the propulsion system, it was decided that the ROV would operate very close to the surface of the water. This meant that the system could use a wireless control interface between the ROV's CONTROL ELECTRONICS and the operator's laptop. The use of a wireless control interface would bridge the physical barrier created by the watertight ROV's chassis. This also avoids the extra electrical, structural, and sealing aspects that would need to be addressed if a tethered connection was used, however, one will likely be necessary if future iterations need to travel deeper in the water since water is inherently opaque to higher-frequency radio communication.

Finally, it is worth noting that the physical ports in the walls of the ROV CHASSIS that the PROPULSION SYSTEM needed to expel water from and take water in from posed a unique challenge for the waterproofing of the enclosure. Especially since the chassis needed to be resealable and the propulsion system needed to be removable from the chassis after the initial assembly.

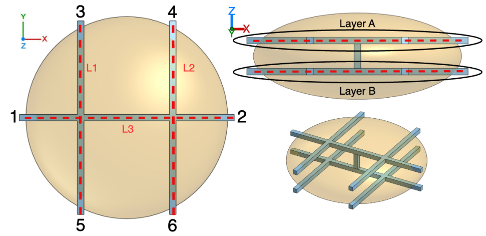


Fig. 5. High-Level architecture of ROV prototype

TABLE I
Summary of ROV motion created by combinations of valves opened

| Valves Open | Resultant motion |
|---|---|
| 5A, 6A, 5B, 6B | Translational movement in +Y direction |
| 3A, 4A, 3B, 4B | Translation movement in the –Y direction |
| 1A, 1B | Translational movement in the +X direction |
| 2A, 2B | Translational movement in the –X direction |
| 3A, 6A, 3B, 6B | Anti-clockwise rotation about the Z axis |
| 4A, 5A, 4B, 5B | Clockwise rotation about the Z axis. |
| 2A, 1B | Anti-Clockwise rotation about the Y axis |
| 1A, 2B | Clockwise rotation about the Y axis |

## B. Propulsion system design

The propulsion system chosen is similar to that used in Fittery et al's Omni Egg ROV. A central water pump takes in ambient water outside the ROV and pumps it into a network of electronic valves. The network of valves selectively directs the flow of water to jet nozzles on the exterior of the ROV chassis. There are 12 jet nozzles on the ROV, arranged in a dual-layer H configuration (Fig. 5). When water is ejected out of one of these nozzles, it imparts a force on the ambient water surrounding the ROV. Due to Newton's third law of motion, this creates a reaction force that pushes against the jet nozzle, propelling the ROV.

How the ROV moves in the water depends on which combination of solenoids is open when the pump is switched on, controlling which jet nozzles the water is allowed to flow through.

When water flows through a jet, the thrust force it creates acts along an imaginary line, or Line of Action (LoA), that

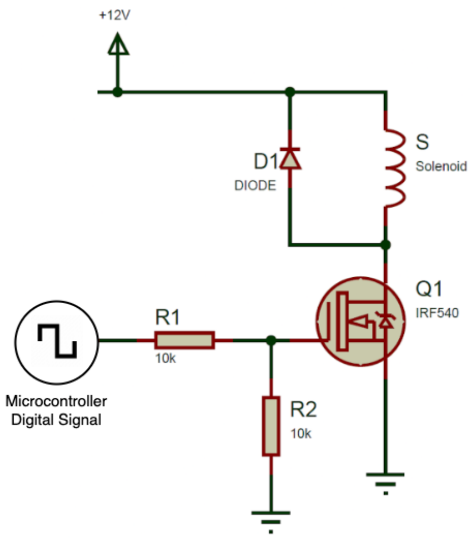


Fig. 6. Topology of one driver circuit

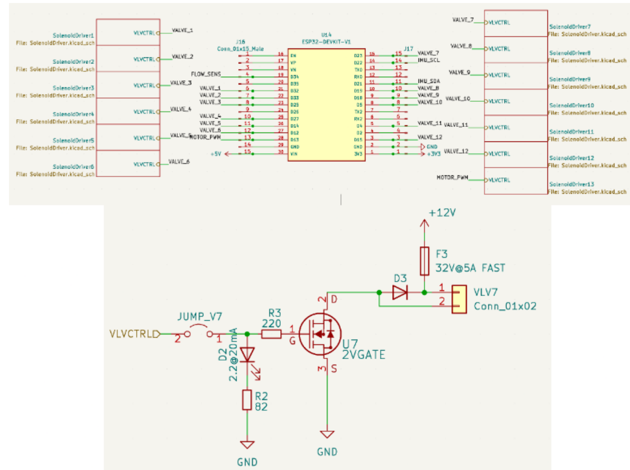


Fig. 7. Top: KiCad schematic showing repeated driver circuits connected to the microcontroller. Bottom: The KiCad schematic of one single driver circuit.

passes through the nozzle. In both layers of the propulsion system, there are three shared LoAs. Nozzles at positions 1 and 2 each generate a force along LoA L3, which goes through the center of the ROV. Meanwhile, the pairs of nozzles at positions 3, 5, and 4, 6 each have their own separate LoA, which run parallel to one another and do not intersect with the center of the ROV.

The consequence of this arrangement is that nozzles at positions 1 and 2 control the translational motion of the ROV along the Y axis, assuming that the same valves at both layers are opened. Meanwhile, nozzles at positions 3, 4, 5, and 6 can cause the ROV to move either translationally along the X-axis or rotate around the Z-axis (yaw), depending on which of these valves are open and if the resulting torque that they create cancels out.

A similar interaction can also occur between the nozzles of the two layers if the pattern of open valves differs between them. Depending on the combination, rotation along the x and y axis (pitch and yaw respectively) is also possible, but the efficiency of this movement depends on the torque produced by the nozzles and the layer separation.

Table I provides a summary of how specific combinations of open valves in the propulsion system result in different motion patterns for the ROV. This table demonstrates that this design enables the ROV to achieve translational movement in both the front/back and left/right directions and can rotate in the yaw, pitch, and roll axes, fulfilling the minimum requirement of four degrees of freedom (Spec. 1b).

At this prototype stage, it is expected that the propulsion system will only be moving along one of these DoFs at a time, which means that only four valves in the propulsion system should be open at a time.

### *C. Propulsion system electronics*

The propulsion system design described in the previous section requires the intelligent operation of 12 electronic valves and one water pump to control the ROV's movement. These electronic elements can't be controlled directly by the human operator of the ROV, as by necessity of its liquid environment, these electronic components must be sealed in the ROV's watertight chassis. Microcontrollers are inexpensive and powerful electronic devices used to control electronic devices according to instructions executed from software programming [15]. Microcontrollers typically feature digital input/output pins, and some microcontrollers also include wireless communication modules, allowing them to interface with remote devices [16]. This made them an ideal component to include in the control electronics.

However, one drawback of relying on microcontrollers is their limitation in supplying substantial current and voltage directly to devices like motors and solenoids. This means that they aren't capable of controlling these components of the propulsion system directly. MOSFETs, or Metal Oxide Semiconductor Field Effect Transistors, are electronic switches that can be actuated by low voltage signals applied to their Gate terminal [17]. MOSFETs can be used to give microcontrollers control over higher voltage circuits that are beyond their capability.

The specific circuit topology chosen for switching the valves and the pump in the propulsion system is shown in Fig. refdriverTop1. The inductive load (Solenoid valve or pump) is connected to a 12V power rail, however, its electrical ground connection is interrupted by the MOSFET. This arrangement is known as a low-side MOSFET configuration. When the driving voltage provided by the microcontroller's digital output goes HIGH, the MOSFET switches ON, completing the high-voltage circuit, and allowing current to be supplied to the load. A flyback diode is also placed in parallel to the inductive load, containing the high-current spike caused by back emf from the load when the high-voltage circuit is opened.

This simple switching block is repeated 13 times, once for each of the twelve solenoid valves and once more for the water pump (See Fig. 7).

To facilitate the evaluation of the system and the development of future projects on the ROV, a flow sensor and an Inertial Measurement Unit (IMU) have been added to the

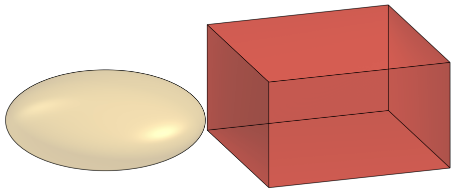

Fig. 8. Top: KiCad schematic showing repeated driver circuits connected to the microcontroller. Bottom: The KiCad schematic of one single driver circuit.

control circuitry.

Finally, all of these electronics require an onboard power supply if the ROV is not going to be tethered.

### D. Chassis design

The design of an ROV chassis that would have met all of the specifications (see Specifications 2) is an undertaking large enough to have been the sole focus of any undergraduate honors project. Since the project's scope also included the creation of the propulsion system, the control circuitry, and the control software, some of these specifications needed to be sacrificed for the chassis to be completed within the scope of the project.

A more hydrodynamic ellipsoid shape and a simple box were two chassis designs that were considered for this project (See Fig. 8). The shape chosen was a box, because it would be straightforward to custom-make it using sheet material, which is a time and cost-economical fabrication method, and could be precisely dimensioned to control the buoyancy of the system and ensure smooth integration with the propulsion system. Unfortunately, this design sacrifices a hydrodynamic shape. The manufacture of an ellipsoidal chassis would have required a more sophisticated manufacturing process to produce, such as 3D printing.

A re-sealable lid needed to be added to the ROV chassis to allow consistent access to the propulsion system and the propulsion electronics inside the chassis. The challenge with this is that it requires a watertight seal that can be broken and sealed repeatedly, so the lid can be easily removed but still prevent the ingress of water when closed. Gaskets are mechanical components made out of elastic materials that are used to seal up microscopic gaps between the mating of two surfaces, in this case between the lid and the rest of the chassis. When they are compressed between two surfaces, they plastically deform and fill any surface irregularities, preventing the ingress of water. A custom gasket can be made by cutting out the shape of the lid from a rubber sheet and placing it in between the lid and the chassis. This can then be compressed by tightening up screws between the lid and the chassis at regular intervals.

### E. Software Design

The software component of the prototype ROV is broken up into two key programs: The control app, which runs on the operator's laptop, and the firmware, which is executed on the ROV's internal microcontroller. Together these programs need to translate the user input from the operator into propulsion system adjustments that cause the desired motion of the ROV. They are also responsible for logging data collected from the ROV's sensors and displaying these readings to the user.

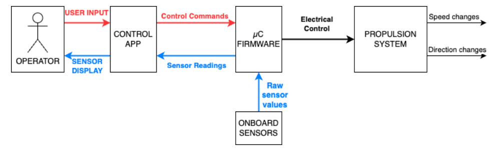


Fig. 9. Flow of data between Operator's laptop and ROV microcontroller. The downstream communication (instructions going from user to ROV) is highlighted in RED and the upstream communication (readings going from ROV to user) is highlighted in BLUE.

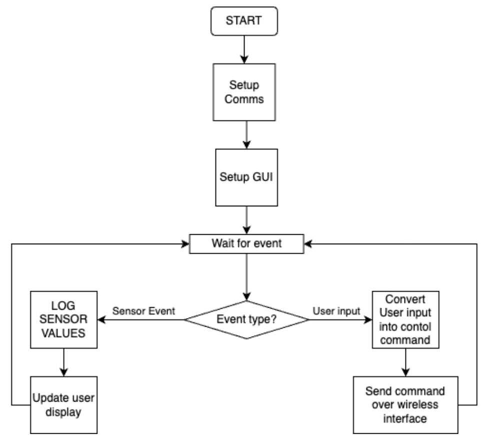


Fig. 10. Behavioural flow diagram of the Control App

As there is no physical tether between the ROV and the operator's laptop in this iteration, these programs need to make use of a wireless interface to communicate this data with each other. The flow of this data is summarised in Fig 9.

Since the system must handle lots of asynchronous inputs, such as user input and sensor readings, an event-driven architecture has been used in the ROV software where possible. Event-driven programming is a software design approach that focuses on the emission of events and the asynchronous handling of them in callback methods [18]. This approach leads to a more transparent and inherently robust software solution.

*1) Control app:* The control app is responsible for collecting keystroke inputs from the operator, translating them into command instructions for the propulsion system, and transmitting them downstream to the microcontroller in the ROV. It is also responsible for receiving sensor events that have been transmitted upstream from the ROV microcontroller, logging them to a file on the operator's laptop, and displaying them to the user.

A Graphical User Interface (GUI) was created to handle the capturing of user input and the display of data to the user. A GUI was a good choice for user interaction when compared to the option of using a Command Line Interface (CLI) instead. A paper [19] investigated how the choice between a GUI or CLI for a program impacted user experience and their

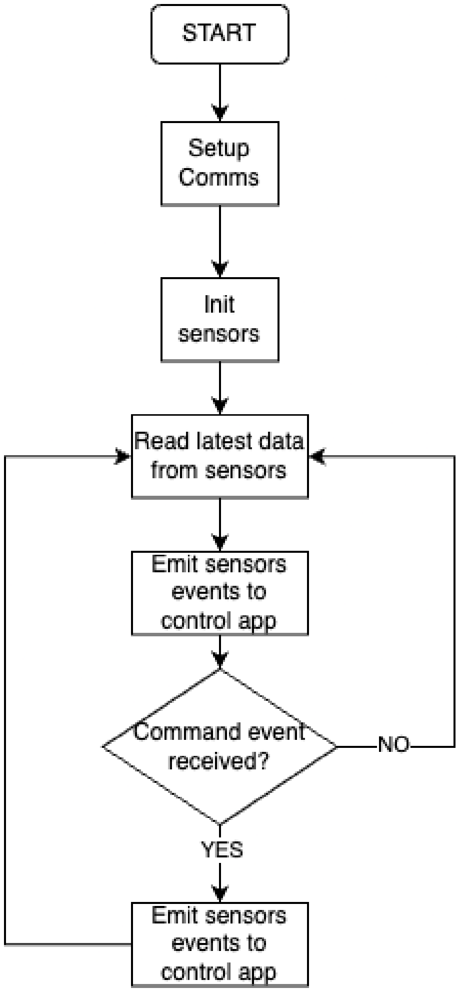


Fig. 11. Behavioural flow diagram of microcontroller firmware.

ability to perform tasks. The results were that CLIs were more confusing and more difficult to use than GUI, while GUIs were more intuitive for participants and easier for them to carry out actions quickly. This made a GUI ideal for a real-time control application for the ROV.

Fig 10 shows a behavioral flow chart of the control app program. After performing the initial setup of the application it waits for a new event. Once it is received, it sends out a new command instruction event to the ROV if it is a user event or logs it to file if it is a new sensor value event.

*2) Microcontroller firmware:* The microcontroller firmware is responsible for receiving command events from the control app and implementing them by making the appropriate low-level adjustments to the propulsion system to cause the desired changes in the ROV's motion. The microcontroller firmware also needs to get the latest readings from the ROV's onboard sensors and transmit them as sensor events upstream to the control app.

Fig 11 shows the behavioral flow chart of the microcontroller firmware. After the initial setup, it periodically scans the IMU and flow sensor for new data to send upstream as sensor events. It breaks from this loop to actuate the propulsion system whenever a command event is received from upstream.

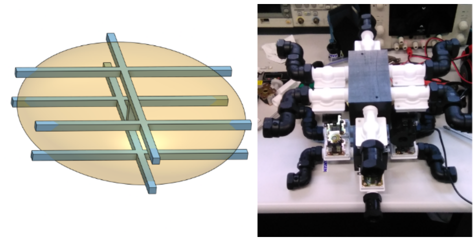

Fig. 12. Side-by-side comparison of the propulsion system concept and physical realization

Fig. 13. Physical implementation of jet nozzles

## IV. Implementation

### A. Propulsion System

A side-by-side comparison of the conceptual design of the propulsion system and its physical implementation is shown in Fig. 12. It features a 12V 19 Watt centrifugal water pump, the flow from which is channeled by a network of 12V Adafruit 997 solenoid valves to 12 custom resealable jet nozzles made from HANSEN TrueFit irrigation fittings and laser-cut acrylic washers. The plumbing of the propulsion system is also constructed from HANSEN fittings and two custom distribution manifolds machined from Acetal plastic. The layout of the propulsion system has been kept as symmetrical as possible to try and keep the ROV's center of gravity as close to its physical center as possible to help maneuverability in the required four DoF (Spec. 1b).

$$F_{Thrust} = \rho_{water} \times q_{pump} \times (\frac{q_{pump}}{A} - v_{ROV}) \qquad (1)$$

$\rho_{water}$ is water density, $q_{pump}$ is the volumetric flow rate of water from the pump, A is the area of the nozzle outlet and $v_{ROV}$ is the current velocity of the ROV in the water

*1) Jet Nozzle design:* Fig. 13 shows the completed jet nozzles used in the propulsion system. They are made from one male-male hex nipple and one female end cap with a hole drilled through its center. They are designed to screw into the propulsion system's output sockets from outside the ROV's chassis. This allows them to be removed from the propulsion system after the ROV has been completely assembled, so they do not prevent the propulsion system from being lifted out of the chassis.

The diameter of the outlet drilled into the jet nozzle had to be determined experimentally. Eq. 1 states that the reduction of outlet diameter (and reduction of outlet area A) increases the thrust force of the water exiting the nozzle. This however comes at the cost of a greater pressure differential required

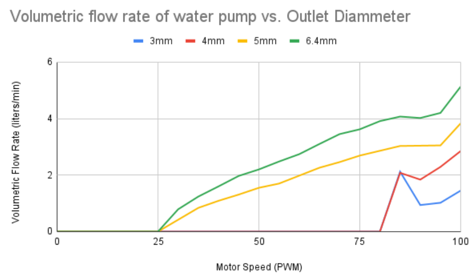


Fig. 14. Volumetric flow rate curves measured for different outlet diameters. 5mm optimizes both flow speed and water exit velocity, which was observed by how far the water stream traveled through open air.

from the propulsion pump. The more pressure that the pump has to work against, the lesser the flow rate it can produce [20], which would start to reduce the thrust generated by the nozzle if the outlet was too small.

It was found that 5mm was the outlet diameter that optimized both the volumetric flow rate of water from the pump and the exit velocity of the water leaving the nozzle (see Fig. 14).

In other words, it was found that an outlet diameter of 5mm would give the ROV the best chance at meeting Spec 1a.

To make the jet nozzles form a watertight seal with the ROV chassis during assembly, washers made out of laser-cut acrylic were made to fit over the hex nipple fitting. These compress a rubber washer over that jet's particular port in the ROV chassis when the jet nozzle is screwed into the propulsion system.

*2) Propulsion system plumbing:* HANSEN True Fittings were used to construct the propulsion system. They were selected because they are made out of durable and corrosion-resistant plastic that will not leech any contaminants into the ocean [21]. These fittings also grant the propulsion plumbing a high level of modularity. Should any of these components break, they are easily replaceable, which will help contribute to the longevity of the system.

*3) Valve selection:* Adafruit 997 valves were selected because their response time is less than 3 tenths of a second and have a very long actuating life of more than 50 million cycles. Fast response times are very important in the response times of remote-operated robots [22]. The long actuating lifespan of the valves means that they would not need to be replaced as often, which helps to reduce electrical waste.

### *B. Control Electronics*

The Control electronics are comprised of 13 identical solenoid driver circuits, made from an IRLU014PBF N-channel MOSFET, a 1N4001 50V diode and 2 1/4W current limiting resistors. One of these circuits is used to drive the water pump, while the rest each drive one of the 12 solenoid valves in the propulsion system. All of these circuits are controlled by an ESP32 microcontroller breakout board plugged into the PCB. The circuitry also has a Fermion BMX160 9-axis Inertial Measurement Unit (IMU) and a 114991172 Brass Flow sensor attached to the inlet of the propulsion pump to facilitate

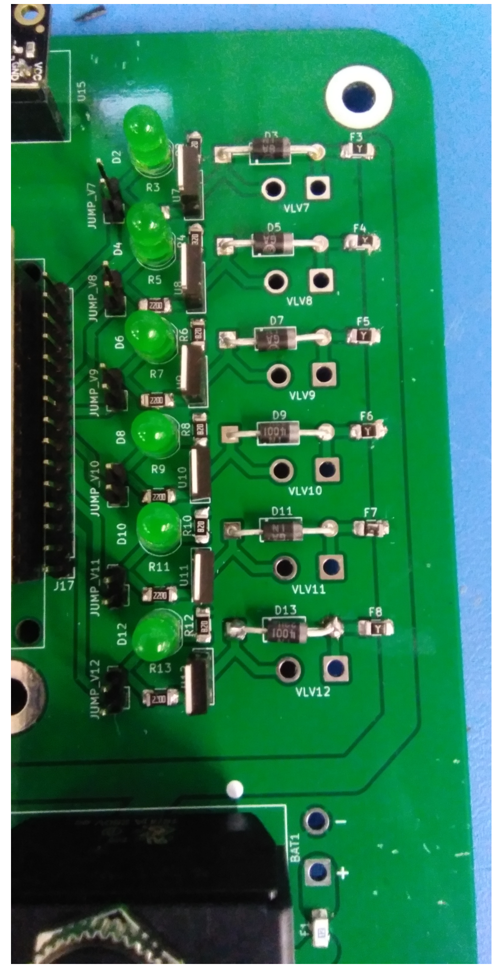

Fig. 15. A column of the completed driver circuit units.

evaluation of the system and to provide basic resources for future projects to develop advanced control algorithms.

Finally, various LEDs have been added to the circuitry to indicate the state of each driver circuit and the presence of a power supply.

*1) Microcontroller:* The ESP32 is a versatile microcontroller developed by Espressif Systems [23]. This microcontroller features a 2.4 GHz WiFi and Bluetooth transceiver, which is used to establish a Wi-Fi access point for the operator's laptop to connect to. This enables the operator to control the ROV without an existing wireless network (Spec. 1) The ESP32 also features 34 GPIO pins and a SCL/SDA interface, which is sufficient to interface with the 13 driver circuits and the onboard sensors.

*2) Driver circuit:* Fig. 15 shows the completed driver subcircuit. The IRLU014PBF MOSFET features a low gate threshold of 2V, which can be directly supplied by the ESP32's 3.3V GPIO pins.

*3) Power supply:* The control electronics are powered by 2 BP2.3-12-T1 12V 2.3 Ah Lead acid batteries (BP2s for short) wired in parallel, with a K7805-2000R3 DC-DC switching converter stepping down the battery voltage to a 5 volt supply for the ESP32 microcontroller.

Before the BP2s were purchased, the current draw of the

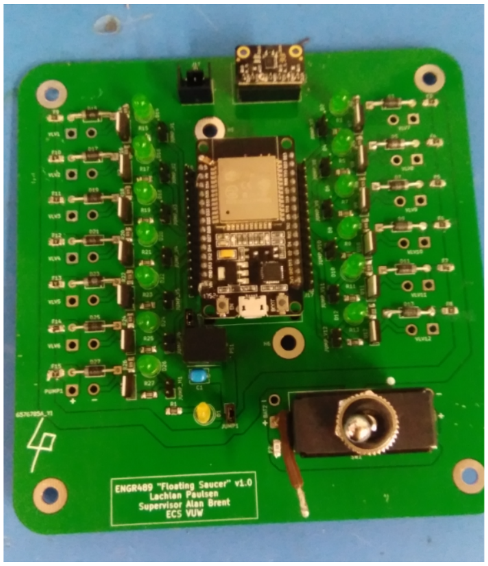


Fig. 16. Final design of the control electronics implemented on a PCB.

TABLE II
CALCULATION OF REQUIRED BATTERY SUPPLY CAPACITY FROM THE CURRENT DRAW OF COMPONENTS

| Component | Current Draw (A) |
|---|---|
| Propulsion Pump | 0.64 |
| Solenoid Valve (Switched on) | 0.5 times 4 = 2 A |
| Microcontroller, other circuitry | Assumed negligible |
| TOTAL CURRENT DRAW | 2.64 A |
| WITH BUFFER ADDED (20%) | 3.16 A |
| CAPACITY REQUIRED (minimum) | 3.16 Ah |

key components of the control electronics was measured to make sure that they would be sufficient to power the ROV for one hour (Spec. 3a) Table II shows that the combined current draw of these elements would be approximately 3.64 A. This meant that the power supply needed to have at least 3. Ah of capacity after a 20% buffer was added for robustness. The two BP2s in parallel combined provide 3.16 Ah of capacity for the control electronics, which met this requirement and were purchased.

The BP2s were also chosen for their slim form factor, which helps to reduce the total volume of the ROV chassis. This helps to reduce drag when the ROV moves through the water and to reduce the amount of ballast required to make the ROV neutrally buoyant. Having separate batteries also means that their weight can be more evenly distributed in the ROV chassis. This helps to keep the ROV's center of gravity close to its physical center, which improves the maneuverability of the ROV in the required four DoF (Spec. 1b) (See section 1b).

Fig 16 Shows the final implementation of the control electronics.

## C. Control software

The final control software of the ROV prototype is made up of both a control app for the operator's laptop and firmware for the ROV's microcontroller, as detailed in section blah blah. The firmware is written in C++ using the Arduino framework while the control app is written in Python using

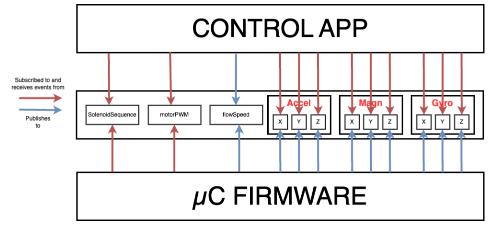


Fig. 17. Final design of the control electronics implemented on a PCB.

the library PyGame. The communication protocol MQTT is used to transport data between the ESP32's firmware and the control app on the operator's laptop. Finally, the ROV's onboard sensor readings that are captured by the control app are logged into individual CSV files, one for each MQTT topic.

*1) Programming language selection:* C++ is an excellent choice for programming microcontrollers due to its efficiency and low-level control capabilities. Microcontrollers often have limited resources, and C++ allows for fine-grained memory management and optimization [24] [25]. Its statically-typed nature also ensures reliability and predictable performance, crucial in embedded systems. On the other hand, Python is well-suited for GUI applications on laptops, thanks to its simplicity and rich library support. Python's clean syntax enables rapid development, making it ideal for creating user-friendly interfaces [26] [25]. Its extensive standard libraries and third-party packages simplify GUI development, making Python a top choice for desktop applications where ease of development and functionality are paramount.

*2) Communication protocol:* MQTT, or (Message Queuing Telemetry Transport) is a lightweight, publish-subscribe messaging protocol designed for efficient communication between devices in IoT and other resource-constrained environments [27]. This communication protocol involves clients publishing and/or subscribing to an individual MQTT broker. When a new message is published to a topic, the clients that have subscribed to it automatically receive message events, containing the new value of the topic.

The C++ library PicoMqtt and the Python library Paho-Mqtt both implement the MQTT communication protocol and automatically handle the reception of message events. They both intelligently handle data communication over the WiFi interface and provide convenient callback methods for the rest of the software. Ordinarily, an additional computer device needs to host the MQTT broker for the clients to be able to communicate. Fortunately, the PicoMqtt library also hosts an MQTT broker on the ESP32 while also providing a client for other software on the microcontroller.

How the control app and the ESP32 firmware communicate is summarised in Fig 17.

*3) GUI + User inputs summary:* Fig 18 shows a screenshot of the Control app's PyGame GUI. The/ WASD keys control the translational motion of the ROV while the UP, DOWN, LEFT and RIGHT keys control yaw and pitch rotation. Th*e Q and Z keys change the Queued Motor pWM and SPACE sneds that queued value to the ROV.

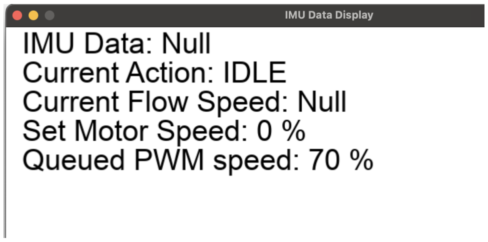


Fig. 18. Control app GUI

Fig. 19. Completed ROV prototype

### D. ROV Chassis

Fig 19 shows the completed ROV chassis. It has been assembled from laser-cut parts of 6mm transparent acrylic and glued together. It features a lid with a resealable compression gasket made from a cutout of nitrile rubber sheet. It has been sealed with SILIFLEX plumbing sealant, to ensure a watertight seal that protects the electronics and propulsion system.

## V. Evaluation

To evaluate ROV buoyancy and motion in the water, it was placed inside a 100L Living and Co storage container [28]. This container measures x cm long, y cm wide and z cm deep.

This is long enough for the ROV to travel 30cm from one end of the container to the other, deep enough for it to be submerged completely and wide enough for it to freely rotate.

### A. Buoyancy (Spec. 2d)

The completed ROV weighs 10kg and displaces approximately 26.75 kg of water when completely submerged. In freshwater, this equates to a net positive buoyancy force of 164N (see table or equation). Saltwater in the ocean is denser, at 1,030 kilograms per cubic meter (As opposed to $1,000 kg/m^3$ for freshwater). This would make the ROV have a net positive buoyancy force of 172 N when placed in salt water. This meant that the ROV had to be held under the water by a human, which did not allow the ROV to move freely.

To solve this, four water bottles filled with a mixture of sand and water were placed inside the ROV for ballast. Each bottle weighed approximately 2kg, making the final mass of the ROV 18kg. The addition of this ballast was enough to partially submerge the ROV so that the top-level nozzles were submerged under the water, which enabled the ROV to move freely for testing.

Unfortunately, since the ballast bottles were a late addition to the design of the ROV, the bottles weren't able to be placed in a perfectly symmetrical arrangement. This shifted the ROV's center of gravity away from its physical center, causing the ROV to tip inside the container. To combat this, more weights had to be placed on top of the ROV's lid and manually shifted until the lean of the ROV wasn't as pronounced so testing could continue.


Fig. 20. Result of watertightness test.

### B. Watertightness. (ROV Spec. 2a)

To test if the ROV chassis was watertight, paper towels were placed inside the chassis, immediately under the lid. The chassis was then fully submerged in the water and held down for one minute. Fig. 20 shows the results of this test. Unfortunately, the corners of the paper towel layer got wet, indicating that the corners of the lid allowed water to leak inside. This was likely caused by the longer separation between the screws about each corner of the lid. Thankfully this was a very slow leak, meaning that a small amount of water would pool in the bottom corners of the chassis. Since the batteries were elevated off of the bottom of the chassis, this meant that testing could continue, although the chassis would eventually completely flood if left underwater for long periods. This would make the current design of the ROV lid unsuitable for deployment in the ocean full-time.

### C. ROV disassembly and reassembly (ROV Spec. 2c, 2e)

The ROV chassis and the propulsion system were designed so that the chassis could be partially disassembled to allow the propulsion system to be lifted out of the top of the chassis. This design worked successfully. To formally evaluate it, the manual disassembly and re-assembly of the chassis from fully sealed to the propulsion system being removed was timed.

TABLE III
ASSEMBLY AND DISASSEMBLY TIMES OF THE ROV PROTOTYPE

| Disassembly | Min:Sec | Reassembly | Min:Sec |
|---|---|---|---|
| START | 0:00 | START | 0:00 |
| LID | 5:39 | PROPULSION | 0:10 |
| NOZZLES | 7:23 | NOZZLES | 2:59 |
| BALLAST | 7:37 | BALLAST | 3:25 |
| PROPULSION | 7:42 | LID | 10:59 |

It took approximately 8 minutes to unscrew all of the jet nozzles, remove the lid and lift out the propulsion system and 11 minutes to assemble it again (See Table III). It should be noted that the majority of this time was consumed by manually screwing and unscrewing the 26 M4 bolts that hold the lid to the chassis. Modifications to the lid design should be considered for the next revision of this prototype.

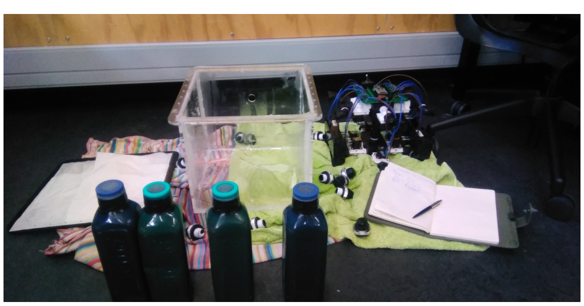

Fig. 21. Dissassembled Chassis with propulsion system removed.

The fully assembled ROV can be seen in Fig. 19 and the disassembled chassis can be seen in Fig. 21.

TABLE IV
EVALUATION OF TRANSLATIONAL SPEED AND ROTATION TIME OF ROV

| Distance: 30cm Run | Forwards Time (s) | Backwards Time (s) | Left Time (s) | Right Time (s) | Complete Yaw rota Time (min:sec) |
|---|---|---|---|---|---|
| 1 | 20 | 12.69 | 20.6 | 21.5 | 1:00.61 |
| 2 | 18 | 19.46 | 16 | 18.78 | 0:59.78 |
| 3 | 15 | 17 | 18.24 | 24.3 | 1:01 |
| 4 | 16 | 15.15 | 17.43 | 20.92 | |
| 5 | 17 | 17.82 | 15.96 | 15.83 | |
| 6 | 15.5 | 14.98 | 17.54 | 15.62 | |
| 7 | 15.27 | 18.87 | 18.6 | 15.82 | |
| Average Time | 16.68 | 16.57 | 17.77 | 18.97 | 1:00.7 |
| Average Speed (cm/s) | 1.8 | 1.81 | 1.69 | 1.58 | |
| Average speed Across all axes (cm/s) | 1.71 | | | | |

### D. DoF and ROV speed (ROV Spec. 1b, 1a)

To assess the motion of the ROV in the water, it was operated inside the storage container filled with water. When the ROV is placed at one end of the storage container, there is a 30cm gap between the wall of the ROV chassis and the opposite end of the container. The time it took for the ROV to travel this 30cm when going forwards/backward and sideways was individually timed in two experiments.

The ROV was able to travel in both of these translational axes, however at speeds slower than the required 0.1 m/s.

Table IV shows the time that it took for the ROV to travel forward, backward, left and then right from one end of the tank to the other. This equates to an overall average speed of 1.7 cm/s for translational motion through the water.

The ROV's ability to rotate in the water was also assessed. It took the ROV approximately 1 minute to complete a full rotation about the Y axis (yaw).

Unfortunately, the ROV was not able to adjust its pitch when in the water.

### E. Control app range (Control system Spec. 3)

The control app's effective operation range of the ROV is dependent on the maximum transmission range of the ESP32 microcontroller's wireless antenna.

The ROV was placed at one end of a long empty hall and the operator's laptop running the control app was taken down the hall until the communication was lost. The laptop was able to be moved approximately 63 meters away from the ROV before the control app lost communication.

## VI. CONCLUSION

This project aimed to create a prototype ROV that utilizes a novel propulsion system to monitor sensitive marine ecosystems. The propulsion system developed uses a dual-layer network of water jet thrusters, powered by a central centrifugal pump. It primarily uses modular, off-the-shelf, fittings and electronic valves which can be easily replaced to ensure a long service life.

Evaluation of the ROV was conducted in a 110L storage tank filled with water. It was found that the ROV was only capable of motion in 3 Degrees of Freedom (DoF) in the translational directions surge and sway as well as yaw rotation. The maximum translational speed of the ROV is 1.7 cm/s and it can complete a full yaw rotation in one minute. A more powerful pump would be required to achieve faster movement speed. A redesign of the ROV chassis will also be necessary to improve its water tightness, hydrodynamics, stability and resistance to pressure.

Apart from these improvements, future projects could also include a buoyancy engine that takes in surrounding water to control the ROV buoyancy and the implementation of feedback control algorithms that use the ROV's onboard sensors to enable precise motion of the craft.

Ideally, this project has produced a first revision of the ROV prototype that can inform the design of improved versions, working up until New Zealand's precious marine ecosystems are maintained by a fleet of underwater drone caretakers.

## Acknowledgments

This should be a simple paragraph before the References to thank those individuals and institutions who have supported your work on this article.

## References

[1] A. MacDiarmid, "The treasures of the sea: Ngataonga a tangaroa," *A summary of the biodiversity in the New Zealand Marine Ecoregion. Wellington: WWF-New Zealand*, vol. 193, 2007.

[2] "New zealand seafood exports, summary report december 2022," tech. rep., Seafood New Zealand, 2022.

[3] Department of Conservation, New Zealand, "Threats facing our oceans." Website. Accessed on 2023-10-11.

[4] A. M. et al., "Assessment of anthropogenic threats to new zealand marine habitats new zealand aquatic environment and biodiversity report," tech. rep., 2012.

[5] C. S. L. et al., "Ocean acidification in new zealand waters: trends and impacts," *New Zealand Journal of Marine and Freshwater Research*, vol. 52, no. 2, pp. 155–195, 2018.

[6] United Nations, "United nations sustainable development goals: Oceans." Website. Accessed on 2023-10-11.

[7] N. Z. M. for the Environment., *Improving regulation of environmental effects in New Zealand's exclusive economic zone : discussion paper.* Ministry for the Environment, 2007.

[8] R. McGeady, R. M. Runya, J. S. Dooley, J. A. Howe, C. J. Fox, A. J. Wheeler, G. Summers, A. Callaway, S. Beck, L. S. Brown, G. Dooly, and C. McGonigle, "A review of new and existing non-extractive techniques for monitoring marine protected areas," 2023.

[9] M. J. Forrest, F. Favoretto, Z. A. Nisa, and O. Aburto-Oropeza, "A deeper dive into the blue economy: the role of the diving sector in conservation and sustainable development goals," *Frontiers in Marine Science*, vol. 10, 9 2023.

[10] T. Fileman, S. de Mora, and T. Vance, *Physical Effects of Ships on the Environment*, pp. 216–224. Cambridge University Press, 1 2021.

[11] M. A. Harbrow, G. Cessford, B. Kazmierow, *et al.*, "The impact of noise on recreationists and wildlife in new zealand's natural areas: a literature review.," *Science For Conservation*, no. 314, 2011.

[12] G.-H. Yang, K.-S. Kim, S.-H. Lee, C. Cho, and Y. Ryuh, "Lnai 8103 - design and control of 3-dof robotic fish 'ichthus v5.5'," 2013.

[13] A. Fittery, A. Mazumdar, M. Lozano, and H. H. Asada, "Omni-egg: A smooth, spheroidal, appendage free underwater robot capable of 5 dof motions," *OCEANS 2012 MTS/IEEE: Harnessing the Power of the Ocean*, 2012.

[14] "M400s-v - tcs micropumps."

[15] R. Keim, "What is a microcontroller? the defining characteristics and architecture of a common component," 2019. Accessed: 21-10-2023.

[16] G. Gridling and B. Weiss, "Introduction to microcontrollers," 2007.

[17] L. Balogh, "Fundamentals of mosfet and igbt gate driver circuits," 2017.

[18] "What is event-driven architecture? — redhat.com." https://www.redhat.com/en/topics/integration/what-is-event-driven-architecture, 2019. [Accessed 23-10-2023].

[19] A. Feizi and C. Y. Wong, "Usability of user interface styles for learning a graphical software application," in *2012 International Conference on Computer & Information Science (ICCIS)*, vol. 2, pp. 1089–1094, IEEE, 2012.

[20] W. Shi, L. Zhou, W. Lu, B. Pei, and T. Lang, "Numerical prediction and performance experiment in a deep-well centrifugal pump with different impeller outlet width," *Chinese Journal of Mechanical Engineering*, vol. 26, no. 1, pp. 46–52, 2013.

[21] HANSEN, "Hansen product catalogue, 2022-2023."

[22] S. A. Moezi, M. Rafeeyan, E. Zakeri, and A. Zare, "Simulation and experimental control of a 3-rpr parallel robot using optimal fuzzy controller and fast on/off solenoid valves based on the pwm wave," *ISA Transactions*, vol. 61, pp. 265–286, 2016.

[23] E. Systems, "Esp32 product homepage." https://www.espressif.com/en/products/socs/esp32. [Accessed 24-10-2023].

[24] Wikipedia contributors, "C++ — Wikipedia, the free encyclopedia." https://en.wikipedia.org/w/index.php?title=C%2B%2B&oldid=1181156798, 2023. [Online; accessed 24-October-2023].

[25] Jim Anderson, "Python vs c++: Selecting the right tool for the job." https://realpython.com/python-vs-cpp/, 2019. [Online; accessed 24-October-2023].

[26] Wikipedia contributors, "Python (programming language) — Wikipedia, the free encyclopedia." https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&oldid=1180741065, 2023. [Online; accessed 24-October-2023].

[27] Mqtt.org, "Mqtt homepage." https://mqtt.org/. [Online; accessed 24-October-2023].

[28] "Living and co rolling organiser 110l — the warehouse." https://www.thewarehouse.co.nz/p/living-co-rolling-organiser-110l/R2789485.html. [Accessed 25-10-2023].