

Thruster Allocation for Underactuated ROVs: Simulation and Real-World Implementation

Amanda Ramos Lima do Nascimento^{1*}, Lucas Marins Batista²

¹ SENAI CIMATEC, Centro de Competências em Robótica e Sistemas Autônomos, Salvador, Bahia, Brasil

² SENAI CIMATEC, Centro de Competências em Robótica e Sistemas Autônomos, Salvador, Bahia, Brasil

*SENAI CIMATEC; Av. Orlando Gomes, 1845 - Piatã, Salvador; amanda.lima@fbter.org.br; lucas.batista@feb.org.br

Abstract: Underwater exploration gained prominence starting in the 1950s with the emergence of the first ROVs (Remotely Operated Vehicles). Among their categories, observation ROVs stand out for their ability to collect data in hard-to-reach environments. In this context, the OpenROV — an open-source mini ROV with a modular architecture — was studied. One of the units had mechanical damage and serial communication failures with the original electronics, requiring a retrofit process. Additionally, a thruster allocation study was conducted to evaluate the platform's movement in both simulation and real operation. Through this technique, it was possible to verify that the current configuration allows movements along the surge, heave, and yaw axes, with correspondence between simulation and actual actuation. The study contributed to the analysis of the system's degrees of freedom and its potential navigability in an aquatic environment.

Keywords: OpenROV; Thruster allocation; Simulation; Underwater vehicles; Retrofit

1. INTRODUCTION

Ocean exploration has driven technological advances since the 20th century. In 1953, Dimitri Rebikoff developed the POODLE, considered the first Remotely Operated Vehicle (ROV) in history, designed for underwater archaeological research [1]. Despite being pioneering, its impact was limited. In 1961, the United States Navy developed a mobile underwater system that culminated in the CURV (Cable-controlled Underwater Research Vehicle), a landmark for the evolution of ROVs [1]. These vehicles have been widely used in various applications ever since. ROVs are commonly classified by their functionality. Observation ROVs are built for data and image collection. Work-class ROVs are equipped with manipulators for interventions. Special-use ROVs are designed for specific missions [1]. ROVs are particularly valuable in scientific research, as they enable data collection in inhospitable and hard-to-reach environments.

Over time, advances in technology enabled the creation of smaller, more accessible solutions. Based on this evolution, platforms like OpenROV emerged [2]. Two OpenRovs models 2.8 [3] in the Robotics and Autonomous Systems Laboratory are used for educational and research purposes. However, one of the units was damaged and a system retrofit process was proposed. Then, an investigation into thruster allocation was conducted, a technique to properly distribute forces among thrusters, enabling the vehicle to perform the desired movements [4]. Therefore, the implementation of thruster allocation has become a promising line of research as the vehicle is undergoing a redesign due to its damaged structure. These vehicles can be classified according to the relationship between the number of actuators and their degrees of freedom. When the number of actuators is fewer than the six typical degrees of freedom of these vehicles, the system is considered underactuated [4]. This group includes the OpenROV, which has only three



thrusters and only moves in the surge, heave, and yaw directions. The relevance of this classification for the study of thruster allocation in ROVs lies in the fact that designing a control system capable of stabilization, trajectory tracking, and trajectory control in underactuated vehicles is a non-trivial task [4]. Such systems require handling physical and dynamic constraints and, in many cases, the adoption of nonlinear control strategies. Thruster allocation, in turn, provides an approach that maximizes the potential of the available actuators, allowing the vehicle to operate efficiently within a reduced set of target movements. This makes it feasible to design simpler and more effective control strategies that take into account the physical limitations of the system. Therefore, the goal of this work is to evaluate the technique's functionality in the OpenROV's motion control system by implementing it on both a physical prototype and in a simulated environment.

2. METHODOLOGY

In this study, it has been necessary to create a simulation environment using the Gazebo Ignition Fortress platform [5] and its buoyancy plugins, aiming to simulate the effect of water on the robot. To make the environment more realistic, the inertia matrix (1) of the OpenROV 3D model, extracted from the Fusion 360 modeling software, was used:

$$\mathbf{I} = \begin{bmatrix} 77.42 & 0 & 0 \\ 0 & 77.42 & 0 \\ 0 & 0 & 77.42 \end{bmatrix} \text{ kg} \cdot \text{m}^2 \quad (1)$$

A mass of 8 kg was estimated for the simulated model, including the thrusters and complete electronics. The vehicle's weight must be equal to the buoyant force generated by (2):

$$F_b = \rho \cdot V \cdot g \quad (2)$$

By doing this, the vehicle achieves neutral buoyancy, which keeps it stable from sinking or floating [6]. The value of ρ for water density was set to 1,000 kg/m³, the standard for water; V is the volume of the ROV, and g is gravity. A native plugin was used to implement these calculations based on the provided data [7]. Besides Gazebo, the ROS2 Humble Hawksbill software was used, specifically the "ros2_control" package suite [8]. This module is responsible for managing controllers and hardware interfaces and integrating them with both Gazebo and the physical robot. The use of these resources enabled the application of the thruster allocation matrix. The matrix calculation process was carried out considering the linear thruster model described by equation (3) found in [4]:

$$F = Ku \quad (3)$$

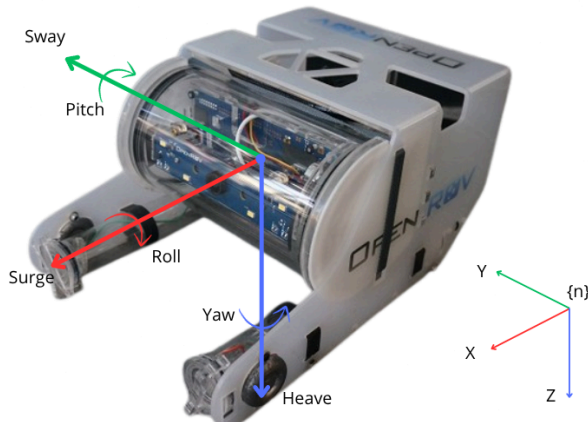
Where u is a matrix containing the control signals sent to the thrusters, and K is the diagonal matrix containing the force coefficients

of each thruster. To obtain the allocation matrix, formula (4) is used:

$$\tau = T(\alpha)F = T(\alpha)Ku \quad (4)$$

Where τ is the vector of generalized forces and moments, and $T(\alpha)$ is the thruster configuration matrix dependent on the steering angles α , according to the NED (North-East-Down) reference frame, standard for underwater vehicles, as shown in Figure 1:

Figura 1. - Coordinate system in NED's pattern



The vector τ is given by (5):

$$\tau = \begin{bmatrix} f \\ r \times f \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ F_z l_y - F_y l_z \\ F_x l_z - F_z l_x \\ F_y l_x - F_x l_y \end{bmatrix} \quad (5)$$

To use equation (5), it is necessary to measure the distances of the thrusters relative to the robot's center of mass. The measured values are shown in the table below:

Table 1. Moment arms of the 3 thrusters of the OpenROV relative to the center of mass

T_i	l_{xi} (mm)	l_{yi} (mm)	l_{zi} (mm)
$T1$	-26	45	0
$T2$	-2	0	26
$T3$	-26	-45	0

By applying (5) for each thruster, the following matrix is obtained, where each column represents a thruster:

$$T(\alpha) = \begin{bmatrix} -1.0 & 0.0 & -1.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ -0.045 & 0.0 & 0.045 \end{bmatrix} \quad (6)$$

The implementation of the allocation is given by formula (7):

$$u = K^{-1} T^{-1} \tau \quad (7)$$

Given a control signal, a vector of corresponding values is obtained to be applied to each thruster. Since the OpenROV matrix is non-square, the Moore-Penrose method must be used to compute the pseudo-inverse T^+ :

$$T^+ = T^T (TT^T)^{-1} \quad (8)$$

$$u = K^{-1} T^+ \tau \quad (9)$$

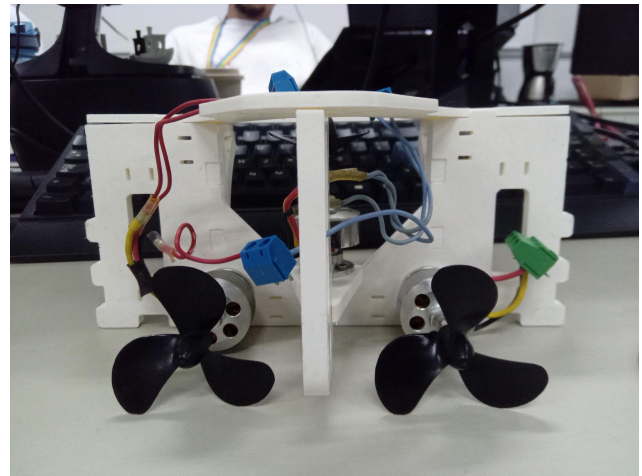
These computations were automated using a controller to handle the specified calculations. An OpenROV platform retrofit was initiated to verify the elements that were described outside of the simulated environment. This procedure was required because of the platform's damage. Figure 2 illustrates how the thrusters' supporting framework was broken.

Figure 2. The acrylic structure supporting the thrusters was damaged.



The CAD files available on the OpenROV GitHub platform [9] were used, and all parts were 3D printed in ABS material, as shown in Figure 3.

Figure 3. Thruster support structure printed in ABS.



Therefore, because of the mechanical capabilities and water resistance, the ABS polymer was selected for submerged applications [10]. Additionally, serial communication with the original electronic board for thruster control was not possible. This limitation also required the development of a new board with new electronic components. For testing, PWM signals were sent via serial commands to activate the motors using an Arduino Nano, a breadboard, and a bench power supply that provided the 12 V required to power the motor ESCs. The experiments with the robot were not conducted underwater.

3. RESULTS AND DISCUSSION

The results of the allocation calculations aimed to enable movements in surge, heave, and yaw. It is worth noting that, initially, tests to verify this were conducted in the Gazebo Ignition simulation environment. Table 2 shows the



values applied to the thrusters, derived from the previous calculations.

Table 2. Force values in Newtons generated by the thrusters activated with a control signal value of 1.

T_i	Surge	Sway	Heave	Roll	Pitch	Yaw
P1	-0.5	0	0	0	0	-0.5
P2	0	0	1	0	0	0
P3	-0.5	0	0	0	0	0.5

The absence of pictures displaying a top perspective and the precise location of the thrusters in relation to the NED coordinate system led to the creation of Figure 4. A rear perspective of the thrusters' position is shown in Figure 5 more clearly:

Figure 4 – Top view of the OpenROV.

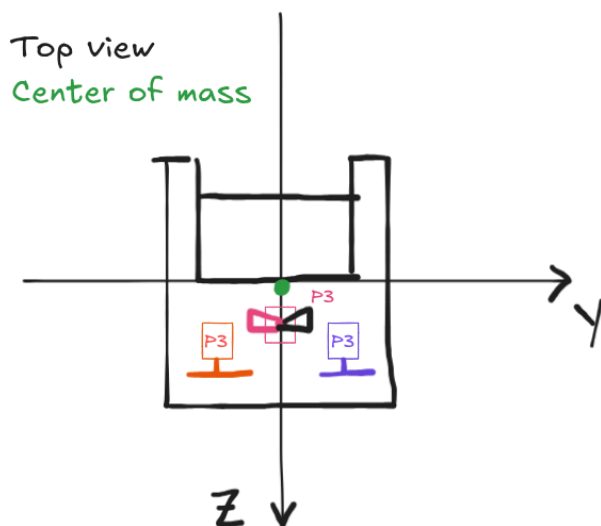
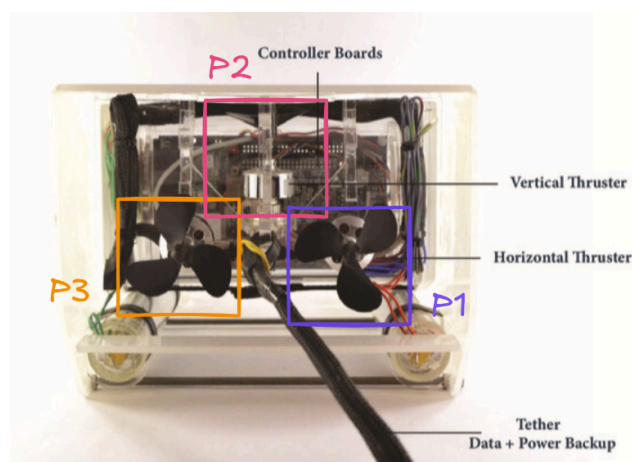


Figure 5. - Rear view of the OpenROV.



Fonte: NADDAF-SH; et al. 2018

The graphs in Figures 6, 7, and 8 provide a clearer visual representation of the numbers found in Table 2.

Figure 6. Control signal in X (surge) vs. right and left thrusters being activated with the same force

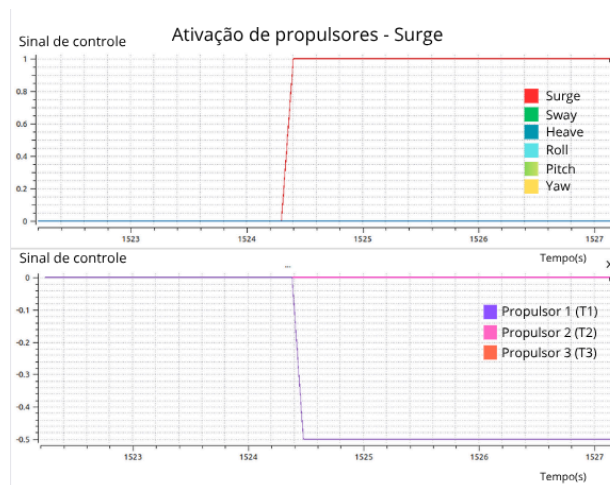


Figure 6 illustrates how the right and left thrusters engage with forces of the same magnitude, direction, and sensation when the ROV is instructed to travel in the direction of the surge. The ROV will move forward thanks to the combined force of both.



Figure 7. Control signal in the heave vs. vertical thruster being activated.

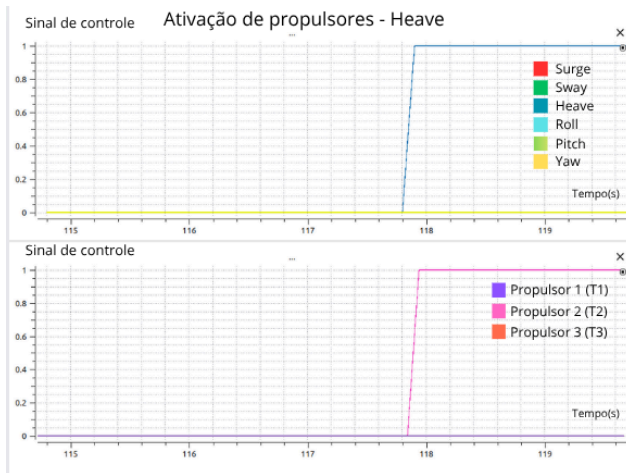
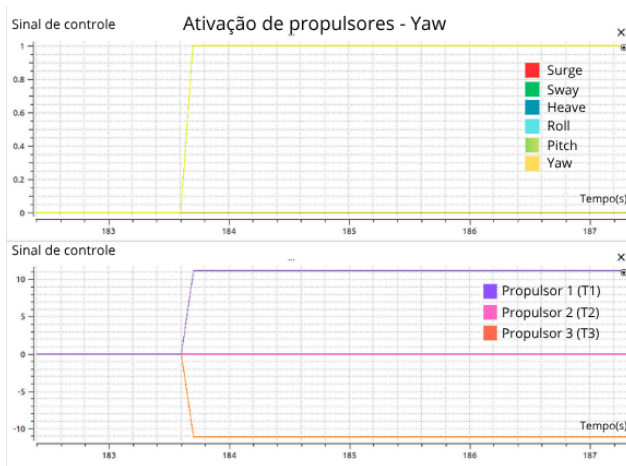


Figure 7 illustrates the activation of the vertical thruster while the others remained inactive in response to an instruction in the heave direction. The upward and downward motions of the ROV are generated by the thruster forces.

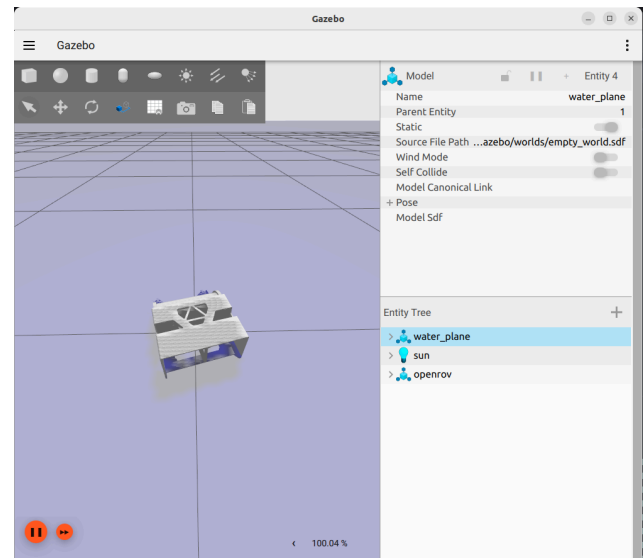
Figure 8. Control signal in yaw vs. right and left thrusters being activated with equal magnitude but opposite forces.



The yaw movement is presented in Figure 8, where the left and right thrusters were activated with forces of equal magnitude but in opposite

directions. The resulting force from this combination is responsible for rotating the ROV around its axis, as illustrated in Figure 9.

Figure 9. OpenROV in simulation performing yaw to the left.



Following the simulation testing, the allocation matrix was implemented on the actual hardware, and the same procedure for gathering data was followed, as seen in Figure 10.

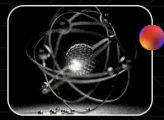
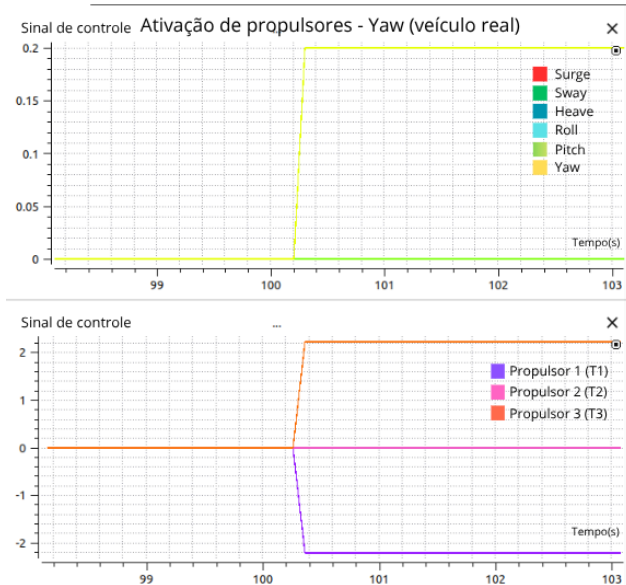


Figure 10. Control signal in yaw vs. right and left thrusters being activated with equal magnitude but opposite forces on the real vehicle.



The yaw movement presented the same response in the simulation, Figure 10. This outcome was confirmed in every other movement, proving the efficacy of both the method, with the real vehicle hardware and the simulation environment.

4. CONCLUSION

It was possible to determine the OpenROV controllable degrees of freedom by comparing and implementing thruster allocation in both simulated and real-world settings. This confirmed that the OpenROV could move along the yaw, heave and surge axes even with fewer thrusters. The concordance between simulation and real-world tests showed the efficacy of the thruster operation technique.

These results provide a foundation for future improvements, such as submerged ROV testing and control of the current axes using a PID controller for robot stabilization, as well as the integration of inertial measurement units (IMUs) to carry out other research activities.

5. REFERENCES

- [1] CHRIST, R. D.; WERNLI, R. L. The ROV Manual. [s.l.] Elsevier, 2011.
- [2] MUNZ, J. Design and Implementation of Software for the ROV Neptunus. [s.l.: s.n.]. Available in: <https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2364892/13963_FULLTEXT.pdf>. Acesso em: 6 ago. 2025.
- [3] OpenROV Dozuk, Wayback Machine. Available in: <https://web.archive.org/web/*/https://openrov.dozuki.com/*/>.
- [4] FOSSEN, T. I. Handbook of Marine Craft Hydrodynamics and Motion Control. [s.l.] John Wiley & Sons, 2021.
- [5] Ignition Fortress — Gazebo fortress documentation. Available in: <<https://gazebosim.org/docs/fortress/install/>>.
- [6] MARZBANRAD, A. et al. Design, Construction and Control of a Remotely Operated Vehicle (ROV). Volume 7: Dynamic Systems and Control; Mechatronics and Intelligent Machines, Parts A and B, 1 jan. 2011.
- [7] Gazebo Sim.: Available in: <https://gazebosim.org/api/sim/8/theory_buoyancy.html>.
- [8] Getting Started — ROS2_Control: Humble Mar 2024 documentation. Available in: <https://control.ros.org/humble/doc/getting_started/getting_started.html>.
- [9] OPENROV. GitHub - OpenROV/openrov-hardware: Hardware for OpenROV. Include lasercut parts and other diagrams. Available in: <<https://github.com/OpenROV/openrov-hardware>>. Acesso em: 8 ago. 2025.
- [10] LEITE, M. et al. Mechanical properties and water absorption of surface modified ABS 3D printed by fused deposition modelling. Rapid Prototyping Journal, v. 24, n. 1, p. 195–203, 2 jan. 2018.
- [11] NADDAF-SH, M-MAHDI.; MYLER, H.; ZARGARZADEH, H. Design and Implementation of an Assistive Real-Time Red Lionfish Detection System for AUV/ROVs. Complexity, v. 2018, p. 1–10, 11 nov. 2018.