IOP Conference Series:
Earth and
Environmental Science

PAPER • OPEN ACCESS

LQR-Based Tracking Control for Remotely Operated Vehicles: A Co-Simulation Framework for Sustainable Environmental Monitoring

To cite this article: Hoai Thanh Nguyen *et al* 2025 *IOP Conf. Ser.: Earth Environ. Sci.* **1565** 012025

LQR-Based Tracking Control for Remotely Operated Vehicles: A Co-Simulation Framework for Sustainable Environmental Monitoring

Hoai Thanh Nguyen[1], Van Qui Nguyen[1], Hoang Quan Vo[2], Van Thuyen Ngo[1], Duc Thien Tran[1]*

[1] Automatic Control Department, Ho Chi Minh City University of Technology and Education, Ho Chi Minh City, 700000, Vietnam
[2] Faculty of Vehicle and Energy Engineering, Ho Chi Minh City University of Technology and Education, Ho Chi Minh City, 700000, Vietnam

*E-mail: thientd@hcmute.edu.vn

Abstract. This paper presents a tracking controller for a remotely operated underwater vehicle (ROV). The kinematic and dynamic models of the BlueROV2 are introduced. The proposed control strategy is based on linear quadratic regulator (LQR) tracking, designed using the dynamic model of BlueROV2. Additionally, a co-simulation environment integrating MATLAB and Gazebo is developed to evaluate the effectiveness of the control system. Simulation results demonstrate that the proposed method enhances the stability and tracking performance of the ROV. Furthermore, a hardware-in-the-loop simulation (HILS) platform, incorporating an external joystick and the ROV within the co-simulation environment, is implemented to assess teleoperation controllability. The results confirm that the LQR tracking controller improves stability and responsiveness under complex underwater conditions.

Keywords: Co-simulation; HILS; LQR Tracking; Marine environmental monitoring; ROV

1. Introduction

A remotely operated vehicle (ROV) is an unmanned underwater system connected to a surface ship by cables, enabling remote navigation and control. ROVs are increasingly vital in marine exploration, infrastructure inspection, and scientific research, with the global market reaching $4.49 billion in 2022 [1].

Control of underwater vehicles is difficult due to waves, currents, and varying pressures. While adaptive sliding mode, fuzzy logic, and PID controllers have been investigated [2–7], most works lack consideration of environmental applications. Recent advances highlight the need for sustainable approaches, including adaptive monitoring of coastal ecosystems [8], environmentally adaptable marine robots [9], and AI-driven ocean observing frameworks [10]. These studies emphasize aligning ROV control research with environmental monitoring and sustainable development.

This study proposes an LQR Tracking controller for BlueROV2 to improve stability in dynamic marine conditions, supporting tasks such as environmental monitoring and sustainable underwater inspections. The method is validated through mathematical modeling, Gazebo/MATLAB co-simulation, and Hardware-In-the-Loop Simulation (HILS). While LQR Tracking has advantages in optimizing control, it remains underexplored for ROVs, especially in environmental missions. By combining high-fidelity simulation and HILS, this research offers a practical step toward reliable real-world deployment.

The remainder of this paper is structured as follows: Section 2 presents the mathematical modeling of the system, forming the foundation for controller design. Section 3 introduces LQR Tracking for heading and depth control, and Section 4 details the co-simulation environment integrating MATLAB, Gazebo, and HILS. Section 5 applies the method to inspect Tien Sa Pier at Da Nang Port. Finally, Section 6 concludes the study and outlines future development directions.

2. Mathematical Modeling

2.1 Dynamics model

The underwater vehicle can be modeled as a rigid body with six degrees of freedom, and its translational and rotational equations can be formulated using the laws of Newton. For analyzing the motion and physical behavior of the ROV, it is essential to define two coordinate systems: the NED frame (n-frame) is an Earth-fixed frame, with the x, y, and z axes pointing towards north, east, and downwards perpendicular to the surface of the Earth, respectively. Its origin can be set at any arbitrary location. The BODY frame (b-frame) is fixed to the ROV, with its origin typically placed at the geometric center of the vehicle.

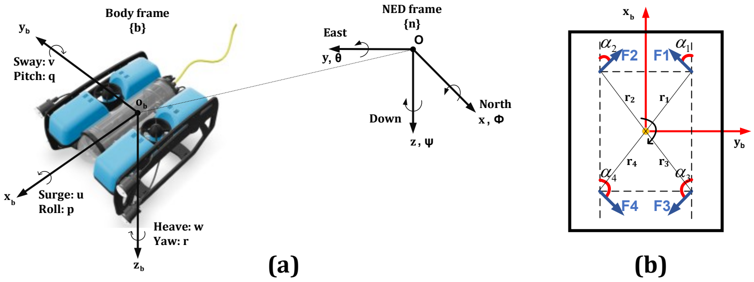


Figure 1. Frame coordinate system and lateral thruster distribution of BlueROV2. **(a)** Frame coordinate system, **(b)** lateral thruster distribution.

For each degree of freedom, the motion of an underwater robot, defined by a variable in both coordinate systems, can be represented using the notation established by the Society of Naval Architects and Marine Engineers (SNAME), as shown in Table 1.

Table 1. The notations for marine vehicles according to SNAME (1950).

DoF	Movement Notation	Forces & moments (b-frame)	Linear & angular velocities (b-frame)	Position & Euler angles (n-frame)
1	Motions in x direction (surge)	X	u	x
2	Motions in y direction (sway)	Y	v	y
3	Motions in z direction (heave)	Z	w	z
4	Rotation about the x axis (roll, heel)	K	p	ϕ
5	Rotation about the y axis (pitch, trim)	M	q	θ
6	Rotation about the z axis (yaw)	N	r	ψ

The following vectors are denoted for the convenience of system model analysis:

$$\boldsymbol{\eta}=\begin{bmatrix}\boldsymbol{\eta}_1^{\mathrm{T}} & \boldsymbol{\eta}_2^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}};\ \boldsymbol{\eta}_1=[x\quad y\quad z]^{\mathrm{T}};\ \boldsymbol{\eta}_2=[\phi\quad \theta\quad \psi]^{\mathrm{T}};\ \mathbf{v}=\begin{bmatrix}\mathbf{v}_1^{\mathrm{T}} & \mathbf{v}_2^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}};\mathbf{v}_1=[u\quad \upsilon\quad w]^{\mathrm{T}}$$
$$\mathbf{v}_2=[p\quad q\quad r]^{\mathrm{T}};\ \boldsymbol{\tau}=\begin{bmatrix}\boldsymbol{\tau}_1^{\mathrm{T}} & \boldsymbol{\tau}_2^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}};\ \boldsymbol{\tau}_1=[X\quad Y\quad Z]^{\mathrm{T}};\ \boldsymbol{\tau}_2=[K\quad M\quad N]^{\mathrm{T}} \tag{1}$$

The 6-DoF kinematic equations of motion for the ROV are represented in two coordinate frames [11]:

$$\dot{\boldsymbol{\eta}}=\mathbf{J}(\boldsymbol{\eta})\mathbf{v} \tag{2}$$

where $\mathbf{J}(\boldsymbol{\eta})\in\mathbb{R}^{6\times 6}$ is the transformation matrix from the {b} frame to the {n} frame.

The transformation for linear velocities from {b} to {n} is given by:

$$\dot{\boldsymbol{\eta}}_1=\mathbf{J}_1(\boldsymbol{\eta}_2)\mathbf{v}_1 \tag{3}$$

where $\mathbf{J}_1(\boldsymbol{\eta}_2)$ is the rotation matrix from {b} to {n} and computed as:

$$\mathbf{J}_1(\boldsymbol{\eta}_2)=\mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi) \tag{4}$$

$$\mathbf{J}_1(\boldsymbol{\eta}_2)=\begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi+c_\psi s_\theta s_\phi & s_\psi s_\phi+c_\psi c_\phi s_\theta \\ s_\psi c_\theta & c_\psi c_\phi+s_\psi s_\theta s_\phi & -c_\theta s_\phi+s_\theta s_\psi c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \tag{5}$$

where $s_a \triangleq \sin(a)$, $c_a \triangleq \cos(a)$, and $t_a \triangleq \tan(a)$ with (a) representing the angles of rotation.

Similarly, the transformation of angular velocities is given by:

$$\dot{\boldsymbol{\eta}}_2=\mathbf{J}_2(\boldsymbol{\eta}_2)\mathbf{v}_2 \tag{6}$$

where $\mathbf{J}_2(\boldsymbol{\eta}_2)$ is the angular transformation matrix from {b} to {n} and derived as:

$$\mathbf{J}_2(\boldsymbol{\eta}_2)=\begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi/c_\theta & c_\phi/c_\theta \end{bmatrix},\theta\neq\pm\frac{\pi}{2} \tag{7}$$

As a consequence, the 6-DoF kinematic equation (2). can be represented in a vector setting by:

$$\begin{bmatrix}\dot{\boldsymbol{\eta}}_1 \\ \dot{\boldsymbol{\eta}}_2\end{bmatrix}=\begin{bmatrix}\mathbf{J}_1(\boldsymbol{\eta}_2) & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{J}_2(\boldsymbol{\eta}_2)\end{bmatrix}\begin{bmatrix}\mathbf{v}_1 \\ \mathbf{v}_2\end{bmatrix} \tag{8}$$

In a general form, the nonlinear dynamic equations of motion in 6-DOF can be written as follows [11]:

$$\mathbf{M}\dot{\mathbf{v}}+\mathbf{C}(\mathbf{v})\mathbf{v}+\mathbf{D}(\mathbf{v})\mathbf{v}+\mathbf{g}(\boldsymbol{\eta})=\boldsymbol{\tau} \tag{9}$$

where the system inertia matrix $\mathbf{M}\in\mathbb{R}^{6\times 6}$, coriolis and centripetal matrix $\mathbf{C}(\mathbf{v})\in\mathbb{R}^{6\times 6}$, hydrodynamic damping matrix $\mathbf{D}(\mathbf{v})\in\mathbb{R}^{6\times 6}$, restoring forces vector $\mathbf{g}(\boldsymbol{\eta})\in\mathbb{R}^{6x1}$, forces and moments acting on the ROV $\boldsymbol{\tau}\in\mathbb{R}^{6x1}$.

3. Design of a tracking controller

3.1 Subsystems of the ROV model

ROV heading and depth stability are crucial for underwater surveying and equipment manipulation. This study focuses on implementing a control system to maintain them during underwater disturbances.

According to Fossen [11], the linear equations of motion for heading and depth control can be expressed in a compact form as follows: Heading equations of motion are presented in equation (10), and diving equations of motion are presented in equation (11).

$$\begin{bmatrix} m-Y_{\dot{\upsilon}} & mx_g-Y_{\dot{r}} & 0 \\ mx_g-N_{\dot{\upsilon}} & I_z-N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \dot{\upsilon} \\ \dot{r} \\ \dot{\psi} \end{bmatrix}+\begin{bmatrix} -Y_{\upsilon} & mu_0-Y_r & 0 \\ -N_{\upsilon} & mx_g u_0-N_r & 0 \\ 0 & -1 & 0 \end{bmatrix}\begin{bmatrix} \upsilon \\ r \\ \psi \end{bmatrix}=\begin{bmatrix} Y \\ N \\ 0 \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} m-Z_{\dot{w}} & mx_g-Z_{\dot{q}} & 0 & 0 \\ mx_g & I_y-M_{\dot{q}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix}+\begin{bmatrix} -Z_w & mu_0-Z_q & 0 & 0 \\ -M_w & mx_g u_0-M_q & \left(z_g-z_b\right)W & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & u_0 & 0 \end{bmatrix}\begin{bmatrix} w \\ q \\ \theta \\ z \end{bmatrix}=\begin{bmatrix} Z \\ M \\ 0 \\ 0 \end{bmatrix} \tag{11}$$

In this model, the control signals for maintaining stable depth include the force Z applied along the z-axis of the vehicle to regulate the depth and the moment M around the y-axis to prevent oscillations. To maintain the heading of the ROV, the force Y applied along the y-axis controls lateral movement, while the moment N around the z-axis ensures heading stability.

The force components Y, Z, and moments M, N are all part of $\boldsymbol{\tau}$. Therefore, the general control signal for the system to maintain stability according to reference values is $\boldsymbol{\tau}$. These forces and moments are generated by the combined thrust produced by the thrusters of the BlueROV2. The calculation of control signals $\mathbf{u}$ for the thrusters is based on control allocation.

Consequently, the state space is typically represented as:

$$\begin{cases} \dot{\mathbf{x}}(t)=\mathbf{A}\mathbf{x}(t)+\mathbf{B}\boldsymbol{\tau}(t) \\ y(t)=\mathbf{C}\mathbf{x}(t) \end{cases} \tag{12}$$

where $\mathbf{x}(t)$ is considered a state vector, $y(t)$ is an output of the system and $\boldsymbol{\tau}(t)$ is the control input. In the context of heading control: $\mathbf{x}(t)=\begin{bmatrix}\upsilon & r & \psi\end{bmatrix}^{\mathrm{T}}$, $\boldsymbol{\tau}(t)=\begin{bmatrix}Y & N\end{bmatrix}^{\mathrm{T}}$, $y(t)=\psi$, while in the context of depth control: $\mathbf{x}(t)=\begin{bmatrix}w & q & \theta & z\end{bmatrix}^{\mathrm{T}}$, $\boldsymbol{\tau}(t)=\begin{bmatrix}Z & M\end{bmatrix}^{\mathrm{T}}$, $y(t)=z$.

3.2 LQR controller

For a linear system, the LQR controller is employed to determine the optimal control input, ensuring the system's adherence to the desired state while minimizing energy consumption. LQR is a type of full-state feedback controller, with the control structure as depicted in Figure 2. The objective is to determine the optimal gain matrix K, as defined in the control law:

$$\boldsymbol{\tau}(t)=-\mathbf{K}\mathbf{x}(t) \tag{13}$$

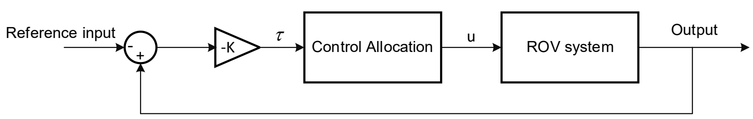


Figure 2. Structure of the LQR-Tracking controller

Given a linear time-invariant system:

$$\dot{\mathbf{x}}(t)=\mathbf{A}\mathbf{x}(t)+\mathbf{B}\boldsymbol{\tau}(t) \tag{14}$$

The infinite horizon LQR problem can be solved by finding the solution to the algebraic Riccati equation:

$$\mathbf{A}^{\mathrm{T}}\mathbf{P}+\mathbf{P}\mathbf{A}-\mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P}+\mathbf{Q}=0 \tag{15}$$

To minimize the cost function:

$$J(\tau)=\int_0^\infty (\mathbf{x}^\mathrm{T}\mathbf{Q}\mathbf{x}+\boldsymbol{\tau}^\mathrm{T}\mathbf{R}\boldsymbol{\tau})dt \quad (16)$$

where $\mathbf{Q}$ and $\mathbf{R}$ are the weighting matrices that determine the performance of the system; $\mathbf{Q}$ assigns weights to the system states, while $\mathbf{R}$ assigns weights to the control inputs.
The optimal gain matrix $\mathbf{K}$ can be computed as:

$$\mathbf{K}=\mathbf{R}^{-1}\mathbf{B}^\mathrm{T}\mathbf{P} \quad (17)$$

3.3 Thruster model and control allocation

Considering BlueROV2 with 6 thrusters, the generalized forces and moments in 6 DoFs $\boldsymbol{\tau}\in\mathbb{R}^6$ due to 6 thrusters in terms of control inputs $\mathbf{u}\in\mathbb{R}^6$ can be then modeled as:

$$\boldsymbol{\tau}=\mathbf{T}(\boldsymbol{\alpha})\mathbf{F}=\mathbf{T}(\boldsymbol{\alpha})\mathbf{K}\mathbf{u} \quad (18)$$

where $\mathbf{T}=\begin{bmatrix}\mathbf{t}_1 & \mathbf{t}_2 & \mathbf{t}_3 & \mathbf{t}_4 & \mathbf{t}_5 & \mathbf{t}_6\end{bmatrix}\in\mathbb{R}^{6\times6}$ is the thrust configuration matrix, $\mathbf{t}_i$ are column vectors of dimension 6×1 (with i = 1, 2, ..., 6) and $\boldsymbol{\alpha}\in\mathbb{R}^6$ is the thrust rotation angle vector, $\mathbf{F}=[F_1,F_2,F_3,F_4,F_5,F_6]^\mathrm{T}$ is the thruster forces and the thrust coefficient $\mathbf{K}=diag[K_1,K_2,K_3,K_4,K_5,K_6]$.

Control allocation calculates the input signal for the thrusters to generate the desired overall control force. Since the forces and moments produced by the thrusters depend on the control inputs, which are expressed by (18), the control input vector can be determined as follows [12]:

$$\mathbf{u}=\mathbf{K}^{-1}\mathbf{T}^{-1}\boldsymbol{\tau} \quad (19)$$

4. Build a co-simulation environment and HILS

4.1 Developing a Joystick for HILS

The joystick, programmed using an ESP32-S3 microcontroller, features six axes and thirty-one buttons [13]. Its signals are read by the joy_node [14], which publishes the data to a designated topic. The joystick transmits setpoint values for ROV control, where two specific values, depth (z) and heading (ψ), are regulated by the control system. Figure 3 illustrates the data flow from the joystick to Simulink™.

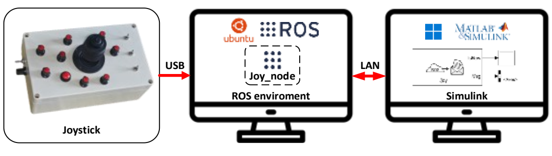


Figure 3. Data from Joystick to Simulink™.

4.2 Simulink™-Gazebo co-simulation with HILS integration

In the co-simulation framework between Simulink™ and Gazebo, Simulink™ is used for designing and analyzing control algorithms, while Gazebo allows testing the ROV physical model under various simulated conditions. This integration reduces development costs, improves testing accuracy, and enables realistic scenario simulations. The physical properties of the BlueROV2 system are modeled using BlueROV2 ROS Simulation [15], and the underwater environment is simulated via uuv_simulator [16]. The Simulink™ blocks are designed to receive desired depth and heading data from the joystick and feedback data from the IMU and pressure sensor, which are then fed into the LQR-Tracking controller. Subsequently, the control signals are transmitted through the Thruster Dynamic (Control Allocation) block, yielding six distinct control signals for the thrusters. These signals are then sent from Simulink™ to ROS, enabling the control of the BlueROV2 model within the Gazebo environment.

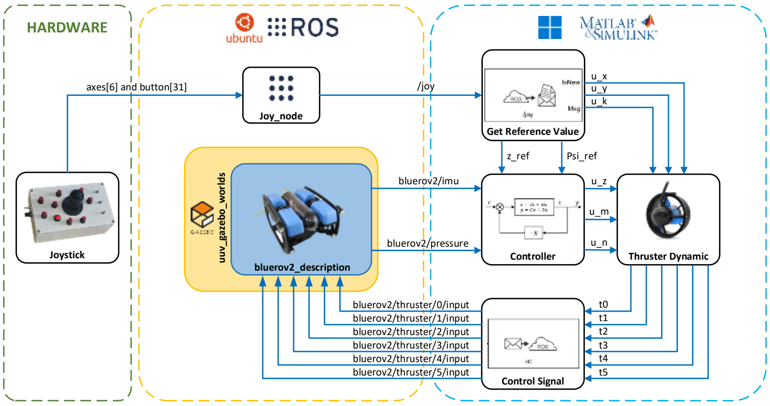
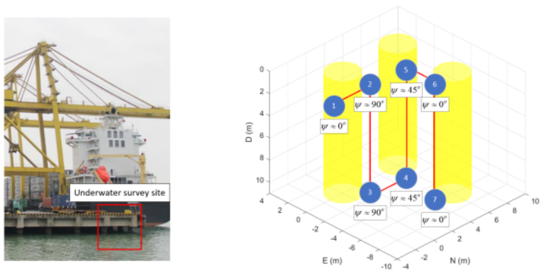


Figure 4. Overview of Simulink™-Gazebo co-simulation with HILS.

5. Case Study-Tien Sa Pier Inspection at Da Nang Port with BlueROV2

To evaluate the performance of the controller, an experiment was conducted using the BlueROV2 to inspect underwater structures at Da Nang Port in a simulated environment. Da Nang Port, especially the Tien Sa pier, requires regular underwater inspections to ensure safety and compliance [17]. The inspection method using specialized equipment for structures that cannot be visually observed is a key approach [17]. The BlueROV2, equipped with an HD camera and illumination lights, is ideal for this task. It will inspect three near-shore pillars (3m diameter) at depths of 10-12m under mild currents (0.2 m/s). The ROV will navigate from a near-shore starting point to pillar 1, aligning its heading towards pillar 1 ($\psi \approx 90°$), and then descend to inspect pillar 1. Upon reaching a depth of 10m, the ROV will move to pillar 2 while aligning its heading towards pillar 2 ($\psi \approx 45°$), and subsequently, ascend to inspect pillar 2. Similarly, for pillar 3, the heading will be aligned towards pillar 3 ($\psi \approx 0°$) for inspection. Due to its tethered connection, full pillar circumnavigation is not possible. Figure 5 **(b)** illustrates the trajectory.



(a) **(b)**

Figure 5. Inspection campaign. **(a)** Actual location, **(b)** trajectory of the inspection campaign.

The model will be operated entirely manually via a joystick. Coordinates such as x and y, which are controlled manually, are therefore not discussed in this paper. The control signal from the joystick is also depicted in Figure 7. The reference values for z and ψ are set manually, resulting in values that approximate the planned trajectory outlined in Figure 5. Figure 6 provides a view of the simulation interface, including the camera display and the pillars requiring inspection.

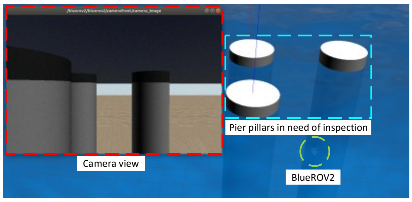
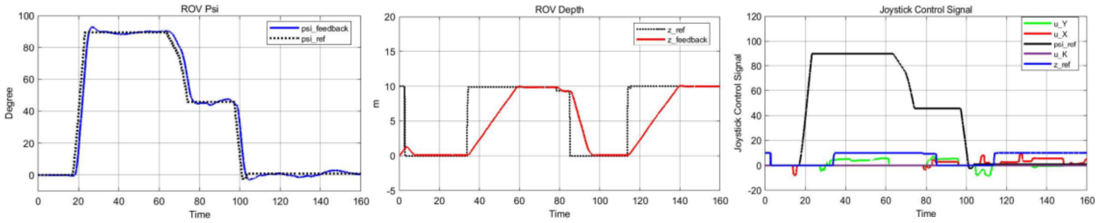


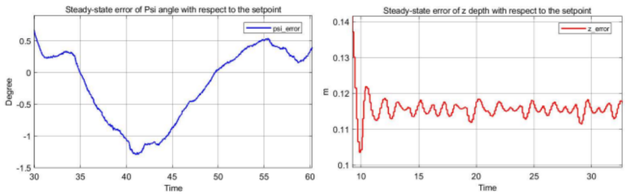
Figure 6. Simulation interface.



(a) **(b)** **(c)**

Figure 7. Response of the system output to joystick signals. **(a)** z coordinate response, **(b)** ψ orientation response, **(c)** Joystick control signal.

From Figure 7, in the heading control, it can be seen that the settling time of the system is 12s, with a ψ value of 90 degrees. For depth control, it takes 25 seconds to establish a depth of 10m.



(a) **(b)**

Figure 8. Output error**. (a)** ψ error, **(b)** z error.

BlueROV2 specifications (6 thrusters, 457 mm length, 100 m maximum depth) were included in the model. The **Q** and **R** matrices were tuned using a trial-and-error approach: larger **Q** values for depth and heading improved tracking accuracy, while higher **R** values reduced thruster saturation, smoothed responses, and optimized control energy usage at the cost of slightly slower settling times. This tuning achieved a balance between accuracy and efficiency. For benchmarking, we compared our LQR controller against PID designs reported in [8] and [9]. The ψ angle accuracy improves from ±2° [9] to ±1.5°, while the settling time shows a clear advantage: 25 s for a 10 m step, compared with 18 s per meter in [9] and 10 s per meter in [8]. These results align with recent surveys on advanced underwater vehicles and control strategies [10,18,19], supporting the robustness and sustainability of our approach

6. Conclusion

This study validated an LQR tracking controller for BlueROV2 in MATLAB–Gazebo co-simulation with HILS. Beyond technical validation, the framework contributes to sustainable environmental monitoring by reducing the need for repeated field trials and supporting safe underwater inspections. This complements recent advances in soft robotics for environmental sensing [20] and reviews of marine robot adaptation to environmental conditions [9]. Together, these works underline the environmental value of our framework for ecosystem monitoring and underwater infrastructure assessment. In the future, the algorithm can be tested on real hardware or further advanced controllers can be developed to optimize ROV control performance.

Acknowledgment

The authors gratefully acknowledge the support from Ho Chi Minh City University of Technology and Education for funding project SV2025-72. We would also like to express our sincere gratitude to MES Service Co., Ltd. for their invaluable support throughout this research.

References

[1] Grand View Research 2023 *Underwater Robotics Market Size, Share & Trends Analysis Report by Type (ROV, AUV), By Application (Commercial Exploration, Defense & Security, Scientific Research), By Region, and Segment Forecasts, 2023–2030*. Grand View Research.
[2] Bessa W M, Dutra M S and Kreuzer E 2010 An adaptive fuzzy sliding mode controller for remotely operated underwater vehicles. *Robotics and Autonomous Systems* 58 16–26.
[3] Abidin A Z, Mardiyanto R and Purwanto D 2016 Implementation of PID controller for hold altitude control in underwater remotely operated vehicle. *Proc. Int. Seminar on Intelligent Technology and Its Applications (ISITIA)* (Lombok, Indonesia) pp 665–670.
[4] Rúa Pérez S and Vasquez R 2016 Development of a low-level control system for the ROV Visor3. *International Journal of Navigation and Observation* 2016 1–12.
[5] Dong M, Li J and Chou W 2020 Depth control of ROV in nuclear power plant based on fuzzy PID and dynamics compensation. *Microsystem Technologies* 26 811–821.
[6] Azmi M W N, Aras M S M, Zambri M K M, Farihan M I, Yasin M H M and Razlan Z M 2018 Comparison of controllers design performance for underwater remotely operated vehicle (ROV) depth control. *International Journal of Engineering and Technology* 7 419–423.
[7] Binugroho E H, Jatmiko W, Santoso A, Nugroho S M, Santoso S, Pramono A and Mas'udi M I 2019 EROV: depth and balance control for ROV motion using fuzzy PID method. *Proc. International Electronics Symposium (IES)* (IEEE) pp 637–643.
[8] Ford D A, Carey S, Lerner J, Boss E, Fearnbach H, Karpovich S, Matos L, Pizarro O and Xu J 2022 A solution for autonomous, adaptive monitoring of coastal ocean ecosystems. *Frontiers in Marine Science* 9 1067174.
[9] Chen Z, Yang W, Wang J, Liu Y, Lu G, Tang K and Wu Z 2023 A survey on environmental adaptation technologies for marine robots. *Ocean Engineering* 286 115234.
[10] Xu J, Li G, Tang K, Chen Z and Liu Y 2024 State-of-the-art underwater vehicles and technologies enabling smart ocean. *arXiv* 2412.18667.
[11] Fossen T I 1994 *Guidance and Control of Ocean Vehicles* (Chichester: Wiley).
[12] Wu C J 2018 6-DoF modelling and control of a remotely operated vehicle. MSc Thesis, Flinders University.
[13] Espressif Systems 2025 *Gamepad.ino arduino-esp32*. Available at: https://github.com/espressif [Accessed 20 February 2025].
[14] ROS Wiki 2025 *joy ROS Wiki*. Available at: http://wiki.ros.org/joy [Accessed 10 February 2025].
[15] UUVControl 2025 *bluerov2 GitHub*. Available at: https://github.com/uuvcontrol/bluerov2 [Accessed 10 February 2025].
[16] Sørensen A J, Schjølberg I and Borhaug E 2025 *uuv_simulator GitHub*. Available at: https://github.com/uuvsimulator [Accessed 10 February 2025].
[17] Thu Vien Phap Luat 2025 *Periodic inspection activities for seaport works: methods and general requirements* (in Vietnamese). Available at: https://thuvienphapluat.vn [Accessed 5 March 2025].
[18] Thanh P M T, Can V V and Anh N D 2018 Research, design and control of an ROV. *Journal of Transport Science and Technology* 27–28 45–52. (in Vietnamese).
[19] Tao T V, Huy T N, Dat H T and Hai C T 2018 Design of a mini-ROV. *Journal of Transport Science and Technology* 30 60–66. (in Vietnamese).
[20] Li G, Yang W, Zhai D, Xu J, Tang K, Lu G, Chen Z, Wang J and Liu Y 2023 Bioinspired soft robots for deep-sea exploration. *Nature Communications* 14 7097.