CEA Comité Español de Automática

XLV JORNADAS AUTOMÁTICA MÁLAGA 2024

UNIVERSIDADE DA CORUÑA

Jornadas de Automática

Seguimiento de trayectoria de un AUV para la inspección de jaulas de red utilizando control por modos deslizantes

López-Barajas, S.[a,b,*], Sanz, P. [a,b], Marín, R.[b], Solis, A.[b], Echagüe, J.[b], Castañeda, H.[c]

[a] *ValgrAI—Valencian Graduate School and Research Network for Artificial Intelligence, Camí de Vera S/N, Edificio 3Q, 46022 Valencia, Spain*
[b] *Interactive Robotic Systems Lab, Jaume I University, 12071 Castellón de la Plana, Spain. ORCID: (0009-0004-0410-2888), (0000-0002-2340-4126), (0000-0003-3382-1553), (0009-0006-1261-2209) and (0000-0001-8310-6713)*
[c] *Tecnologico de Monterrey, School of Sciences and Engineering, Monterrey 64849, Mexico. ORCID: 0000-0002-2432-7740*

To cite this article: López-Barajas, S., Sanz, P., Marín, R., Solis, A., Echagüe, J., Castañeda, H. 2024. Trajectory tracking of a AUV for net cage inspection using sliding code control.
Jornadas de Automática, 45. https://doi.org/10.17979/ja-cea.2024.45.10789

Resumen

A medida que la industria de la acuicultura sigue expandiéndose, la necesidad de métodos de inspección eficientes y precisos para estructuras submarinas se vuelve cada vez más crucial. El control de modos deslizantes es un método de control no lineal que ha demostrado una gran robustez frente a perturbaciones e incertidumbres en el modelo. Esto se ha probado en varios estudios donde esta técnica se ha aplicado en robótica móvil, incluyendo vehículos terrestres, drones aéreos y vehículos submarinos. Este artículo presenta el diseño e implementación de un controlador adaptativo por modos deslizantes para el seguimiento de trayectorias de un vehículo submarino autónomo (AUV) para la inspección de jaulas de red.

Los resultados de la simulación destacan la efectividad del controlador guiar al AUV a lo largo de trayectorias predefinidas. El estudio enfatiza el potencial del control por modos deslizantes para mejorar la capacidad del AUV de navegar con precisión en entornos complejos susceptibles a perturbaciones como las corrientes oceánicas, facilitando inspecciones eficientes de jaulas para el monitoreo de la acuicultura.

Palabras clave: Guiado, navegación y control en sistemas marinos, Sliding modes control, Vehículos submarinos autónomos, Control adaptativo y robusto en sistemas marinos.

Trajectory tracking of an AUV for net cage inspection using sliding code control

Abstract

As the aquaculture industry continues to expand, the need for efficient and precise inspection methods for underwater structures, such as net cages, becomes increasingly crucial. Sliding Mode Control is a non-linear control method that has demonstrated significant robustness when facing disturbances and model uncertainties. This has been proven in various studies where this technique has been applied in mobile robotics, including ground vehicles, aerial drones, and underwater vehicles. The following article presents the design and implementation of an Adaptive Sliding Mode Controller for trajectory tracking of an Autonomous Underwater Vehicle (AUV) during net cage inspection.

Simulation results highlight the effectiveness of the proposed approach in guiding the AUV along predefined trajectories with high accuracy. The study emphasizes the potential of Sliding Mode Control to improve the AUV's ability to accurately navigate in complex environments susceptible to disturbances such as ocean currents, facilitating efficient net cage inspections for aquaculture monitoring.

Keywords: Guidance, navigation, and control in marine systems, Sliding mode control, Autonomous underwater vehicles, Adaptive and robust control in marine systems.

*Autor para correspondencia: barajas@uji.es

1. Introducción

Nowadays, aquaculture holds a significant position within the food industry. As of 2020, aquaculture accounted for nearly 50% of the global fish production, as reported by the Food and Agriculture Organization (FAO) FAO (2022). The majority of aquaculture operations are situated in large sea-based fish farms with net cages. These net cages are essentially cylindrical structures ranging from 20 to 30 meters in diameter and with depths spanning approximately 15 to 48 meters Holen et al. (2018).

Fish farm environments are hostile, workers are daily exposed to hazards and safety risks. Most security risks are associated with routine tasks such as net inspection, cleaning the net mooring and net repairing. These tasks are done by divers. It has been demonstrated in the literature and in some commercial products that underwater robots embedded with autonomous behaviour can increase the comfort and safety to workers in fish farms. A recent work about a robotic platform composed by a surface vehicle and an underwater robot capable of detecting holes at net cages was published by the authors of this article López-Barajas et al. (2024). That article shows an overview of how an inspection can be executed and all the procedures that must be taken into account. Other examples of robotic applications of fish farms can be found in Vasileiou et al. (2022), Ohrem et al. (2020) or Sousa et al. (2010).

Net inspection is one of most time-consuming tasks for divers. This is why in the literature underwater robots had been used to manually or semi-autonomous inspect the total area of a net cage. However, the lack of an accurate positioning method and a high bandwidth wireless communications combined with ocean currents and numerous obstacles during navigation, amplifies the complexity of the problem. Articles like Lin et al. (2020), Akram et al. (2022) and López-Barajas et al. (2023) show how researchers have boarded this problem.

This article contributes in the area of disturbance rejection when tracking the trajectory of an underwater vehicle using an adaptive sliding mode controller. This is one of the first steps in the research line of autonomous net inspection, further steps will be obstacle avoidance and embedding autonomous decision making. The rest of the paper is organized as follows: Section 2 describes the basic concepts of the modelling an underwater vehicle and tools used to develop the simulation. Section 3 describes the design of the controller. Section 4 presents the results and discussion of the experiment. Finally, Section 5 exposes the conclusions and future research lines.

2. Materials and Methods

In this section, a general description of the kinematics and hydrodynamics of underwater vehicles is provided. Once the concepts of underwater vehicles have been defined, the modeling of the BlueROV2 is described. Subsequently, the simulation and experimental setup are presented. Finally, the desired trajectories and disturbances are presented.

2.1. Underwater Vehicles Kinematics and Hydrodynamics

The model for marine vehicles used was proposed by Fossen Fossen (2011), in this model two frames are used to describe the motion of the robot, the body fixed frame and the North-East-Down (NED) frame. The orthonormal axes for the body frame are x_b, y_b, z_b and the axes for the NED frame are N, E, D. An example of this is shown in Fig. 1.

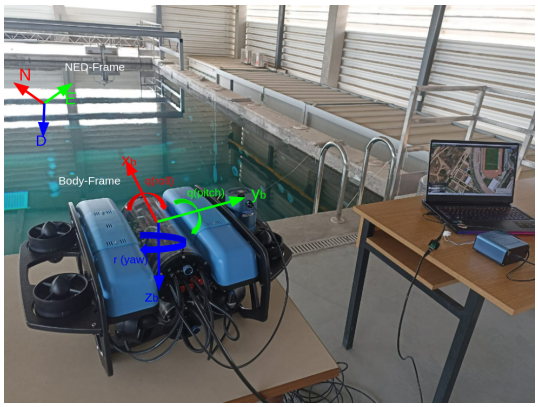

Figure 1: Body Frame and NED of the BlueROV2

Based on the reference system described before, the Society of Naval Architects and Marine Engineers (SNAME) established the next standard to express position, orientation, moments, and forces. The respective equations are shown in table 1.

Variables	Equations
Position (η)	$\eta = (x, y, z, \varphi, \theta, \psi)^T$
Velocity (υ)	$\upsilon = (u, v, w, p, q, r)^T$
Force and Moments (τ)	$\tau = (X, Y, Z, K, M, N)^T$

Table 1: Position, Velocity, Forces and Moments standard by the SNAME.

In Table 1, η stands for both position and Euler rotation angles, υ represents both linear and angular velocities, and the vector τ denotes forces and moments.

When discussing a vehicle's motion dynamics, the model is represented using Newton-Euler equilibrium laws. An adapted version for a maritime vehicle model would be as follows:

$$M\dot{\nu} + C(\nu)\dot{\nu} + D(\nu)\nu + g(\eta) = \tau + w \tag{1}$$

$$\tau = B * u \tag{2}$$

In equations (1) and (2), the variable M denotes the combined inertial mass matrix and added mass. The symbol C stands for the rigid body matrix and added mass, incorporating Coriolis and centripetal values. D represents hydrodynamic damping, B signifies the thruster configuration, g accounts for restoring forces, u denotes the forces generated by the thrusters, w represents disturbances, and lastly, τ symbolizes the controller outputs vector.

If water currents are taken into account the model can be expressed as follows:

$$M_{RB}\dot{\nu} + C_{RB}(\nu)\nu + M_A\dot{\nu}_w + C_A(\nu_w)\nu_w + D(\nu_w)\nu_w + g(\eta) = \tau \tag{3}$$

Where

$$M = M_{RB} + M_A \tag{4}$$

$$C = C_{RB} + C_A \tag{5}$$

$$\nu_w = \nu - \nu_c \tag{6}$$

Equation (4) shows the rigid body and added mass matrices. While (5) represents the rigid body Coriolis and centripetal and the added mass Coriolis and centripetal matrix. ν_w represents the relative velocity vector and ν_c represents the water current vector, both in m/s.

Assuming that the vehicle is a rigid body and the NED frame is an inertial system, transformation between the body fixed frame and NED frame can be developed as Fossen states in Chapter II of the book Guidance and Control of Ocean Vehicles Fossen (2011). The transformation equations between the body fixed frame and the NED fixed frame, in this case the North East Down (NED) frame are presented next.

$$\dot{\eta} = J(\eta_2)\nu \longleftrightarrow \nu = J^{-1}(\eta_2)\dot{\eta} \tag{7}$$

$$\ddot{\eta} = J(\eta_2)\dot{\nu} + \dot{J}(\eta_2)\nu \longleftrightarrow \dot{\nu} = J^{-1}(\eta_2)[\ddot{\eta} - J(\eta_2)\nu] \tag{8}$$

Where

$$J(\eta_2) = \begin{bmatrix} J_1(\eta_2) & 0_{3\times3} \\ 0_{3\times3} & J_2(\eta_2) \end{bmatrix} \tag{9}$$

$$J_1(\eta_2) = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + s_\phi s_\theta c_\psi & s_\psi s_\phi + s_\theta c_\psi c_\phi \\ s_\psi c_\phi & c_\psi c_\phi + s_\phi s_\theta s_\psi & -c_\psi s_\phi + s_\theta s_\psi c_\phi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \tag{10}$$

$$J_2(\eta_2) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix} \tag{11}$$

$$\eta = [N, E, D, \varphi, \theta, \psi]^T \tag{12}$$

$$\eta_2 = [\varphi, \theta, \psi] \tag{13}$$

In equations (10) and (11) s_{angle}, c_{angle} and t_{angle} are abbreviations for $sin(angle)$, $cos(angle)$ and $tan(angle)$. Equation (13) represents the Euler angles considering the singularity at $\theta = \pm\pi/2$. Using equations from (7) to (13) the transformation of the equation of motion from the body to the NED frame can be derived as follows:

$$M^*(\eta)\ddot{\eta} + C^*(\nu,\eta)\dot{\eta} + D^*(\nu,\eta)\dot{\eta} + g^*(\eta) + g_o^*(\eta) = \tau^* \tag{14}$$

Where

$$\begin{aligned} M^*(\eta) &= J^{-T}(\eta)MJ^{-1}(\eta) \\ C^*(\nu,\eta) &= J^{-T}(\eta)\left[C(\nu) - MJ^{-1}(\eta)\dot{J}(\eta)\right]J^{-T}(\eta) \\ D^*(\nu,\eta) &= J^{-T}(\eta)D(\nu)J^{-1}(\eta) \\ g^*(\eta) + g_o^*(\eta) &= J^{-T}(\eta)\left[g(\eta) + g_o\right] \\ \tau^* &= J^{-T}(\eta)(\tau + w) \end{aligned} \tag{15}$$

These transformations are summarized from the chapter 7 of Fossen (2011), and from here the controller can be designed to track a desired trajectory in the NED frame.

2.2. BlueROV2

In this subsection the modelling of the BlueROV2 heavy configuration (underwater robot used for this experiment) is presented. The inertial mass, rigid body, added mass, Coriolis and centripetal, hydrodynamic damping and the hydrostatic matrix are expressed bellow.

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mZ_g & 0 \\ 0 & m & 0 & -mZ_g & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & -mZ_g & 0 & I_x & 0 & 0 \\ mZ_g & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{bmatrix}$$

$$M_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & -Y_{\dot{v}} & 0 & 0 & 0 & 0 \\ 0 & 0 & -Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & -K_{\dot{p}} & 0 & 0 \\ 0 & 0 & 0 & 0 & -M_{\dot{q}} & 0 \\ 0 & 0 & 0 & 0 & 0 & -N_{\dot{r}} \end{bmatrix}$$

$$C_{RB}(\nu) = \begin{bmatrix} 0 & 0 & 0 & 0 & mw & 0 \\ 0 & 0 & 0 & -mw & 0 & 0 \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & mw & -mv & 0 & -I_z r & -I_y q \\ -mw & 0 & -mu & -I_z r & 0 & -I_x p \\ mv & -mu & 0 & I_y q & -I_x p & 0 \end{bmatrix}$$

$$C_A(\nu) = \begin{bmatrix} 0 & 0 & 0 & 0 & z_{\dot{w}} w & 0 \\ 0 & 0 & 0 & -z_{\dot{w}} w & 0 & -X_{\dot{u}} u \\ 0 & 0 & 0 & -Y_{\dot{v}} v & X_{\dot{u}} u & 0 \\ 0 & Z_{\dot{w}} w & Y_{\dot{v}} v & 0 & -N_{\dot{r}} r & M_{\dot{q}} q \\ Z_{\dot{w}} w & 0 & X_{\dot{u}} u & N_{\dot{r}} r & 0 & -K_{\dot{p}} p \\ -Y_{\dot{v}} v & X_{\dot{u}} u & 0 & -M_{\dot{q}} q & K_{\dot{p}} p & 0 \end{bmatrix}$$

$$\begin{aligned} D(\nu) = -\mathrm{diag}[&(X_u + X_{u|u|}|u|), (Y_v + Y_{v|v|}|v|), \\ &(Z_w + Z_{w|w|}|w|), (K_p + K_{p|p|}|p|), \\ &(M_q + M_{q|q|}|q|), (N_r + N_{r|r|}|r|)] \end{aligned}$$

$$g(\eta) = \begin{bmatrix} (W - B)\sin\theta \\ -(W - B)\cos\theta\sin\phi \\ -(W - B)\cos\theta\cos\phi \\ z_g W\cos\theta\sin\phi \\ z_g W\sin\theta \\ 0 \end{bmatrix}$$

The experimental values are shown bellow as Wu Wu (2018) published on this Master Thesis.

Param.	Value	Param.	Value
m	11.5 kg	W	112.8 N
r_b	[0, 0, 0] m	$[X_g, Y_g, Z_g]$	[0, 0, 0.02] m
B	114.8 N	I_x	0.16 kg m^2
I_y	0.16 kg m^2	I_z	0.16 kg m^2
$\dot{X}_u$	−5.5 kg	$\dot{Y}_v$	−12.7 kg
$\dot{Z}_w$	−14.57 kg	$\dot{K}_p$	−0.12 kg m^2/rad
$\dot{M}_q$	−0.12 kg m^2/rad	$\dot{N}_r$	−0.12 kg m^2/rad
X_u	−4.03 Ns/m	$X_{u\|u\|}$	−18.18 Ns^2/m^2
Y_v	−6.22 Ns/m	$Y_{v\|v\|}$	−21.66 Ns^2/m^2
Z_w	−5.18 Ns/m	$Z_{w\|w\|}$	−36.99 Ns^2/m^2
K_p	−0.07 Ns/rad	$K_{p\|p\|}$	−1.55 Ns^2/rad^2
M_q	−0.07 Ns/rad	$M_{q\|q\|}$	−1.55 Ns^2/rad^2
N_r	−0.07 Ns/rad	$N_{r\|r\|}$	−1.55 Ns^2/rad^2

Table 2: BlueROV2 modelling parameters.

2.3. Simulation setup

2.3.1. Simulink

The mathematical model of the robot, the controller and set point was developed in Simulink. Equation (14) was constructed based in the model described in section 2.2 and using Simulink environment tools (Gains, Integrators and Matlab Functions). Fig. 2 shows a diagram of the complete system and the simulation parameters are: Start time: 0.0s; Stop time: 180s; Solver: ode1 (Euler); Type: Fixed-step; Fixed-step size: 0.01s.

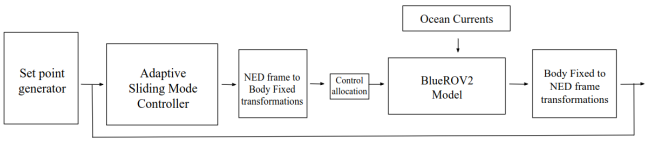

Figure 2: Simulink block model, including controller, set point, robot model and transformations from body to the NED frame.

2.3.2. Unity simulation environment

A middle step between testing the simulator in simulink and testing it on the real robot is the Unity Simulator designed in Sanz et al. (2023). This simulator was designed taking into account the real size of the CIRTESU (Research Centre for Robotics and Underwater Technologies) water tank and the real model of the BlueROV2. The actuators and sensors can be accessed using ROS, same as the real one. In this context, the visualization and monitoring of the trajectory and controller performance play crucial roles when tunning the control parameters. This tool is significant for bridging the gap between testing the controller in Simulink and deploying it on the actual robot. Fig. 3 shows the simulation environment.

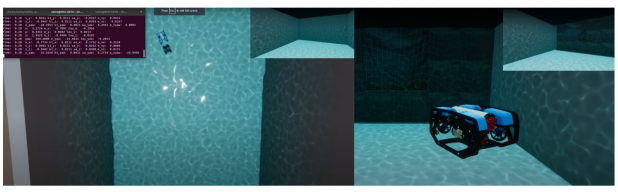

Figure 3: Left image top view of the simulator scene; Right image close up to the BlueROV2 model.

2.3.3. Desired trajectories and Disturbances

Performance of the controller will be tested using two desired trajectories, basically the trajectory is the same but the velocity changes. There is the Optimal Net Inspection Trajectory (ONIT) and the Stress Testing Trajectory (STT). Both trajectory have the same waypoints, but ONIT is done in 180s (60s per cycle) and STT is done in 60s (20s per cycle). Table 3 shows the parameters of the helical trajectories and the initial parameters for both experiments.

Parameters	**Simulink**	**Unity**
Total duration (ONIT)	180 sec	180 sec
Total duration (STT)	60 sec	60 sec
Circle iterations	3 turns	3 turns
Radio	2.5 m	2.5 m
Center of the radio	[0,0] m	[0,0] m
Initial depth	1 m	1 m
Final depth	4 m	4 m
Initial yaw	0 rad	0 rad
Final yaw	6π rad	6π rad
Initial position	[2.5, 0, -1]	[0, 2.5, -4]

Table 3: Desired set point trajectory parameters.

Ocean currents were also modeled in the simulator, In this specific case, irrotational currents were proposed and the maximum amplitude chosen was 0.3 m/s as Haug reported in his Master Thesis "Hydrodynamic Study of ROV (Remotely Operated Vehicle) Operations at Net-based Fish Farms" Haug (2020). The Ocean current vector is expressed next:

$$\nu_c(t) = [0.1 * cos(t), 0.2 * cos(t), \nu_{c_z}(t), 0, 0, 0] \tag{16}$$

where

$$\nu_{c_z} = \begin{cases} 0, & \text{si } t < 40 \\ -0.3, & \text{si } t \geq 40 \end{cases}$$

3. Controller Design

The controller proposed in this article is a non-linear Adaptive Sliding Mode Controller with practical finite time stability. From equation 14 and solving from $\ddot{\eta}$:

$$\ddot{\eta} = M^{-1}(\tau - C(\dot{\eta})\dot{\eta} - D(\dot{\eta})\dot{\eta} - g(\eta)) \tag{17}$$

For trajectory tracking the position error states as follow:

$$e = \eta_d - \eta \tag{18}$$

Where η_d represent the desired trajectory and η the real NED frame position. Once the error was stated, the sliding surface can be represented as follow:

$$\sigma = \dot{e} + \lambda e \tag{19}$$

$$\dot{\sigma} = \ddot{e} + \lambda \dot{e} \tag{20}$$

From (18) and (20):

$$\dot{\sigma} = \ddot{\eta}_d - \ddot{\eta} + \lambda \dot{e} \tag{21}$$

After some mathematical manipulation from equation 17 and 21, the sliding surface can be expressed as:

$$\sigma = \ddot{\eta}_d - M^{-1}(\tau - C(\dot{\eta})\dot{\eta} - D(\dot{\eta})\dot{\eta} - g(\eta)) + \lambda \dot{e} \tag{22}$$

The proposed linearized controller to overcome the dynamics of the model and to drive the system to a sliding mode condition equal to zero will be:

$$\tau = C(\dot{\eta})\dot{\eta} + D(\dot{\eta})\dot{\eta} + g(\eta)) + M(\ddot{\eta}_d + \lambda\dot{e} - u_a) \tag{23}$$

Where u_a is an auxiliary controller proposed as:

$$u_a = -k_1(t)\,|\sigma(t)|^{1/2}\, sign(\sigma(t)) - k_2\sigma(t) \tag{24}$$

with, $k_2 > 0$ and

$$\dot{k_1}(t) = \begin{cases} k\ \text{sign}(|\sigma| - \mu) & \text{if } k_1 > k_{\min} \\ k_{\min} & \text{if } k_1 \leq k_{\min} \end{cases} \tag{25}$$

In equation (25) k_1 control the adaptability rate of adaptive gain k_1; μ is the adaptability threshold gain and k_{min} symbolizes the minimum effort gain.

In the case of this work a non-model-based controller is proposed, this means that the linearizing terms from equation 23 are omitted, and the expression for the non-model-based controller is:

$$\tau = \ddot{\eta}_d + \lambda\dot{e} - u_a \tag{26}$$

Stability analysis of this controller can be find in section IV of the article published by A. Gonzalez-Garcia Gonzalez-Garcia and Castañeda (2022).

4. Results and Discussion

4.1. Simulink Results

As discussed in section 2.3.1 the controller will be tested using the ONIT and the STT. Fig. 4 and 5 show the performance of the sliding mode controller in the trajectory tracking problem while perturbed by irrotational ocean currents.

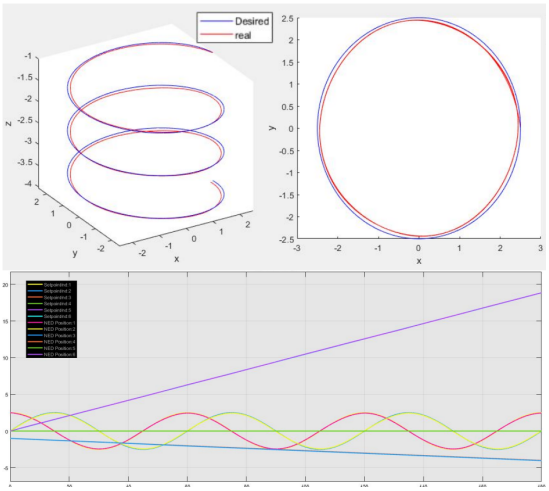

Figure 4: ONIT (60 seconds period); Up left image shows an isometric view; Upright image shows the top view; Bottom image shows the evolution of the trajectory over time.

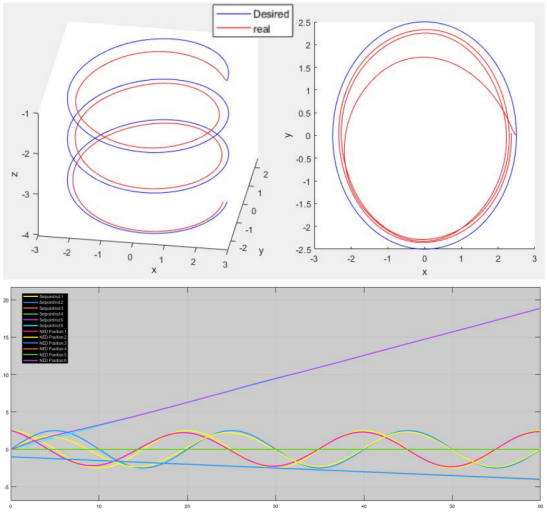

Figure 5: STT (20 seconds period); Up left image shows an isometric view; Upright image shows the top view; Bottom image shows the evolution of the trajectory over time.

Adaptive gain k1 evolution over time is shown in Fig. 6.

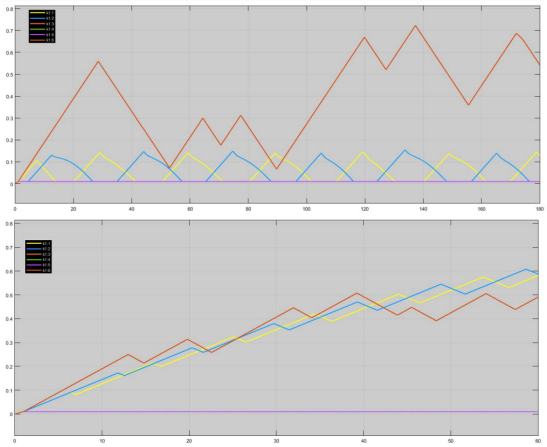

Figure 6: Adaptive gain k1 over time; Top image 60 second cycle period; Bottom image 20 second cycle period.

4.2. Unity Simulation environment

The control law was implemented on a Rosnode programmed in Python. Same as in the Simulink experiment the controller was tested using the ONIT and the STT, but as mentioned in 3 in this experiment the initial position was at a different depth (4m). Results of these two experiments are shown in figures 7 and 8. The videos of the experiment are shown in the next links:

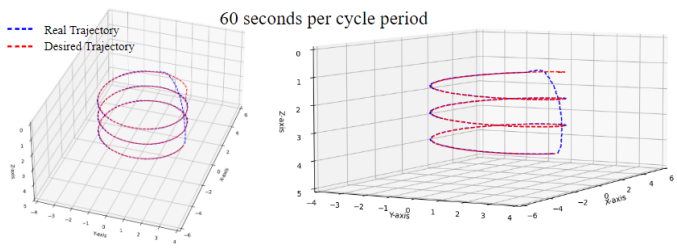

Figure 7: Both images are results of the ONIT at the Unity simulator; Left image shows an isometric view; Right image shows the horizontal view.

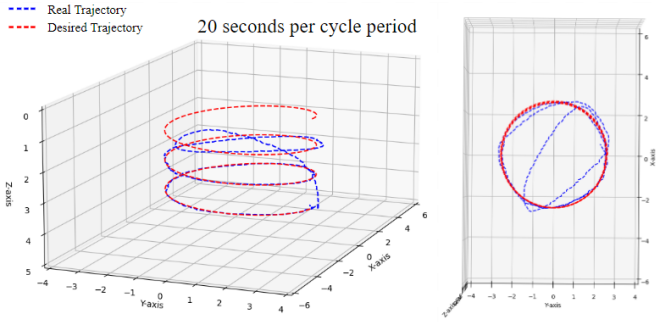

Figure 8: Both images are results of the STT at the Unity simulator; Left image shows an isometric view; Right image shows the top view.

- Video of the rendered ONIT
  - `https://youtu.be/jrdCnUtcL6M`
- Video 3D plot results ONIT
  - `https://youtu.be/TRohMLp5hVM`
- Video 3D plot results STT
  - `https://youtu.be/CE8lu0Tr3a8`

Fig. 9 shows the evolution of the adaptive gain over time.

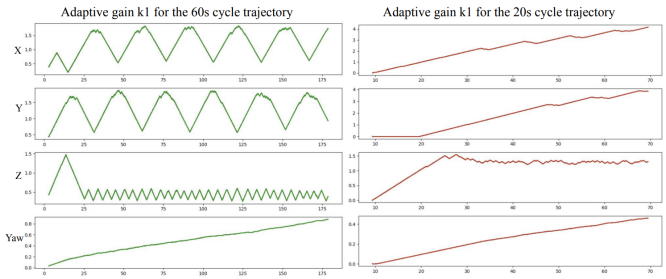

Figure 9: Adaptive gain k1 over time; Left image ONIT; Right image STT.

4.3. Discussion

It can be deduced from the results that the adaptive gain increases or decreases its value depending on the part of the circle (cosine function). When the velocity of the AUV in an axis is low, the adaptive gain decreases; conversely, it increases when the velocity is high. In the case of the STT, the adaptive gain increases until the real position converges with the desired position. Once this occurs, the adaptive gain stabilizes, resulting in a smaller peak. It is also evident that the STT trajectory in the Unity simulator causes the controller to take longer to converge. This is primarily due to tuning parameters. However, in the end, once the real trajectory converges with the desired one, the error remains small.

Finally, the parameter μ is crucial, mainly because it needs to strike a balance between the allowed error and controller chattering

5. Conclusions and Future Work

The controller proposed in this work demonstrated excellent performance in the trajectory tracking of an AUV. An important feature of this controller is its Free-model capability, which means that knowledge of the robot's model is not necessary to achieve accurate results, primarily due to the adaptive gain. For future work, implementing this controller in the Girona500 AUV and designing an adaptive guidance trajectory that depends on the actual error is recommended. For instance, if a disturbance increases the trajectory error to a significant value, the velocity of the desired trajectory will decrease. This aids the controller in returning to the sliding surface.

Agradecimientos

Thanks to ValgrAI – Valencian Graduate School and Research Network for Artificial Intelligence and Generalitat Valenciana for the economic support in the first author's PhD studies. This paper has been partially funded by MICINN under PDC2021-120791-C22 grant, and by MICINN through NextGenerationEU PRTR-C17.I1 grant, and by GVA ThinkInAzul/2021/037 grant.

References

Akram, W., Casavola, A., Kapetanović, N., Miškovic, N., 2022. A visual servoing scheme for autonomous aquaculture net pens inspection using rov. Sensors (Basel) 22 (9), 3525, pMID: 35591214; PMCID: PMC9099505.
DOI: 10.3390/s22093525

FAO, 2022. The State of World Fisheries and Aquaculture 2022-Towards Blue Transformation. FAO, Rome, Italy.
DOI: 10.4060/cc0461en

Fossen, T. I., 2011. Introduction. John Wiley & Sons, Ltd, Chichester, UK, pp. 3–12.

Gonzalez-Garcia, A., Castañeda, H., 2022. Adaptive integral terminal supertwisting with finite-time convergence for an unmanned surface vehicle under disturbances. International Journal of Robust and Nonlinear Control 32 (18), 10271–10291.
DOI: 10.1002/rnc.6368

Haug, L. T., Sep 2020. Hydrodynamic study of rov (remotely operated vehicle) operations at net-based fish farms. Master's thesis, Inst. Marin Tek., Fac. Ing., NTNU, Trondheim, Norway.

Holen, S. M., Utne, I. B., Holmen, I. M., Aasjord, H., 2018. Occupational safety in aquaculture—part 1: Injuries in norway. Marine Policy 96, 184–192.
DOI: 10.1016/j.marpol.2017.08.009

Lin, T. X., Tao, Q., Zhang, F., 2020. Planning for fish net inspection with an autonomous osv. In: Proceedings of the 2020 International Conference on System Science and Engineering (ICSSE). Kagawa, Japan, pp. 1–5.
DOI: 10.1109/ICSSE50014.2020.9219318

López-Barajas, S., Sanz, P. J., Marín, R., Solis, A., Marxer, R., Hugel, V., 2023. Automatic visual inspection of a net for fish farms by means of robotic intelligence. In: Proceedings of the OCEANS 2023 - Limerick. Limerick, Ireland, pp. 1–5.
DOI: 10.1109/OCEANSLimerick52467.2023.10244549

López-Barajas, S., Sanz, P. J., Marín-Prades, R., Gómez-Espinosa, A., González-García, J., Echagüe, J., 2024. Inspection operations and hole detection in fish net cages through a hybrid underwater intervention system using deep learning techniques. Journal of Marine Science and Engineering 12, 80.
DOI: 10.3390/jmse12010080

Ohrem, S. J., Kelasidi, E., Bloecher, N., 2020. Analysis of a novel autonomous underwater robot for biofouling prevention and inspection in fish farms. In: Proceedings of the 28th Mediterranean Conference on Control and Automation (MED). Saint-Raphaël, France, pp. 1002–1008.
DOI: 10.1109/MED48518.2020.9183157

Sanz, P. J., Marín, R., López-Barajas, S., Solis, A., Marxer, R., Hugel, V., 2023. 1st year of running mir at uji. In: Proceedings of the OCEANS 2023 - Limerick. Limerick, Ireland, pp. 1–5.
DOI: 10.1109/OCEANSLimerick52467.2023.10244270

Sousa, D., Hernandez, D., Oliveira, F., Luís, M., Sargento, S., 2010. A platform of unmanned surface vehicle swarms for real-time monitoring in aquaculture environments. In: Proceedings of the OCEANS 2010 IEEE SYDNEY. Sydney, NSW, Australia, pp. 1–6.
DOI: 10.1109/OCEANSSYD.2010.5603510

Vasileiou, M., Manos, N., Kavallieratou, E., 2022. Iura: An inexpensive underwater robotic arm for kalypso rov. In: Proceedings of the 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME). Maldives, Maldives, pp. 1–6.
DOI: 10.1109/ICECCME55909.2022.9988259

Wu, C.-J., July 2018. 6-dof modelling and control of a remotely operated vehicle. Master's thesis, Flinders University.