

Article

Experimental Validation of a Model-Free High-Order Sliding Mode Controller with Finite-Time Convergence for Trajectory Tracking of Autonomous Underwater Vehicles

Josué González-García ¹, Alfonso Gómez-Espinosa ^{1,*}, Luis Govinda García-Valdovinos ^{2,*}, Tomás Salgado-Jiménez ², Enrique Cuan-Urquiza ¹ and Jesús Arturo Escobedo Cabello ¹

- ¹ Tecnológico de Monterrey, Escuela de Ingeniería y Ciencias, Av. Epigmenio González 500, Fracc. San Pablo, Queretaro 76130, Mexico; a01208772@itesm.mx (J.G.-G.); ecuanurqui@tec.mx (E.C.-U.); arturo.escobedo@tec.mx (J.A.E.C.)
- ² Center for Engineering and Industrial Development-CIDESI, Energy Division, Queretaro 76125, Mexico; tsalgado@cidesi.edu.mx
- * Correspondence: agomeze@tec.mx (A.G.-E.); ggarcia@cidesi.edu.mx (L.G.G.-V); Tel.: +52-442-238-3302 (A.G.-E.)

Abstract: Several control strategies have been proposed for the trajectory tracking problem of Autonomous Underwater Vehicles (AUV). Most of them are model-based, hence, detailed knowledge of the parameters of the robot is needed. Few works consider a finite-time convergence in their controllers, which offers strong robustness and fast convergence compared with asymptotic or exponential solutions. Those finite-time controllers do not permit the users to predefine the convergence time, which can be useful for a more efficient use of the robot's energy. This paper presents the experimental validation of a model-free high-order Sliding Mode Controller (SMC) with finite-time convergence in a predefined time. The convergence time is introduced by the simple change of a time-base parameter. The aim is to validate the controller so it can be implemented for cooperative missions where the communication is limited or null. Results showed that the proposed controller can drive the robot to the desired depth and heading trajectories in the predefined time for all the cases, reducing the error by up to 75% and 41% when compared with a PID and the same SMC with asymptotic convergence. The energy consumption was reduced 35% and 50% when compared with those same controllers.

Keywords: AUV; SMC; finite-time; trajectory tracking



Citation: González-García, J.; Gómez-Espinosa, A.; García-Valdovinos, L.G.; Salgado-Jiménez, T.; Cuan-Urquiza, E.; Escobedo Cabello, J.A. Experimental Validation of a Model-Free High-Order Sliding Mode Controller with Finite-Time Convergence for Trajectory Tracking of Autonomous Underwater Vehicles. *Sensors* **2022**, *22*, 488. <https://doi.org/10.3390/s22020488>

Academic Editor: Enrico Meli

Received: 16 November 2021

Accepted: 4 January 2022

Published: 9 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Autonomous Underwater Vehicles (AUV) have permitted us to deepen the knowledge of the oceans and seafloors. The use of AUVs in tasks such as structural inspection [1,2], environmental risks detection [3], and mapping underwater structures [4], among others, increases the safety of the mission and the reliability of their results [5], and reduces the operational costs considerably [6]. However, due the challenging conditions in the underwater environment, a high precision autonomous navigation is difficult to achieve [7]. Autonomous navigation can be performed by different methods such as waypoint tracking, path following, and trajectory tracking [8]. In the waypoint tracking method, the vehicle navigates throughout a set of pre-defined waypoints. This is the easier method to be implemented, however, it could result in some uncertainty in the trajectory followed by the vehicle between two waypoints [9]. The path following navigation ensures that the vehicle follows a desired path to reach its destination, which is geometrically defined in Cartesian coordinates [10]. In the trajectory tracking problem, the vehicle must follow a time-parametrized path, meaning it must go to a certain point in a given time [11]. The integration of time and space restrictions makes trajectory tracking the most complex of

the referred autonomous navigation methods. Considering this, the highly non-linear dynamics of the AUV and the presence of external disturbances demands the development of new control methods to fulfill the extremely challenging problem of trajectory tracking in AUVs.

Proportional Integrate Derivative (PID) control was extensively used at the early stages of AUV tracking control [12–14]. PID controllers have a simple structure that allows an easier implementation. They are tuned to deal with very specific conditions and any change on them will cause a performance drop. Additionally, they do not consider non-linearities, which are highly present in the AUVs dynamics and the underwater environment. An alternative to deal with traditional controllers' issues is to include an intelligent algorithm [15,16] in the control law to achieve an adjustment of the controller parameters or self-tuning. This is known as adaptive control, which does not require an exact model of the vehicle and only a small amount of prior knowledge suffices. Backstepping control [17] and Sliding Mode Control (SMC) [18] are other common control methods widely used in the trajectory tracking of AUVs. The control methods used for AUVs trajectory tracking discussed before have an asymptotic or exponential convergence to the trajectory. A finite-time convergence controller would have many advantages such as faster convergence time, higher accuracy, and better anti-disturbance capability [19].

Related Work

Limited literature can be found related to control methods with finite-time convergence for AUV trajectory tracking. Ramezani-al et al. [20] proposed an SMC with an adaptive gain to eliminate the effects of external chattering and noise vulnerability. In numerical simulations, the proposed controller drives the vehicle to the desired trajectory in a limited time. Yu et al. [21] achieved finite-time convergence trajectory tracking of AUVs in presence of model parameter perturbation and ocean currents. The controller is based on a PID-SMC combination that globally stabilizes all tracking errors in a finite-time. Numerical simulations showed absolute tracking errors up to 20 cm in the absence of external disturbances. Qiao et al. [22] used two adaptive integral terminal SMC schemes to achieve trajectory tracking on AUVs. This controller yields finite-time convergence while dealing with dynamic uncertainties and external disturbances. The performance of the proposal was tested by numerical simulations resulting in robustness of 2% in position tracking. In those works, the controlled trajectory converges to the reference in a time that depends either on the vehicle parameters, its initial position, or the own controller parameters and gains. Therefore, prior knowledge of the vehicle is required to estimate the convergence time, and a different initial position and changes on the controller parameters will affect it. González-García et al. [23] presented a model-free high-order SMC with finite-time convergence aiming for cooperative AUVs missions. On contrary to previous works, the finite-time for the convergence can be predefined by the user through the time-base parameter t_b , the value of which—in seconds—is only limited by the thruster's capabilities. The arbitrary setting of the time-base parameter does not affect the robustness and performance of the controller once it meets the desired trajectory and can be used to demand a faster or slower response from the vehicle depending on what the mission requires. Numerical simulations were performed and results showed an outstanding performance when compared with traditional controllers. Even in the presence of strong ocean currents, the error was zero in the predefined time in all the cases simulated.

All of the works discussed before were limited to numerical simulations. However, due to the limitations present in real experiments, such as sensor resolution, measurement rate, processor speed, and so, it is quite possible that a controller does not perform as expected when programmed in a real system. The performance of a model-free high-order SMC with finite-time convergence in a predefined time is validated by real experiments in this paper. Being able to arbitrarily select the convergence time for the controller opens the options for better management of the vehicle energy consumption. It would also be an advantage to coordinate a group of AUVs in cooperative tasks. The performance of the controller was

compared with a PID and an asymptotic 2nd-order SMC. The experiments were performed using the BlueROV2 robot which was modified to be able to follow predefined trajectories autonomously. The experiments consisted of a depth and heading control to follow a predefined sinusoidal trajectory and were carried out in a pool with 1 m depth

The rest of the document is structured as: Section 2 describes the generalities of underwater vehicles, the control algorithm to be validated, and the BlueROV2 vehicle configuration. Section 3 describes the experimental set up including the BlueROV2 modifications and controller's laws and parameters. Section 4 shows the results of the experiments performed and their discussion, and Section 5 contains the concluding remarks.

2. Materials and Methods

General equations for kinematics and hydrodynamics of the AUVs are introduced in this section along with the particularities of the robot used in the experiments. After that, the model-free high-order SMC with finite-time convergence in predefined time is also introduced.

2.1. Autonomous Underwater Vehicles Kinematics and Hydrodynamics

Fossen [24] describes the kinematics of underwater vehicles using two reference frames: an Earth-fixed frame and a Body-fixed frame. As shown in Figure 1, the orthonormal axes are denoted as x , y , and z for the Earth-fixed frame and x_b , y_b , and z_b for the Body-fixed frame.

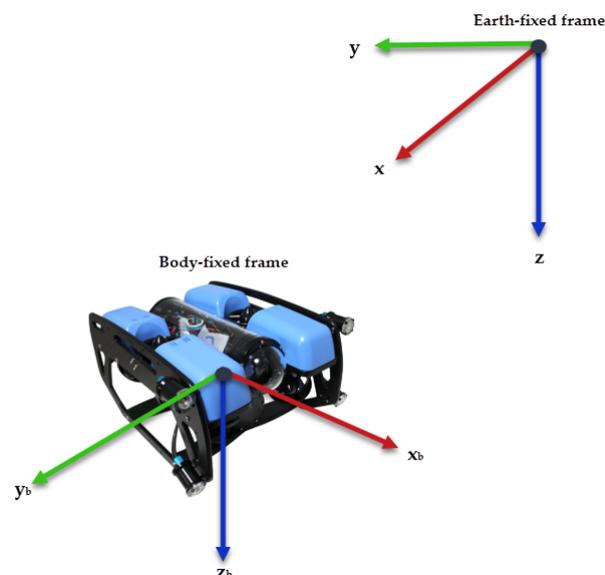


Figure 1. Reference frames for an AUV.

There is a convention for the notation of underwater vehicle's position, orientation, and velocities accepted by the Society of Naval Architects and Marine Engineers (SNAME) [24], which also defines the forces and moments applied to the vehicle. Considering this nomenclature, the vehicle's position regarding the Earth-fixed frame (η) and its velocities regarding the Body-fixed frame (v) can be described as

$$\eta = (x, y, z, \phi, \theta, \psi)^T, \quad (1)$$

$$v = (u, v, w, p, q, r)^T, \quad (2)$$

and the forces and moments with respect to the Body-fixed frame (τ) as

$$\tau = (X, Y, Z, K, M, N)^T. \quad (3)$$

The hydrodynamical model for underwater vehicles, as described by Newton–Euler equations [24], is given by

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau + \omega, \quad (4)$$

$$\tau = B_t u_t, \quad (5)$$

where

$M \in \mathbb{R}^{6 \times 6}$ is the inertial and added mass matrix,
 $C \in \mathbb{R}^{6 \times 6}$ is the rigid body and added mass centripetal and Coriolis matrix,
 $D \in \mathbb{R}^{6 \times 6}$ is the hydrodynamic damping matrix,
 $g \in \mathbb{R}^{6 \times 1}$ is the restitution forces vector,
 $\tau \in \mathbb{R}^{6 \times 1}$ is the control signal vector,
 $B_t \in \mathbb{R}^{6 \times 6}$ is the thruster allocation matrix,
 $u_t \in \mathbb{R}^{6 \times 1}$ is a vector containing the force generated by the thrusters, and
 $\omega \in \mathbb{R}^{6 \times 6}$ represents environmental disturbances.

According to Fossen [24], the model given by Equation (4) can be expressed in the Earth-fixed frame applying kinematic transformations as follows

$$\dot{\eta} = J(\eta_2)v \longleftrightarrow v = J^{-1}(\eta_2)\dot{\eta}, \quad (6)$$

$$\ddot{\eta} = J(\eta_2)\dot{v} + \dot{J}(\eta_2)v \longleftrightarrow \dot{v} = J^{-1}(\eta_2)[\ddot{\eta} - \dot{J}(\eta_2)v], \quad (7)$$

where

$$J(\eta_2) = \begin{bmatrix} J_1(\eta_2) & 0_{3 \times 3} \\ 0_{3 \times 3} & J_2(\eta_2) \end{bmatrix}, \quad (8)$$

with

$$J_1(\eta_2) = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + s_\phi s_\theta c_\psi & s_\psi s_\phi + s_\theta c_\psi c_\phi \\ s_\psi c_\phi & c_\psi c_\phi + s_\phi s_\theta s_\psi & -c_\psi s_\phi + s_\theta s_\psi c_\phi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}, \quad (9)$$

$$J_2(\eta_2) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix}, \text{ and} \quad (10)$$

$$\eta_2 = [\phi, \theta, \psi], \quad (11)$$

c_{angle} , s_{angle} , and t_{angle} are abbreviations for $\cos(angle)$, $\sin(angle)$, and $\tan(angle)$, respectively.

After some mathematical manipulation, the resulting expression for the hydrodynamical model in the Earth-fixed frame is

$$M_\eta(\eta)\ddot{\eta} + C_\eta(v, \eta)\dot{\eta} + D_\eta(v, \eta)\dot{\eta} + g_\eta(\eta) = \tau_\eta, \quad (12)$$

where

$$M_\eta(\eta) = J^{-T}(\eta)MJ^{-1}(\eta), \quad (13)$$

$$C_\eta(v, \eta) = J^{-T}(\eta)[C(v) - MJ^{-1}(\eta)\dot{J}(\eta)]J^{-1}(\eta), \quad (14)$$

$$D_\eta(v, \eta) = J^{-T}(\eta)D(v)J^{-1}(\eta), \quad (15)$$

$$g_\eta(\eta) = J^{-T}(\eta)g(\eta), \quad (16)$$

$$\tau_\eta(\eta) = J^{-T}(\eta)\tau. \quad (17)$$

2.2. Finite-Time Controller with Convergence in a Predefined Time

The High-order SMC with finite-time convergence used in this work is based in the proposal by García-Valdovinos et al. [25–27] and was adapted for trajectory tracking of AUVs in [23] as follows:

Consider that Equation (12) is linearly parametrizable by the product of a regressor $Y(\eta, \dot{\eta}, \ddot{\eta}) \in \mathbb{R}^{n \times p}$, which is composed of known nonlinear functions and a vector $\theta \in \mathbb{R}^p$ with constant parameters. The parametrization of Equation (12) can be rewritten in terms of a nominal reference $\dot{\eta}_r$ and its derivative $\ddot{\eta}_r$ as

$$M_\eta(\eta)\ddot{\eta}_r + C_\eta(v, \eta)\dot{\eta}_r + D_\eta(v, \eta)S_r = \tau_\eta - Y(\eta, \dot{\eta}, \ddot{\eta})\theta. \quad (18)$$

Subtracting Equation (18) from both sides of Equation (12) leads to the open-loop error dynamics expression

$$M_\eta(\eta)\dot{S}_r + C_\eta(v, \eta)S_r + D_\eta(v, \eta)S_r = \tau_\eta - Y(\eta, \dot{\eta}, \ddot{\eta})\theta, \quad (19)$$

where S_r is known as the extended error computed as

$$S_r = \dot{\eta} - \dot{\eta}_r, \quad (20)$$

considering the nominal reference

$$\dot{\eta}_r = \dot{\eta}_d - \alpha\tilde{\eta} + S_d - K_i \int_0^t \text{sign}(S_\eta)d\sigma, \quad (21)$$

where the tracking error of the position is given by $\tilde{\eta} = \eta - \eta_d$, η_d is the desired position, α and K_i are diagonal positive definite $n \times n$ gain matrices, $\text{sign}(x)$ is the signum function of the x vector, and

$$S = \dot{\tilde{\eta}} + \alpha\tilde{\eta}, \quad (22)$$

$$S_d = S(t_0)e^{-kt}, \quad (23)$$

$$S_\eta = S - S_d, \quad (24)$$

with $k > 0$. The extended error can be now rewritten as

$$S_r = S_\eta + K_i \int_0^t \text{sign}(S_\eta)d\sigma, \quad (25)$$

resulting in the model-free second-order SMC which control law is defined as

$$\tau_\eta = -K_d S_r. \quad (26)$$

A stability analysis of the controller just described can be found in [27]. To achieve finite-time convergence with this model-free high-order SMC, a Time-Base Generator (TBG) is used as described in [25,26]. This TBG is a scalar function providing a smooth 0 to 1 transition. The duration of that transition is controllable by a predefined time-base t_b . The derivative of this function provides a bell-shaped speed profile. The functions used for the TBG are

$$\zeta(t) = 10 \frac{(t-t_0)^3}{(t_b-t_0)^3} - 15 \frac{(t-t_0)^4}{(t_b-t_0)^4} + 6 \frac{(t-t_0)^5}{(t_b-t_0)^5} \quad (27)$$

and

$$\dot{\zeta}(t) = 30 \frac{(t-t_0)^2}{(t_b-t_0)^3} - 60 \frac{(t-t_0)^3}{(t_b-t_0)^4} + 30 \frac{(t-t_0)^4}{(t_b-t_0)^5}, \quad (28)$$

where t_0 represents the initial time. Equations (27) and (28) are subjected to the following conditions: $\zeta(t_0) = \dot{\zeta}(t_0) = \zeta(t_b) = 0$. Then, $\zeta(t)$ is used to parameterize the gain α in Equation (22) as time-varying as follows

$$\alpha(t) = \begin{cases} \alpha_0 \frac{\xi}{(1-\xi)+\delta} & 0 \leq t \leq t_b, \\ \alpha_c & t > t_b \end{cases}, \tag{29}$$

with $\alpha_0 = 1 + \epsilon$, $0 < \epsilon \ll 1$, $0 < \delta \ll 1$, and $\alpha_c > 0$. A diagram of the complete control scheme is showed in Figure 2.

Remark 1. The time-base parameter t_b can be chosen arbitrarily and does not depends on the initial conditions or controller parameters. The convergence of the controlled trajectory will occur at the selected t_b , which means that a smaller value for t_b will produce a faster convergence, and a bigger t_b value will produce a slower convergence.

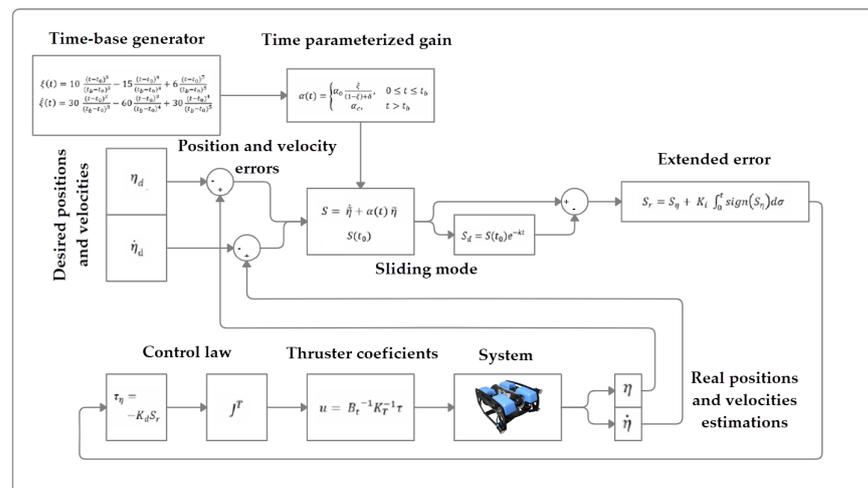


Figure 2. Model-free high-order SMC with finite time convergence block diagram.

2.3. BlueROV2 Robot

The BlueROV2 from Blue Robotics® is a Remotely Operated Vehicle (ROV) with a 6-Thruster vectored configuration as shown in Figure 3. It has an open-source software and hardware configuration, which make it suitable for research purposes.

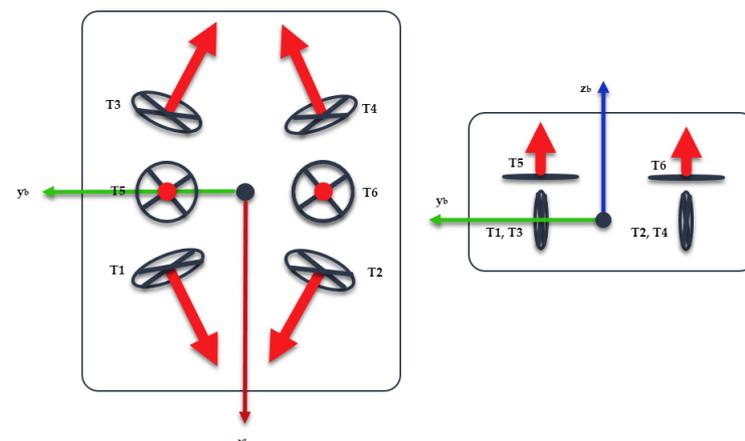


Figure 3. BlueROV2 thruster configuration. (Left) Top view. (Right) Front view.

A complete set of the BlueROV2 physic and hydrodynamic parameters is listed in [23]. However, this work considers a model-free control algorithm, which means that only the thruster allocation matrix B_f is needed. From the BlueROV2 thruster distribution, B_f is defined as

$$B_t = \begin{bmatrix} 0.7071 & 0.7071 & -0.7071 & -0.7071 & 0 & 0 \\ -0.7071 & 0.7071 & -0.7071 & 0.7071 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0.115 & -0.115 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.1773 & 0.1773 & -0.1773 & 0.1773 & 0 & 0 \end{bmatrix}. \quad (30)$$

2.4. Exact Differentiator

The tracking error derivative is needed to compute the control signal. That means that the velocities of the robot need to be estimated. The easiest way to do that is through a simple Euler differentiator. However, this differentiator is quite sensitive to noise and will produce an inaccurate estimation of the robot velocities. An alternative is to use an exact differentiator as the proposed by Davila et al. [28].

3. Experimental Set-Up

The BlueROV2 was modified in its hardware and software to be able to autonomously follow predefined depth and yaw trajectories. The proposed model-free high-order SMC with finite-time convergence was tested in different experiments and its performance was compared with traditional PID and 2nd-order SMC. The experimental set-up can be observed in the Figure 4.

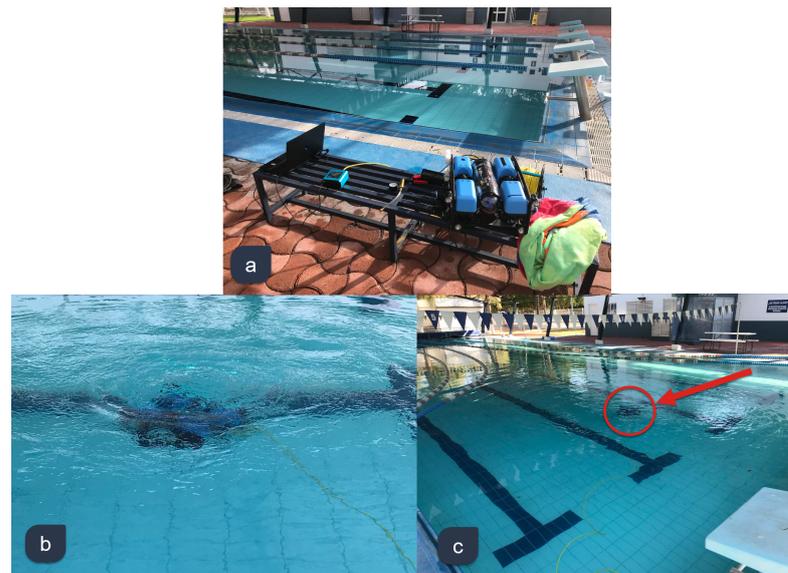


Figure 4. Experimental set-up at Tecnológico de Monterrey, Campus Querétaro. (a) Ground control station at a side of the semi-Olympic pool. (b) BlueROV2 deployed into the water. (c) Autonomous trajectory tracking mission.

The experiments were performed in the Tecnológico de Monterrey, Campus Queretaro semi-olympic swimming pool. All the runs were made in the same robot—BlueROV2—using three full-charged batteries—one for every control scheme—from 9:00 a.m. to 12:00 p.m. This with the aim of perform the experiments under the same conditions. No external disturbances were introduced to the experiments. An OMEN O15 Laptop with a Intel CORE i7 processor running UBUNTU 16 as operative system was used as ground control station.

3.1. Hardware

Inside the BlueROV2, a Raspberry Pi[®] 3 (RPi) acts as the processor, and it is in charge of running the control algorithms and managing the different sensors. The Rpi runs Lubuntu as operative system, and it is connected to a control station through a tether. A BAR-30

high-resolution pressure sensor from BlueRobotics[®] is connected to the robot through an I²C port in the RPi to measure the depth z of the vehicle. A Smart sensor BNO-055 from Bosh[®] is connected to a serial port of the Rpi and is used to estimate the orientations ϕ, θ, ψ . The thruster's speeds are controlled by Pulse Wide Modulation (PWM) signals from the RPi, which goes through a set of 30 A Electronic Speed Controllers (ESC). Finally, a 14.8 V, 18A Ah battery supplies energy for the whole electronics. A diagram of the hardware configuration in the BlueROV2 used for the experiments is shown in Figure 5.

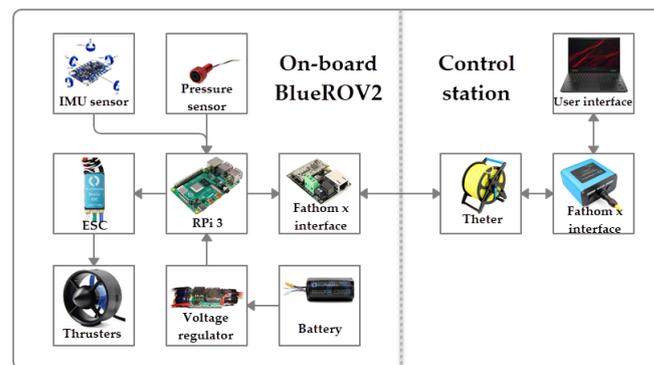


Figure 5. BlueROV2 hardware configuration.

3.2. Software

The Robot Operating System (ROS) in its Kinetic version was used to program the algorithms to manage the sensors, thruster's speed, and control schemes. Five nodes were used in total (shown in Figure 6).

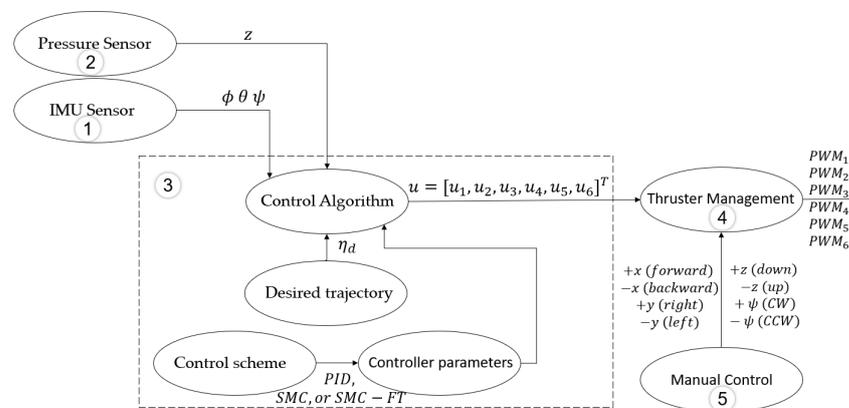


Figure 6. BlueROV2[®] software configuration.

The *IMU sensor* node (1) manages the BNO-055 sensor and it is able to obtain a new set of ϕ, θ , and ψ orientations at a 100 Hz rate. The *Pressure Sensor* node (2) manages the Bar-30 sensor and obtains a new z position at a 10 Hz rate. Both nodes publish the estimated positions so they can be used as inputs to the *Control Algorithm* node (3). In this node, the different control algorithms were programmed along with a user interface to provide the user the option of selecting between the PID control, 2nd-Order SMC, and the model-free high-order SMC with finite-time convergence. The user can also modify the controller parameters and gains from this interface. In the algorithm, the desired position is generated as defined by the user throughout the user interface—either as a set point or as time-variant—and compared with the actual position estimated by the sensors. Then, the selected control law is applied and a thruster's coefficients vector is computed and published as a result. The *Thruster Management* node (4) reads the thruster's coefficients vector and converts it to the proper PWM signal for each thruster to be generated by the

RPI peripherals. The *Manual Control* node (5) is used prior to the experiment to position the BlueROV2 in its initial position.

3.3. Time-Parameterized Trajectories

A sinusoidal trajectory was chosen for the vehicle to track both depth (z_d), and yaw (ψ_d). Those trajectories are given by

$$z_d = z_i + A_z \sin(\omega t), \text{ and } \psi_d = \psi_i + A_\psi \sin(\omega t), \quad (31)$$

where z_i and ψ_i are the initial desired position or offset, A_z and A_ψ are the desired amplitudes, and ω is the frequency for the signal. Then, the desired velocities \dot{z}_d and $\dot{\psi}_d$ are

$$\dot{z}_d = -\omega A_z \cos(\omega t), \text{ and } \dot{\psi}_d = -\omega A_\psi \cos(\omega t). \quad (32)$$

3.4. Control Algorithms

Experiments were carried out using three different control laws for the trajectory tracking of the robot. The aim was to compare the performance of the proposed algorithm with other common controllers used for this purpose.

3.4.1. PID Control

The control law used for this controller is

$$\tau_\eta = -k_p \tilde{\eta} - k_i \int \tilde{\eta} dt - k_d \dot{\tilde{\eta}}, \quad (33)$$

where k_p , k_i , and k_d are the proportional, integral, and derivative gain matrices, respectively. For the experiments on this work, the gain matrices that gave the best results are

$$k_p = \text{diag}[0, 0, 100, 100, 0, 100], \quad k_i = \text{diag}[0, 0, 1, 1, 0, 1], \text{ and} \\ k_d = \text{diag}[0, 0, 80, 10, 0, 80]. \quad (34)$$

3.4.2. Model-Free High-Order SMC (Asymptotic)

The control law used for this controller is

$$\tau_\eta = -K_d S_r, \quad (35)$$

with

$$S_r = S_\eta + K_i \int_0^t \text{sign}(S_\eta) d\sigma, \quad S_\eta = S - S_d, \\ S_d = S(t_0) e^{-\kappa t}, \text{ and } S = \dot{\tilde{\eta}} + \alpha \tilde{\eta}. \quad (36)$$

where $\alpha > 0$, $\kappa > 0$, K_i , and K_d are constant gains, t_0 is the initial time, and $\text{sign}(S_\eta)$ is the sign function of S_η . For the experiments on this work, the gain set that gave the best results is

$$\alpha = 10, \quad \kappa = 5, \quad K_i = \text{diag}[0, 0, 0.05, 0.005, 0, 0.001], \text{ and} \\ K_d = \text{diag}[0, 0, 100, 0.1, 0, 0.1]. \quad (37)$$

3.4.3. Model-Free High-Order SMC (Finite-Time)

The control law used for this controller is the same as the previous section (Equation (35)). To achieve finite-time convergence, the α gain in Equation (36) is substituted by the time-variant gain $\alpha(t)$ defined in Equation (29), with $\alpha_c = 10$ for depth control, $\alpha_c = 5$ for heading control, and, in both cases, $\alpha_0 = 1.005$ and $\delta = 0.001$. The rest of gain parameters (κ , K_i , and K_d) stays as defined in Equation (37).

3.5. Exact Differentiator Algorithm

The algorithm proposed for the exact differentiator is given by

$$\begin{aligned} \dot{j}_0 &= w_0, \\ w_0 &= -\lambda_1(j_0 - f(t))^{1/2} \times \text{sign}(j_0 - f(t)) + j_1, \text{ and} \\ \dot{j}_1 &= -\lambda_2 \times \text{sign}(j_0 - f(t)), \end{aligned} \quad (38)$$

where $\lambda_1 = 1.5$ and $\lambda_2 = 1.1$ are constant, $f(t)$ is the signal we want to differentiate, and $\text{sign}()$ stands for the sign function. After a brief adjustment time, the following is considered as true

$$j_0 = f(t), \quad j_1 = \dot{f}(t), \quad (39)$$

$j_0 = 0$ and $j_1 = 0$ were considered as initial values for the algorithm.

4. Results and Discussion

The functions in Equations (31) and (32) were programmed in the BlueROV2 as desired trajectories for depth and heading, respectively, meanwhile the desired roll was set to $\phi = 0$. The initial desired positions were set to $z_i = 0.50$ m and $\psi_i = 270^\circ$, the desired amplitudes were $A_z = 0.25$ m and $A_\psi = 30^\circ$, and the period was set to $T = \frac{1}{\omega} = 20$ s. Results for different runs with the PID controller implementation are shown in Figure 7.

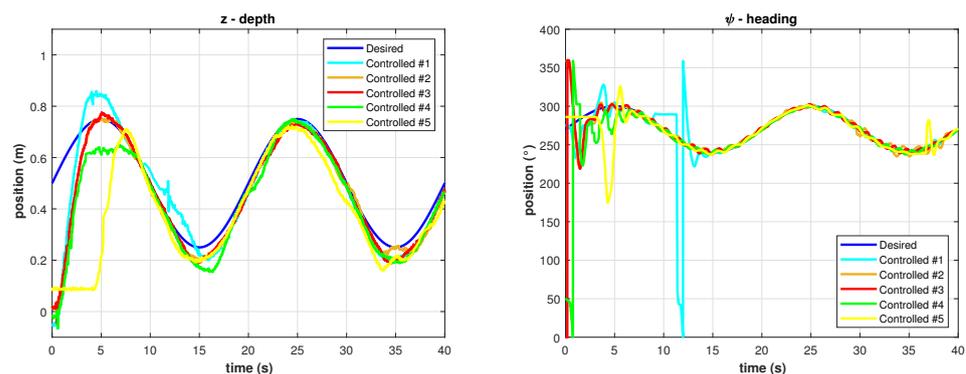


Figure 7. Controlled trajectories with the PID controller. (Left) Depth. (Right) Heading.

The best result for the PID controller is shown in Figure 8, a significant error can be observed in the depth trajectory control meanwhile the heading trajectory control error is significantly minor, but there is an overshoot in its transitory response.

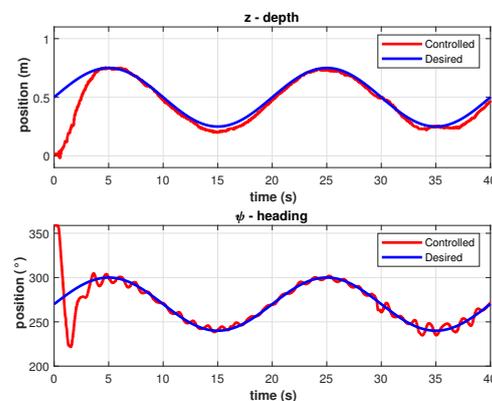


Figure 8. Controlled trajectories with the PID controller.

The same experiment was performed applying the model-free 2nd-order SMC with asymptotic convergence. Results for trajectory tracking of z and ψ in different runs of the

experiment are shown in Figure 9. The graph in Figure 10 includes the best performance of the asymptotic 2nd-order SMC, there is a faster response compared with the PID controller, and the error once the robot meets the trajectory is smaller.

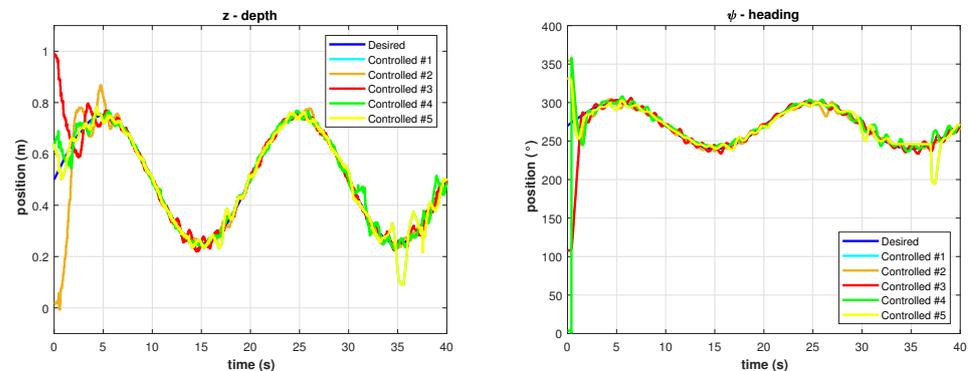


Figure 9. Controlled trajectories with the asymptotic model-free 2nd-order SMC. (Left) Depth. (Right) Heading.

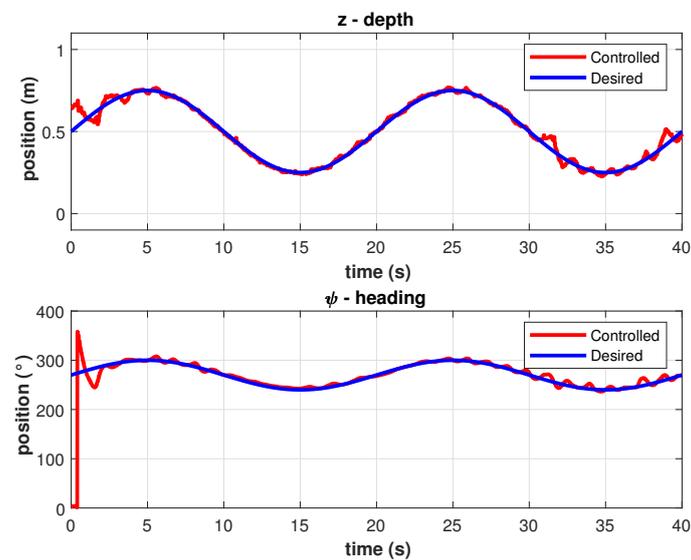


Figure 10. Best controlled trajectories with the asymptotic model-free 2nd-order SMC.

Results for the trajectory tracking by the model-free high-order SMC with finite-time convergence in a predefined time-base of 5 s are shown in Figure 11. Note that, at the beginning of the controlled trajectory, the BlueROV2 approaches slowly to the desired trajectory and then accelerates to reduce the error to practically zero in the selected time-base of 5 s. After the time-base is reached, the controller maintains the robot in the desired trajectory with a minimum Root Mean Square Error (RMSE) (<1 cm and $<2.3^\circ$) as can be seen in Figure 12. Those errors can be attributed to hardware limitations such as sensor accuracy, resolution, and measurement rates, processor speed, communication baud rates, etc., which limits the control cycle frequency to a maximum of 10 Hz compared with a frequency 100 to 1000 times higher in simulations.

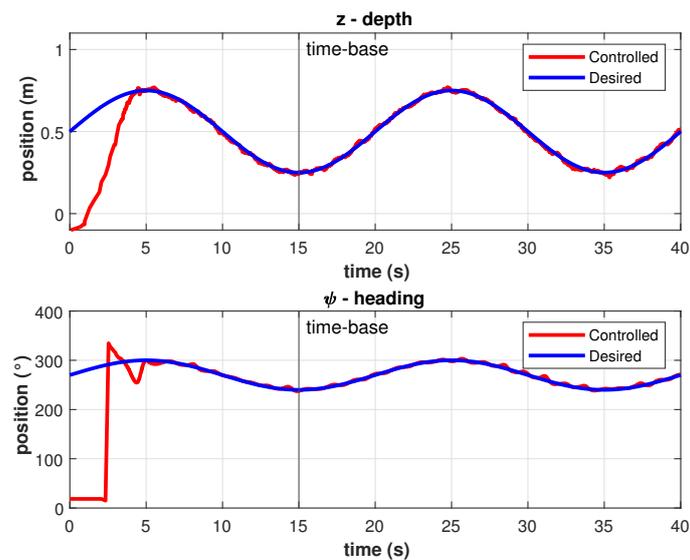


Figure 11. Controlled trajectory with the model-free high-order SMC with finite-time convergence in a 5 s predefined-time.

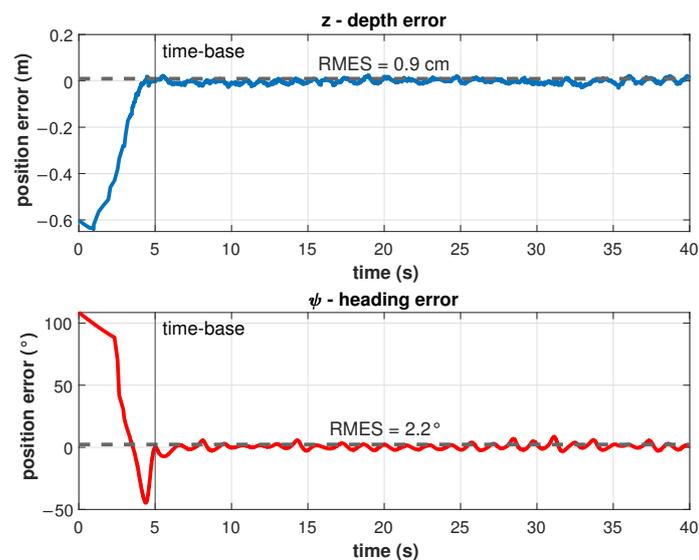


Figure 12. Tracking error for depth and heading control.

The resulting TBG for this experiment is shown in Figure 13. As described in Equations (27) and (28), there is a smooth transition from zero to one which is completed exactly at the given time-base. The resulting time-varying gain $\alpha(t)$ is shown in Figure 14, as described in Equation (29). Several runs of the same experiment were performed with the model-free high-order SMC with finite-time convergence defining different time-bases. Results of the controlled depth and yaw trajectories are shown in Figure 15. It can be observed that the controller drives the robot to the desired trajectory to match it in the desired time-base for all cases.

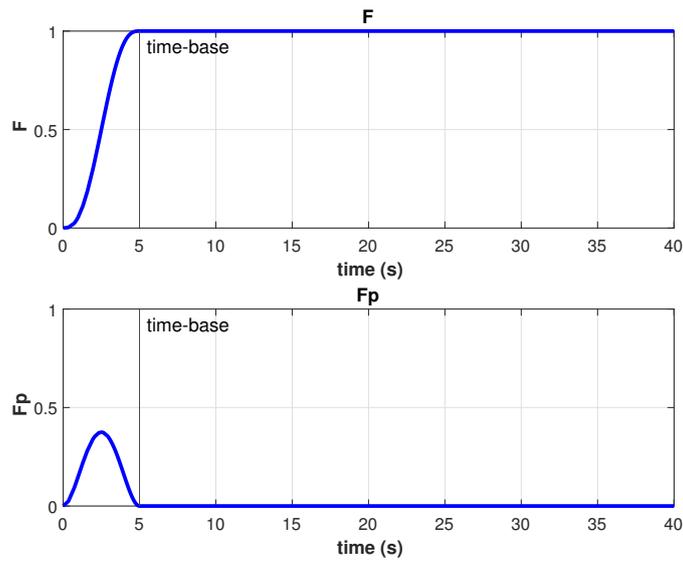


Figure 13. Time base generator results for $t_b = 5$ s.

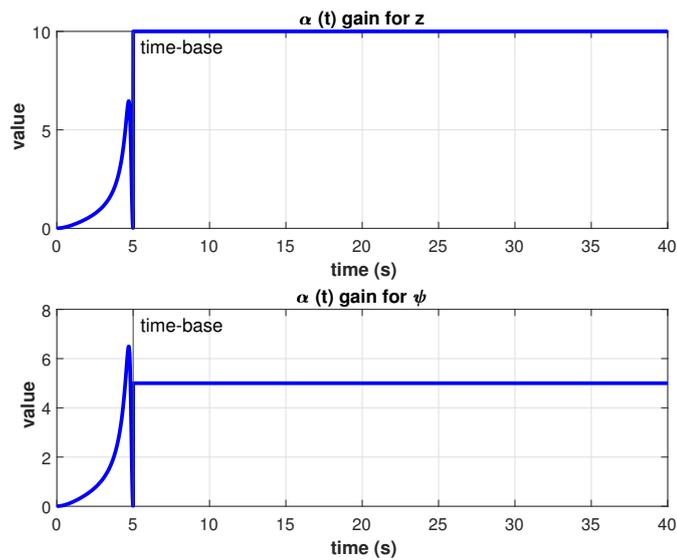


Figure 14. Time-varying gain $\alpha(t)$ result for $t_b = 5$ s.

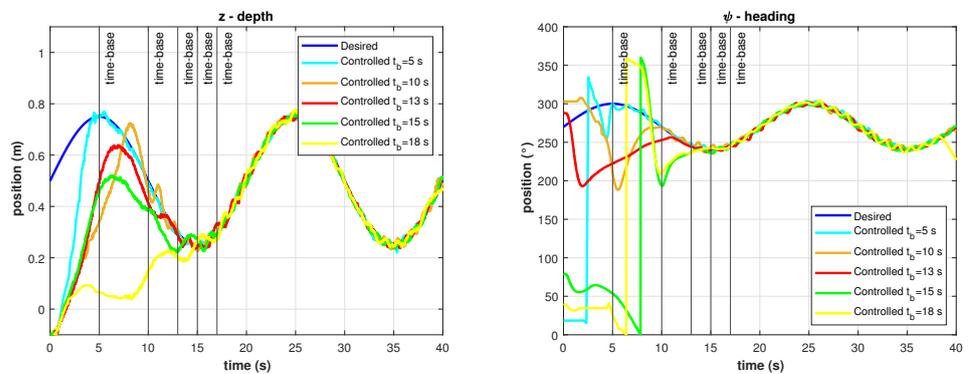


Figure 15. Controlled trajectories with the model-free high-order SMC with finite-time convergence in predefined-time. (Left) Depth. (Right) Heading.

Velocities estimations are another source of noise and, therefore, errors in the trajectory tracking problem for the SMC. To reduce these errors, the exact differentiator described in Equation (38) was used for the experiments in both the asymptotic and finite-time SMC. The results of the \dot{z} and $\dot{\psi}$ estimations for a finite-time controlled trajectory are shown in Figure 16. As can be observed, the uncertainty in the estimations was reduced significantly by the use of the exact differentiator.

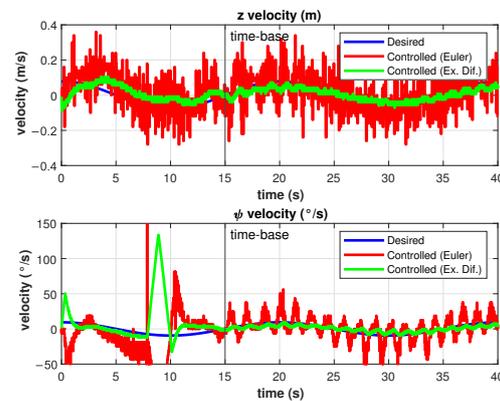


Figure 16. Velocities estimations by Euler and Exact differentiators.

The RMSE was computed as a performance indicator of the different controllers applied. As the error varies during the convergence, the RMSE was calculated after the robot is in the desired trajectory. For the finite-time controller, the robot is considered in the trajectory when $t > t_b$. For the PID and the asymptotic SMC controllers, the robot is considered in trajectory once the controlled trajectory intersects with the desired trajectory. Results for the different control approaches are shown in Figure 17.

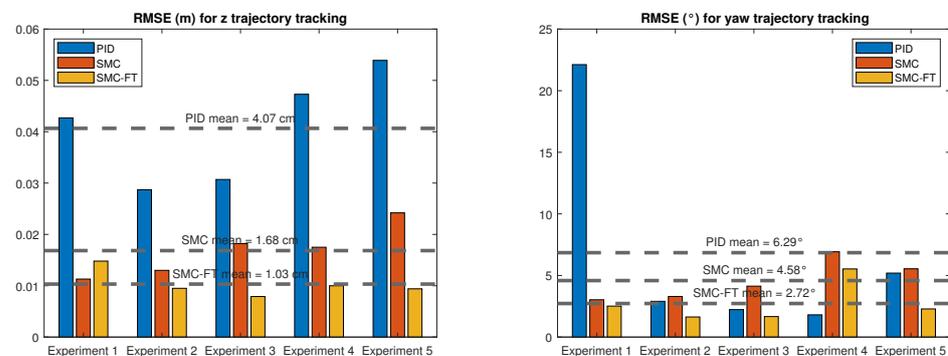


Figure 17. RMSE comparison for the trajectory tracking with the different controllers. (Left) Depth. (Right) Heading.

In the depth trajectory tracking, the mean RMSE—considering all the runs of the experiment—is 4.07 cm for the PID control (PID), 1.68 cm for the asymptotic 2nd-order SMC (SMC), and the best was the high-order SMC with finite-time convergence (SMC-FT) with a value of 1.03 cm. For the yaw trajectory tracking, the mean RMSE was 6.29° for the PID control, 4.58° for the asymptotic 2nd-order SMC, and, again, the best was the high-order SMC with finite-time convergence with a value of 2.72°.

Another performance indicator is the energy consumption in the thrusters. A control coefficient of +1.0 represents the maximum thrust in one direction and a control coefficient of −1.0 represents the maximum thrust in the opposite direction, in both cases, the power consumption is 625 W approximately. A higher magnitude of this coefficient means a bigger energy consumption from the corresponding thruster. The control coefficient for the

different thrusters as result of the experiments shown in Figures 8, 10 and 11 are shown in Figure 18.

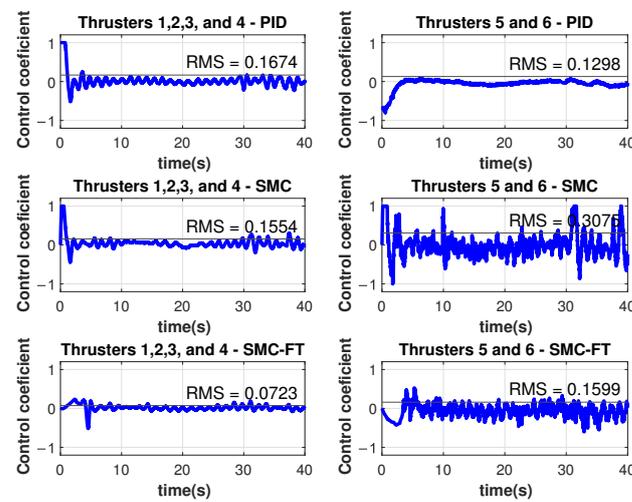


Figure 18. Control coefficients on the thrusters for the different control approaches.

The resulting control coefficients for thrusters 1, 2, 3, and 4 were identical in magnitude, the same occurred for thrusters 5 and 6. The Root Mean Square (RMS) values of the control coefficients were computed as performance indicator as follows

$$u_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N u_i^2}, \quad (40)$$

where N is the number of samples. Considering the six thrusters, the RMS value of the control coefficient was 0.155 for the PID controller, 0.206 for the asymptotic SMC, and 0.101 for the finite-time SMC with $t_b = 5$ s. The finite-time controller was able to drive the vehicle to the desired trajectory with 35 % less energy demanded than the PID controller, even when they reach the desired trajectory at the same time. The asymptotic SMC was faster in its approach to the desired trajectory, resulting in an energy demand twice as bigger as the demanded by the finite-time controller.

As can be observed in the different performance indicators, the finite-time SMC proposed outperforms the asymptotic SMC even when, as stated in previous sections, they were programmed with the same parameters and gains—except for $\alpha(t)$ —which is a clear example of the advantages of using a finite-time convergence solution. Performing a tuning on the PID or asymptotic SMC to achieve a faster or slower convergence will also modify the performance of the controller in the trajectory tracking. This does not happen with the finite-time controller, where a simple change in the t_b parameter will lead to a faster or slower convergence maintaining its performance and robustness. The user can benefit from this to obtain the best performance of the controller according to the needs demanded by the task.

5. Conclusions

In this work, a model-free high-order SMC with finite-time convergence in a predefined time was tested and validated through experimentation in a water pool, using the BlueROV2 robot. This robot was modified and instrumented to be able to follow time-parametrized trajectories at depth and orientation. Results showed that the finite-time convergence controller was able to drive the robot to the desired trajectory in a desired predefined time and maintain it there with a small error for all of the experiment runs. The performance of the controller was then compared with a PID controller and an asymptotic 2nd-order SMC in terms of the RMSE. The mean depth error after the total of the experiments was 75% smaller for the finite-time convergence controller compared with the PID

control. Additionally, it was 38% smaller compared with the asymptotic 2nd-order SMC which kept the same parameters and gains of the finite-time controller with the exception of the α gain. Regarding the yaw trajectory tracking, the same indicator was 57% and 41% smaller for the finite-time controller when compared to the PID and asymptotic SMC, respectively. Although it was expected that the finite-time controller outruns the PID control performance, the fact that it also outperformed its asymptotic equivalent demonstrates the advantages of a finite-time convergence controller in the AUV navigation problem. Another advantage of the use of this finite-time controller is the smaller energy consumption. The RMS thruster coefficient was 35% smaller when compared with the PID solution with both controllers converging at the same time. It was also half of the RMS coefficient computed for the asymptotic SMC. Future work will expand the experimentation to x and y axes. Additionally, this controller will be implemented and tested for cooperative intervention missions of AUVs, where it is important that the vehicles can reach a desired position or trajectory at a predefined time. This controller will be especially useful considering scenarios where the communication between AUVs is limited or does not exist at all.

Author Contributions: Conceptualization, J.G.-G., L.G.G.-V., T.S.-J. and A.G.-E.; methodology, J.G.-G. and L.G.G.-V.; investigation, J.G.-G.; software, J.G.-G.; writing—original draft preparation, J.G.-G.; visualization, J.G.-G.; writing—review and editing, J.G.-G., L.G.G.-V., T.S.-J., A.G.-E., E.C.-U. and J.A.E.C.; supervision, L.G.G.-V., T.S.-J. and A.G.-E.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article.

Acknowledgments: The authors would like to acknowledge the financial support of Tecnológico de Monterrey, in the production of this work. The authors would like to acknowledge support from CONACyT for PhD studies of first author.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AUV	Autonomous Underwater Vehicles
DoF	Degrees Of Freedom
ESC	Electronic Speed Controllers
PID	Proportional Integrate Derivative
PWM	Pulse Wide Modulation
RMS	Root Mean Square
RMSE	Root Mean Square Error
ROV	Remotely Operated Vehicle
ROS	Robot Operating System
RPi	Raspberry Pi
SMC	Sliding Mode Control
SNAME	Society of Naval Architects and Marine Engineers
TBG	Time-Base Generator

References

1. Palomer, A.; Ridaio, P.; Ribas, D. Inspection of an underwater structure using point-cloud SLAM with an AUV and a laser scanner. *J. Field Robot.* **2019**, *36*, 1333–1344. [[CrossRef](#)]
2. Rumson, A.G. The application of fully unmanned robotic systems for inspection of subsea pipelines. *Ocean Eng.* **2021**, *235*, 109214. [[CrossRef](#)]

3. Hwang, J.; Bose, N.; Nguyen, H.D.; Williams, G. Acoustic search and detection of oil plumes using an autonomous underwater vehicle. *J. Mar. Sci. Eng.* **2020**, *8*, 618. [[CrossRef](#)]
4. Hernández, J.D.; Istenič, K.; Gracias, N.; Palomeras, N.; Campos, R.; Vidal, E.; García, R.; Carreras, M. Autonomous underwater navigation and optical mapping in unknown natural environments. *Sensors* **2016**, *16*, 1174. [[CrossRef](#)] [[PubMed](#)]
5. Sánchez, P.J.B.; Papaelias, M.; Márquez, F.P.G. Autonomous underwater vehicles: Instrumentation and measurements. *IEEE Instrum. Meas. Mag.* **2020**, *23*, 105–114. [[CrossRef](#)]
6. Simetti, E. Autonomous Underwater Intervention. *Curr. Robot. Rep.* **2020**, *1*, 117–122. [[CrossRef](#)]
7. González-García, J.; Gómez-Espinosa, A.; Cuan-Urquizo, E.; García-Valdovinos, L.G.; Salgado-Jiménez, T.; Escobedo Cabello, J.A. Autonomous underwater vehicles: Localization, navigation, and communication for collaborative missions. *Appl. Sci.* **2020**, *10*, 1256. [[CrossRef](#)]
8. Kim, E.; Fan, S.; Bose, N.; Nguyen, H. Path Following for an Autonomous Underwater Vehicle (AUV) by Using a High-Gain Observer based on an AUV Dynamic Model. *IFAC-PapersOnLine* **2019**, *52*, 218–223. [[CrossRef](#)]
9. Elmokadem, T.; Zribi, M.; Youcef-Toumi, K. Control for Dynamic Positioning and Way-point Tracking of Underactuated Autonomous Underwater Vehicles Using Sliding Mode Control. *J. Intell. Robot. Syst. Theory Appl.* **2019**, *95*, 1113–1132. [[CrossRef](#)]
10. Li, Y.; qing Jiang, Y.; feng Wang, L.; Cao, J.; cheng Zhang, G. Intelligent PID guidance control for AUV path tracking. *J. Cent. South Univ.* **2015**, *22*, 3440–3449. [[CrossRef](#)]
11. Li, D.; Du, L. Auv trajectory tracking models and control strategies: A review. *J. Mar. Sci. Eng.* **2021**, *9*, 1020. [[CrossRef](#)]
12. Gan, W.Y.; Zhu, D.Q.; Xu, W.L.; Sun, B. Survey of trajectory tracking control of autonomous underwater vehicles. *J. Mar. Sci. Technol.* **2017**, *25*, 722–731. [[CrossRef](#)]
13. Guerrero, J.; Torres, J.; Creuze, V.; Chemori, A.; Campos, E. Saturation based nonlinear PID control for underwater vehicles: Design, stability analysis and experiments. *Mechatronics* **2019**, *61*, 96–105. [[CrossRef](#)]
14. Qiao, L.; Yi, B.; Wu, D.; Zhang, W. Design of three exponentially convergent robust controllers for the trajectory tracking of autonomous underwater vehicles. *Ocean Eng.* **2017**, *134*, 157–172. [[CrossRef](#)]
15. Rezazadegan, F.; Shojaei, K.; Sheikholeslam, F.; Chatraei, A. A novel approach to 6-DOF adaptive trajectory tracking control of an AUV in the presence of parameter uncertainties. *Ocean Eng.* **2015**, *107*, 246–258. [[CrossRef](#)]
16. Yu, C.; Xiang, X.; Zhang, Q.; Xu, G. Adaptive Fuzzy Trajectory Tracking Control of an Under-Actuated Autonomous Underwater Vehicle Subject to Actuator Saturation. *Int. J. Fuzzy Syst.* **2018**, *20*, 269–279. [[CrossRef](#)]
17. Cho, G.R.; Li, J.H.; Park, D.; Jung, J.H. Robust trajectory tracking of autonomous underwater vehicles using back-stepping control and time delay estimation. *Ocean Eng.* **2020**, *201*, 107131. [[CrossRef](#)]
18. García-Valdovinos, L.G.; Fonseca-Navarro, F.; Aizpuru-Zinkunegi, J.; Salgado-Jiménez, T.; Gómez-Espinosa, A.; Cruz-Ledesma, J.A. Neuro-Sliding Control for Underwater ROV's Subject to Unknown Disturbances. *Sensors* **2019**, *19*, 2943. [[CrossRef](#)]
19. Qin, H.; Li, C.; Sun, Y.; Li, X.; Du, Y.; Deng, Z. Finite-time trajectory tracking control of unmanned surface vessel with error constraints and input saturations. *J. Frankl. Inst.* **2020**, *357*, 11472–11495. [[CrossRef](#)]
20. Ramezani-al, M.R.; Tavanaei Sereshki, Z. A novel adaptive sliding mode controller design for tracking problem of an AUV in the horizontal plane. *Int. J. Dyn. Control* **2019**, *7*, 679–689. [[CrossRef](#)]
21. Yu, H.; Guo, C.; Yan, Z. Globally finite-time stable three-dimensional trajectory-tracking control of underactuated UUVs. *Ocean Eng.* **2019**, *189*, 106329. [[CrossRef](#)]
22. Qiao, L.; Zhang, W. Double-Loop Integral Terminal Sliding Mode Tracking Control for UUVs with Adaptive Dynamic Compensation of Uncertainties and Disturbances. *IEEE J. Ocean Eng.* **2019**, *44*, 29–53. [[CrossRef](#)]
23. González-García, J.; Narcizo-Nuci, N.A.; García-Valdovinos, L.G.; Salgado-Jiménez, T.; Gómez-Espinosa, A.; Cuan-Urquizo, E.; Cabello, J.A.E. Model-free high order sliding mode control with finite-time tracking for unmanned underwater vehicles. *Appl. Sci.* **2021**, *11*, 1836. [[CrossRef](#)]
24. Fossen, T.I. *Handbook of Marine Craft Hydrodynamics and Motion Control*; John Wiley & Sons, Ltd.: Chichester, UK, 2011. [[CrossRef](#)]
25. Garcia-Valdovinos, L.G.; Parra-Vega, V.; Mendez-Iglesias, J.A.; Arteaga, M.A. Cartesian sliding PID force/position control for transparent bilateral teleoperation. *IECON Proc. (Ind. Electron. Conf.)* **2005**, *2005*, 1979–1985. [[CrossRef](#)]
26. García-Valdovinos, L.G.; Parra-Vega, V.; Arteaga, M.A. Bilateral Cartesian sliding PID force/position control for tracking in finite time of master-slave systems. *Proc. Am. Control Conf.* **2006**, *2006*, 369–375. [[CrossRef](#)]
27. García-Valdovinos, L.G.; Salgado-Jiménez, T.; Bandala-Sánchez, M.; Nava-Balanzar, L.; Hernández-Alvarado, R.; Cruz-Ledesma, J.A. Modelling, Design and Robust Control of a Remotely Operated Underwater Vehicle. *Int. J. Adv. Robot. Syst.* **2014**, *11*, 1. [[CrossRef](#)]
28. Davila, J.; Fridman, L.; Levant, A. Second-order sliding-mode observer for mechanical systems. *IEEE Trans. Autom. Control* **2005**, *50*, 1785–1789. [[CrossRef](#)]