

Article

Modeling and Control of a Micro AUV: Objects Follower Approach

Jesus Arturo Monroy-Anieva ^{1,2,*}, Cyril Rouviere ³, Eduardo Campos-Mercado ⁴,
Tomas Salgado-Jimenez ^{1,†} and Luis Govinda Garcia-Valdovinos ^{1,†}

¹ Energy Division, Center for Engineering and Industrial Development-CIDESI, Santiago de Queretaro, Queretaro 76125, Mexico; tsalgado@cidesi.edu.mx (T.S.-J.); ggarcia@cidesi.edu.mx (L.G.G.-V.)

² Automation and Control Department, Universidad Politecnica de Tulancingo-UPT, Tulancingo 43629, Hidalgo, Mexico

³ Ecole Nationale Supérieure de Techniques Avancées-ENSTA, Bretagne, Brest 29806, France; pi@cyril-rouviere.fr

⁴ CONACYT, Universidad del Istmo de Tehuantepec-UNISTMO, Tehuantepec 70760, Oaxaca, Mexico; ecampos@conacyt.mx

* Correspondence: jesus.monroy@upt.edu.mx or monroyanieva@gmail.com

† Current address: Av. Playa Pie de la Cuesta No. 702, Desarrollo San Pablo, Santiago de Queretaro, Queretaro 76125, Mexico.

Received: 14 June 2018; Accepted: 19 July 2018; Published: 6 August 2018



Abstract: This work describes the modeling, control and development of a low cost Micro Autonomous Underwater Vehicle (μ -AUV), named AR2D2. The main objective of this work is to make the vehicle to detect and follow an object with defined color by means of the readings of a depth sensor and the information provided by an artificial vision system. A nonlinear PD (Proportional-Derivative) controller is implemented on the vehicle in order to stabilize the heave and surge movements. A formal stability proof of the closed-loop system using Lyapunov's theory is given. Furthermore, the performance of the μ -AUV is validated through numerical simulations in MatLab and real-time experiments.

Keywords: submarine robot; modeling; control; embedded system; micro vehicle; nonlinear PD; stability; computer vision

1. Introduction

Nowadays, the development of autonomous underwater systems is a scientific research topic which deserves a great interest from the international community. There are a huge number of possible applications wherein autonomous underwater vehicles can be used to avoid risky tasks for the human beings. The proper operation of autonomous vehicles requires the convergence of several disciplines among which the automatic control discipline may play an important role when dealing with complex nonlinear dynamics as in the case of AUVs (Autonomous Underwater Vehicles) [1]. The AUVs have traditionally been used for oceanographic research, where these vehicles have been used for mapping and monitoring specific areas; for example, 3D reefs reconstruction, marine cartography, ecosystem exploration, pipeline and structures inspection, etc. [2]. There are other water-based applications which take advantage of the use of AUVs, such as the monitoring of nuclear storage ponds, waste water treatment facilities and archaeology enclosures. These environments differ from the ocean in terms of scale, thus they can be considered as closed spaces [3]. Traditional AUVs have tended to be large scale (meters in length) and high cost, making them unsuitable for small scale environments. One interesting topic from the automatic control point of view is that very often the vertical and horizontal movements are coupled, i.e., forward momentum is required for the AUV

to move in the vertical plane. This means that traditional AUV designs are unsuitable for being used as small-scale sensor platforms [4–6]. The AR2D2 prototype (see Figure 1) has major features, such as its size and maneuverability compared to traditional vehicles.

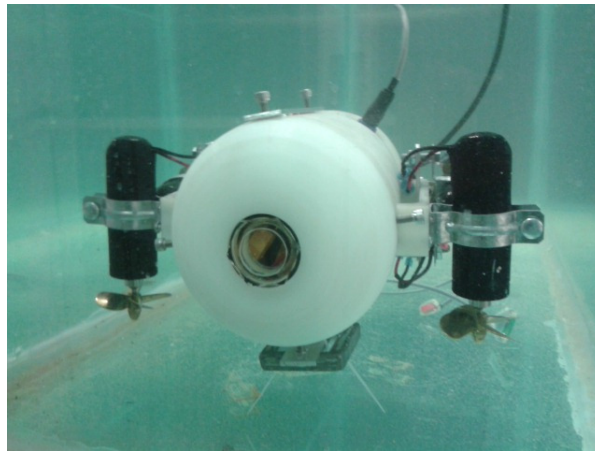


Figure 1. Experimental prototype AR2D2.

The μ -AUV prototype has the shape of a torpedo with the following dimensions: 35 cm \times 20 cm front and 30 cm long. Its weight is less than 5 kg, thus this vehicle is considered like a micro AUV [7]. The different AUV's degrees of freedom are actuated by four thrusters. The geometric shape of the prototype is such that the vertical and horizontal movements can be considered decoupled.

This paper provides details about the platform development and real-time test of a control strategy for heave and surge movements to follow objects. The effectiveness of the proposed control strategy is validated through numerical simulations as well as real-time experiments. Section 2 presents further details on the design and capabilities of the μ -AUV. In Section 3, we briefly describe the dynamic model of the vehicle, whilst Section 4 shows the control strategy for the depth and forward motion through a nonlinear PD controller. Section 5 details the computer vision algorithm developed in order to detect and follow objects, whilst Section 6 presents the simulation of the control law chosen for depth and forward movement, and the experimental results. Finally, concluding remarks and further work proposals are given in Section 7.

2. Prototype Description

2.1. Embedded System

The hardware architecture of the prototype consists of an embedded system that includes a Raspberry Pi 2B, which has a quad-core ARM (Advanced RISC Machines) Cortex-A7 CPU of 900 MHz, 1 GB RAM memory, camera interface (CSI), display interface (DSI), four USB ports, Ethernet port, Full HDMI port, 40 GPIO pins, and a slot to use a micro SD card for an operative system based on a Linux platform that allows for programming the Middleware. This embedded system also includes a compass sensor (CMPS10), an ultrasonic sensor with the driver SRM400, a depth sensor (BMP085) and a Raspberry Pi camera for artificial vision. The Raspberry processes the information from the sensors, computes de control laws and sends signals to the actuators (DC motors) controlled by Pulse Width Modulation using a micro maestro device (six-channel USB) in order to manage four thrusters through the Robbe Rookie 25A ESC (Electronic Speed Control) motor drivers. Real-time communication is provided using an Ethernet LAN interface. In Figure 2, a schematic shows components of the vehicle's hardware and their interactions.

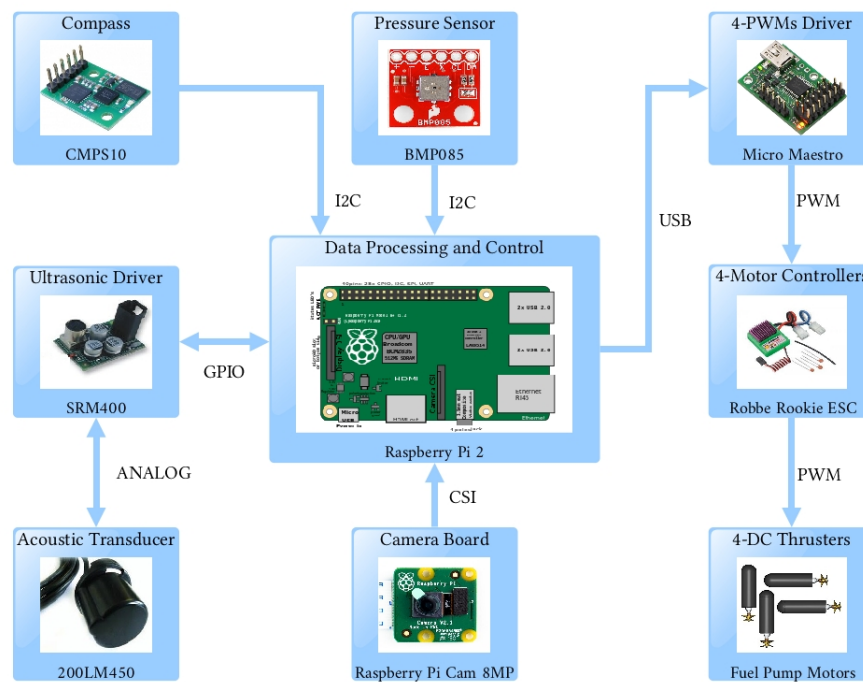


Figure 2. Electronic architecture of the embedded system.

2.2. Computer Vision

The GNSS (Global Navigation Satellite System) signals cannot go through water, acoustic communication throughput is limited, odometry is impossible. However, vision is available. The camera provides rich data about the environment. Moreover, in the case of a Remote Operated Vehicle (ROV), it shows to the user a lot of information to help him drive the robot. Computer Vision is a way to process these data autonomously, in order to implement obstacle avoidance, object following recognition and following, localization and mapping, without any human intervention.

However, as other sensors, underwater vision poses some particular issues to be solved with respect to usual air vision. The main problem is about picture quality, depending on the water (clearness, suspended particles, etc.), the camera will be able to see less or more details. These details are the key to success for computer vision algorithms, especially dealing with features-based vision. Moreover, many features that the camera will detect are not reliable because they are not static (usually, particles floating in water, bubbles, etc.) Thus, we cannot rely on stereo-vision, optical flow and common mapping, which need a strong feature recognition.

Sometimes, pictures quality is so bad that even a human is not able to use it. Colors often do not have enough contrasts, which does not help to detect and differentiate objects that are not close enough to the robot. Finally, the camera can be reliable only as a low-range sensor, which is not adapted for global navigation.

In fact, underwater vision is problematic only if it tries to mimic air vision. Indeed, what is seen as an issue can be turned to our advantage to have better vision quality than expected before.

Classic computer vision complains about the lack of details under water. However, it then has to filter all of these data to select only what is interesting. With an underwater vision point of view, this heavy filtering is not necessary: it is already done by the environment. Only three colors are visible in water: red, orange and yellow; all others appear as blue. This is why objects we want to detect usually have these colors. Thus, to detect them, color filtering is enough (shape and geometry does not matter).

To get around the problem of low contrast level, HSV (Hue–Saturation–Value) filtering can give good results. A lamp also helps to keep the same luminosity exposition (and avoid moving shadows). It is also a way to measure distance to the object: the better luminosity and contrast are, the closer it is.

2.3. Prototype's Movement Description

The AR2D2 micro submarine has been designed and built, which is shown in Figure 3, with its body fixed frame (O_b, x_b, y_b, z_b). The center (O_b) of this frame corresponds to the vehicle's center of gravity, and its axes are aligned with the main axes of vehicle's symmetry. The movement in the horizontal plane is referred to as *surge* (along x_b axis) and *sway* (along y_b axis), while *heave* represents the vertical motion (along z_b axis). Roll, pitch, and yaw, denoted (ϕ, θ, ψ) , are the Euler angles describing the orientation of the vehicle's body fixed frame with respect to the earth-fixed frame (O_I, x_I, y_I, z_I) [8], while (x, y, z) denote the coordinates of the body-fixed frame center in the earth fixed frame. The propulsion system consists of four thrusters that generate the rotational and translational motion. Concerning the rotational movement of this prototype, roll motion is performed through differential speed control of thrusters 1 and 2. In the same fashion, yaw motion is obtained using thrusters 3 and 4; finally, pitch motion is unactuated with respect to the translational movement of the z axis being regulated by decreasing or increasing the combined speed of thrusters 1 and 2. In the same way, the translational movements along the x_b - and y_b -axes are obtained by using thrusters 3 and 4 and by controlling the yaw angle.

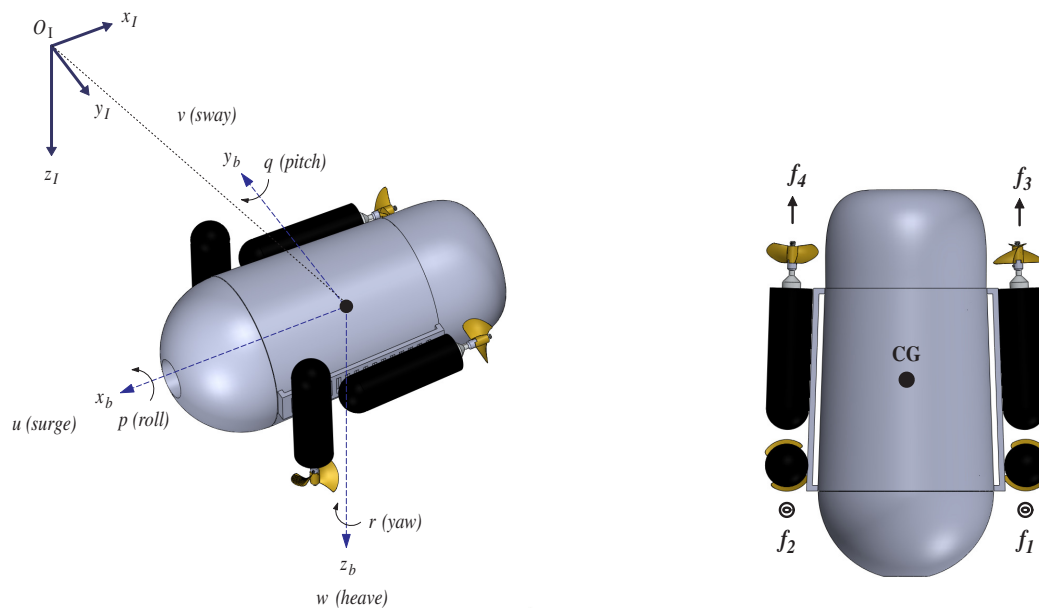


Figure 3. The AR2D2 μ AUV, with the body fixed frame (O_b, x_b, y_b, z_b), the earth-fixed frame (O_I, x_I, y_I, z_I), and the forces generated by its four thrusters.

3. Dynamic Model

The dynamics of the vehicle that are expressed in the body-fixed frame can be written in a vectorial setting according to [8]:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \mathbf{w}_e, \quad (1)$$

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

where $\mathbf{M} \in \mathbb{R}^{6 \times 6}$ is the inertial matrix, $\mathbf{C}(\mathbf{v}) \in \mathbb{R}^{6 \times 6}$ defines the Coriolis-centripetal matrix, $\mathbf{D}(\mathbf{v}) \in \mathbb{R}^{6 \times 6}$ represents the hydrodynamic damping matrix, $\mathbf{g}(\boldsymbol{\eta}) \in \mathbb{R}^{6 \times 1}$ describes the vector of gravitational/buoyancy forces and moments, $\boldsymbol{\tau} = [\tau_1, \tau_2]^T = [[X, Y, Z], [K, M, N]]^T \in \mathbb{R}^{6 \times 1}$

defines the vector of control inputs; $\mathbf{w}_e \in \mathbb{R}^{6 \times 1}$ defines the vector of disturbances; $\mathbf{v} = [\nu_1, \nu_2]^T = [[u, v, w], [p, q, r]]^T \in \mathbb{R}^{6 \times 1}$ denotes the linear and angular velocity vector in the body-fixed frame; $\boldsymbol{\eta} = [\eta_1, \eta_2]^T = [[x, y, z], [\phi, \theta, \psi]]^T \in \mathbb{R}^{6 \times 1}$ is the position and attitude vector decomposed in the earth-fixed frame, and $\mathbf{J}(\boldsymbol{\eta}) \in \mathbb{R}^{6 \times 6}$ is the transformation matrix between body frame and earth-fixed frame (for more details, see [9,10]).

3.1. Gravity/Buoyancy Forces and Torques

According to Archimedes's principle, the buoyant force \mathbf{f}_B applied in the center of buoyancy, which acts on the opposite direction of vehicle weight \mathbf{f}_W , is expressed as follows:

$$\mathbf{f}_B = - \begin{bmatrix} 0 \\ 0 \\ \rho g \nabla \end{bmatrix} \quad \mathbf{f}_W = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}, \quad (3)$$

where ρ represents the fluid density, g the gravitational acceleration, ∇ the displaced fluid volume and m the mass of the vehicle. Now, considering that $W = mg$ and $B = \rho g \nabla$ by using the zyx -convention for navigation and control application [11], the transformation matrix $J_1(\boldsymbol{\eta}_2) = R_{z,\psi}^T R_{y,\theta}^T R_{x,\phi}^T$ is applied in order to obtain the weight and buoyancy forces respect to the body fixed coordinates system:

$$F_B = J_1(\boldsymbol{\eta}_2)^{-1} f_B \quad F_W = J_1(\boldsymbol{\eta}_2)^{-1} f_W; \quad (4)$$

consequently,

$$\mathbf{F}_B = \begin{bmatrix} B \sin(\theta) \\ -B \cos(\theta) \sin(\phi) \\ -B \cos(\theta) \cos(\phi) \end{bmatrix} \quad \mathbf{F}_W = \begin{bmatrix} -W \sin(\theta) \\ W \cos(\theta) \sin(\phi) \\ W \cos(\theta) \cos(\phi) \end{bmatrix}. \quad (5)$$

Thus, the restoring forces acting on the vehicle are $\mathbf{f}_g = \mathbf{F}_B + \mathbf{F}_W$, this is:

$$\mathbf{f}_g = \begin{bmatrix} (B - W) \sin(\theta) \\ (W - B) \cos(\theta) \sin(\phi) \\ (W - B) \cos(\theta) \cos(\phi) \end{bmatrix}. \quad (6)$$

On the other hand, the restoring moments are described by the following equation:

$$\mathbf{m}_g = \mathbf{r}_w \times \mathbf{F}_W + \mathbf{r}_b \times \mathbf{F}_B, \quad (7)$$

where $\mathbf{r}_w = [x_w, y_w, z_w]^T$ and $\mathbf{r}_b = [x_b, y_b, z_b]^T$ represent the positions of the center of gravity (CG) and the center of buoyancy (CB), respectively. Based on the design of the vehicle and in order to reduce further analysis, the origin of the body fixed frame is chosen in the gravity's center; this implies that $\mathbf{r}_w = [0, 0, 0]^T$, while the center of buoyancy is $\mathbf{r}_b = [0, 0, -z_b]^T$. For practical purposes, the buoyancy force is greater than the weight, i.e., $W - B = -f_b$. Notice that f_b should be smaller than the force produced by the thrusters. Then, from Equations (6) and (7), we have:

$$g(\boldsymbol{\eta}) = \begin{bmatrix} \mathbf{f}_g \\ \mathbf{m}_g \end{bmatrix} = \begin{bmatrix} f_b \sin(\theta) \\ -f_b \cos(\theta) \sin(\phi) \\ -f_b \cos(\theta) \cos(\phi) \\ -z_b B \cos(\theta) \sin(\phi) \\ -z_b B \sin(\theta) \\ 0 \end{bmatrix}. \quad (8)$$

3.2. Forces and Torques Generated by the Thrusters

Figure 3 shows the forces generated by the thrusters acting on the micro submarine. These are described relative to the body-fixed coordinate system as:

$$\hat{\mathbf{f}}_1 = \begin{bmatrix} 0 \\ 0 \\ f_1 \end{bmatrix}; \hat{\mathbf{f}}_2 = \begin{bmatrix} 0 \\ 0 \\ f_2 \end{bmatrix}; \hat{\mathbf{f}}_3 = \begin{bmatrix} f_3 \\ 0 \\ 0 \end{bmatrix}; \hat{\mathbf{f}}_4 = \begin{bmatrix} f_4 \\ 0 \\ 0 \end{bmatrix}.$$

Summarizing and using the notation of [12], it follows that:

$$\tau_1 = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} f_3 + f_4 \\ 0 \\ f_1 + f_2 \end{bmatrix} \quad (9)$$

and the body-fixed torques generated by the above forces are defined as:

$$\tau_2 = \sum_{i=1}^4 \mathbf{l}_i \times \hat{\mathbf{f}}_i, \quad (10)$$

where $\mathbf{l}_i = (l_{ix}, l_{iy}, l_{iz})$ is the position vector of the force $\hat{\mathbf{f}}_i \forall i = 1, \dots, 4$, with respect to the body-fixed reference frame. Then, the torques generated by the thrusters are described as:

$$\tau_2 = \begin{bmatrix} K \\ M \\ N \end{bmatrix} = \begin{bmatrix} l_{1y}f_1 + l_{2y}f_2 \\ l_{1x}f_1 + l_{2x}f_2 \\ l_{3y}f_3 + l_{4y}f_4 \end{bmatrix}. \quad (11)$$

Then,

$$\tau = \begin{bmatrix} f_3 + f_4 \\ 0 \\ f_1 + f_2 \\ l_{1y}f_1 + l_{2y}f_2 \\ l_{1x}f_1 + l_{2x}f_2 \\ l_{3y}f_3 + l_{4y}f_4 \end{bmatrix}. \quad (12)$$

4. Control Strategy

For the design of the controller, it is common to assume that the hydrodynamic parameters involved in the dynamical model of the underwater vehicle are unknown. Indeed, they depend on effects and properties that are hard to model or estimate, like added mass, skin friction, vortex shedding, fluid characteristics, etc. Therefore, we propose using a nonlinear PD controller [13].

$u(t)$ is a PD controller, which is depicted as the following equation:

$$u(t) = K_P e(t) + K_D \frac{de(t)}{dt}, \quad (13)$$

where $e(t) = r(t) - y(t)$ is the error, $r(t)$ represents the reference, $y(t)$ is the output measurement, and (K_P, K_D) are the proportional and derivative gains. In Equation (13), we can notice that, if $e(t) \rightarrow \infty$, then $u(t) \rightarrow \infty$; this could lead to system oscillations or in other case saturation of the actuators. In order to prevent damage on the actuators, we propose using a saturation function in each term of Equation (13).

Now, let $\sigma_{\bar{b}_i}(k_i h_i)$ be a saturation function $\forall i = 1, 2$, and $\bar{b}_i, k_i > 0$, described in Figure 4 and defined as the next equation:

$$\sigma_{\bar{b}_i}(k_i h_i) = \begin{cases} \bar{b}_i, & \text{if } k_i h_i > \bar{b}_i, \\ k_i h_i, & \text{if } |k_i h_i| \leq \bar{b}_i, \\ -\bar{b}_i, & \text{if } k_i h_i < -\bar{b}_i. \end{cases} \quad (14)$$

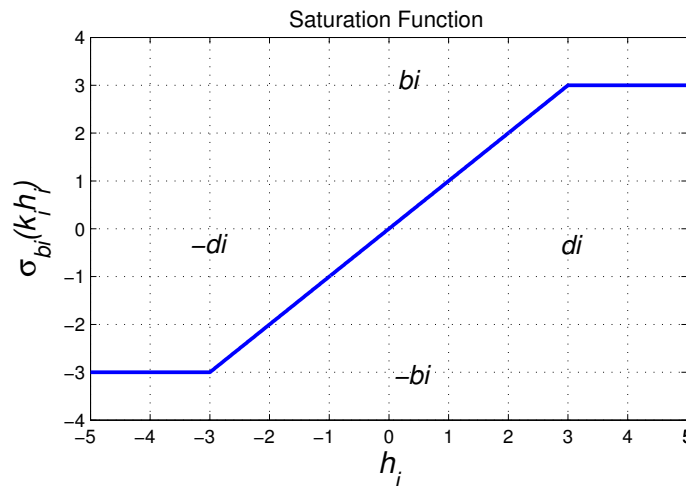


Figure 4. Saturated function.

4.1. A Nonlinear PD Controller Based on Saturation Functions

According to Equations (13) and (14), we propose a nonlinear PD controller based on saturation functions as follows:

$$u(t) = \sigma_{\bar{b}_1}[k_1 e(t)] + \sigma_{\bar{b}_2}\left[k_2 \frac{de(t)}{dt}\right]. \quad (15)$$

The above equation can be represented as:

$$u(t) = \sum_{i=1}^2 u_i, \quad (16)$$

where $u_i = \sigma_{\bar{b}_i}(k_i h_i)$ represents the saturation function, h_1 is the error and h_2 is the derivative error. Then, according to Equation (14), we have:

$$u_i = \sigma_{\bar{b}_i}(k_i h_i) = \begin{cases} \bar{b}_i & \text{if } k_i h_i > \bar{b}_i, \\ k_i h_i & \text{if } |k_i h_i| \leq \bar{b}_i, \\ -\bar{b}_i & \text{if } k_i h_i < -\bar{b}_i, \end{cases} \quad (17)$$

and, for the previous equation, we can rewrite it as:

$$u_i = \begin{cases} \text{sign}(h_i) \bar{b}_i, & \text{if } |k_i h_i| > \bar{b}_i, \\ k_i h_i, & \text{if } |k_i h_i| \leq \bar{b}_i. \end{cases} \quad (18)$$

In Equation (18), we can notice that the parameters tuning of the controller, which is described by Equation (15), are the gains k_i and the saturation values \bar{b}_i , $\forall i = 1, 2$. Notice that the parameters tuning could be the saturation values b_i and the interval of h_i for which u_i is lineal, thus we can choose the

value of h_i for which we want to saturate the law control. As a consequence, we are going to introduce a new parameter. For this, we consider the point of h_i where $|u_i| = \bar{b}_i$; this is:

$$|u_i| = |k_i h_i| = \bar{b}_i \implies |h_i| = \bar{b}_i / k_i. \quad (19)$$

Then, we define:

$$d_i := \bar{b}_i / k_i \quad (20)$$

as the point where:

$$u_i = \text{sign}(h_i) \bar{b}_i \quad \forall \quad |h_i| > d_i. \quad (21)$$

According to Equations (20) and (21), we can represent the control law, which is given by Equation (18), in terms of the parameters b_i and d_i as follows:

$$u_i = \begin{cases} \text{sign}(h_i) \bar{b}_i, & \text{if } |h_i| > d_i, \\ \bar{b}_i d_i^{-1} h_i, & \text{if } |h_i| \leq d_i, \end{cases} \quad (22)$$

where the parameters tuning of the controller are b_i and d_i , $\forall i = 1, 2$. In order that Equation (22) will be expressed in terms of h_i when $|h_i| > d_i$, we have that:

$$\text{sign}(h_i) \bar{b}_i = h_i \text{sign}(h_i) \bar{b}_i h_i^{-1}. \quad (23)$$

Then,

$$\text{sign}(h_i) \bar{b}_i = |h_i| \bar{b}_i h_i^{-1}. \quad (24)$$

Considering that $|h_i| h_i^{-1} = |h_i|^{-1} h_i$, then Equation (22) can be rewritten as:

$$u_i = \begin{cases} \bar{b}_i |h_i|^{-1} h_i & \text{if } |h_i| > d_i, \\ \bar{b}_i d_i^{-1} h_i & \text{if } |h_i| \leq d_i. \end{cases} \quad (25)$$

Finally, the control law defined by Equation (15) can be represented as:

$$u(t) = u_1 + u_2 = k_p(e)e(t) + k_d(\dot{e})\dot{e}(t), \quad (26)$$

with:

$$k_p(e) = \begin{cases} \bar{b}_1 |e(t)|^{-1} & \text{if } |e(t)| > d_1, \\ \bar{b}_1 d_1^{-1} & \text{if } |e(t)| \leq d_1, \end{cases} \quad (27)$$

$$k_d(\dot{e}) = \begin{cases} \bar{b}_2 |\dot{e}(t)|^{-1} & \text{if } |\dot{e}(t)| > d_2, \\ \bar{b}_2 d_2^{-1} & \text{if } |\dot{e}(t)| \leq d_2. \end{cases} \quad (28)$$

The advantage of this controller is that the maximum forces and torques are chosen by the parameters \bar{b}_1 and \bar{b}_2 . Thus, we are sure that the actuators will not be damaged, but, in other cases, it is necessary that the forces and torques are slightly larger to correct the system error.

4.2. Stability Proof

Considering Equations (1) and (2), we propose the following control input:

$$\tau = g(\eta) - J^T(\eta) \tau_{NPD} \quad (29)$$

with:

$$\tau_{NPD} = \sigma_{\bar{b}_p} [k_p e(t)] + \sigma_{\bar{b}_d} [k_d(\dot{e}(t))]. \quad (30)$$

The equation before can be rewritten as:

$$\tau_{NPD} = K_p(\cdot)e(t) + K_d(\cdot)\dot{e}(t), \quad (31)$$

with:

$$K_p(\cdot) = \begin{bmatrix} K_{p1}(\cdot) & 0 & \cdots & 0 \\ 0 & K_{p2}(\cdot) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K_{pn}(\cdot) \end{bmatrix} > 0, \quad (32)$$

$$K_d(\cdot) = \begin{bmatrix} K_{d1}(\cdot) & 0 & \cdots & 0 \\ 0 & K_{d2}(\cdot) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K_{dn}(\cdot) \end{bmatrix} > 0, \quad (33)$$

where:

$$k_{p_i} = \begin{cases} \bar{b}_{p_i} |e_i(t)|^{-1} & \text{if } |e_i(t)| > d_{p_i}, \\ \bar{b}_{p_i} d_{p_i}^{-1} & \text{if } |e_i(t)| \leq d_{p_i}, \end{cases} \quad (34)$$

$$k_{d_i} = \begin{cases} \bar{b}_{d_i} |\dot{e}_i(t)|^{-1} & \text{if } |\dot{e}_i(t)| > d_{d_i}, \\ \bar{b}_{d_i} d_{d_i}^{-1} & \text{if } |\dot{e}_i(t)| \leq d_{d_i}. \end{cases} \quad (35)$$

Considering the regulation case:

$$\eta_d = cte \implies \dot{\eta} = 0. \quad (36)$$

Assuming that:

$$e = \eta - \eta_d \implies \dot{e} = \dot{\eta}, \quad (37)$$

then Equation (29) can be rewritten as:

$$\tau = g(\eta) - J^T(\eta) [K_p(e)e + K_d(\dot{e})\dot{\eta}]. \quad (38)$$

Now, we have the closed-loop system as follows:

$$M\dot{v} + C(v)v + D(v)v = -J^T(\eta) [K_p(e)e + K_d(\dot{e})\dot{\eta}]. \quad (39)$$

Considering Equation (2), we have:

$$M\dot{v} + C(v)v + D(v)v = -J^T(\eta) [K_p(e)e + K_d(\dot{e})J(\eta)v]. \quad (40)$$

We define $K_{dd}(\eta, \dot{e}) = J^T(\eta)K_d(\dot{e})J(\eta)$; then, the previous equation can be rewritten as:

$$M\dot{v} + C(v)v + D(v)v = -J^T(\eta)K_p(e)e - K_{dd}(\eta, \dot{e})v. \quad (41)$$

Rewriting the previous equation, we have:

$$\frac{d}{dt} \begin{bmatrix} e \\ v \end{bmatrix} = \quad (42)$$

$$\begin{bmatrix} J(\eta)v \\ M^{-1} [J^T(\eta)K_p(e)e - K_{dd}(\eta, \dot{e})v - C(v)v - D(v)v] \end{bmatrix}. \quad (43)$$

Observe that the unique origin is the equilibrium point. Now, we can propose the following Lyapunov function candidate:

$$V(e, v) = \frac{1}{2} v^T M v + \int_0^e \xi^T K_p(\xi) d\xi. \quad (44)$$

According to Lemma 2 from [14], we have that $V(e, v)$ is globally positive definite and a radially unbounded function. The time derivative of the Lyapunov function candidate is:

$$\dot{V}(e, v) = v^T M \dot{v} - e^T K_p(e) J(\eta) v. \quad (45)$$

By substituting the closed-loop Equation (41) into (45), we obtain:

$$\dot{V}(e, v) = v^T J^T(\eta) K_p(e) e - v^T K_{dd}(\eta, \dot{e}) v - v^T C(v) v - v^T D(v) v - e^T K_p(e) J(\eta) v. \quad (46)$$

Since $K_p(e) = K_p^T(e)$ and $C(v) = -C(v)^T$, Equation (46) becomes:

$$\dot{V}(e, v) = -v^T [K_{dd}(\eta, \dot{e}) + D(v)] v. \quad (47)$$

Assuming that $D(v) > 0$ and remembering that $K_d > 0 \rightarrow K_{dd} > 0$ and symmetric matrix, we then obtain that $\dot{V}(e, v)$ is a globally negative semidefinite function, and therefore we conclude stability of the equilibrium point. In order to prove asymptotic stability, we apply the Krasovskii–LaSalle’s theorem; then:

$$\Omega = \left\{ \begin{bmatrix} e \\ v \end{bmatrix} = \dot{V}(e, v) = 0 \right\} = \left\{ \begin{bmatrix} e \\ v \end{bmatrix} = \begin{bmatrix} e \\ 0 \end{bmatrix} \in R^{2n} \right\}. \quad (48)$$

Introducing $v = 0$ and $\dot{v} = 0$ into Equation (41), we have $e = 0$; therefore, we conclude that the equilibrium point is globally asymptotically stable.

5. Computer Vision Algorithm

5.1. Data Processing Chain

Processing an image requires many steps, each one narrowing the quantity of data until it gives what we need (see Figure 5).

Firstly, a raw image taken from the camera has to be unnoised thanks to a Gaussian blur. Then, only a pixel of wanted colors are kept, giving a binary image. At this step, the image gives a geometric representation of the interesting object, without any color considerations.

Secondly, the algorithm has to localize the object, in order to have coordinates, orientation, size and variables that can be easily manipulated. It detects all connected groups of pixels (called “blobs”), and returns their size and centroid position. The biggest is considered as the object that the robot has to follow.

Finally, depending on the status of the robot, it can decide to move or follow the detection of objects.

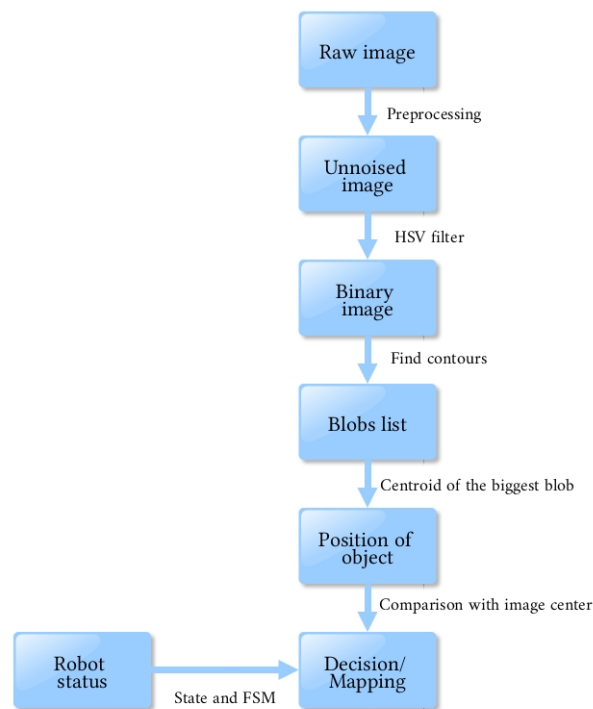


Figure 5. Data processing chain.

5.2. Image Preprocessing

Before trying to extract data from an image, we need to process some filters to remove most noise and uninteresting things. It usually involves spatial and temporal blur to fix noise variance and dynamic filters to fix noise mean.

Images provided by a camera have noise, which can corrupt computer vision algorithm. For instance, in order to detect features, we need to extract gradient, which can be distorted by noise variance. This kind of noise can be reduced by blur, especially Gaussian blur for Gaussian noise, which dilutes variance in neighboring pixels. However, this two-dimensional spatial blur should not be too strong, at the risk of producing diplopia. Moreover, the camera video has also a third dimension: time. A time blur along consecutive images could also reduce noise variance. However, it is rather effective on static video: mixing too many different images produces a ghost effect [15].

These previous filters are static, and they cannot change their parameters (which are calibrated for the hardware). We need dynamic filters to adapt automatically to the environment. Indeed, luminosity and contrast can change along time. A filter can keep the same luminosity, or change color distribution by analyzing and equalizing Hue histogram. As compensation, it increases noise variance [16]. The main issue is that we need to choose parameters depending on the environment. Completely independent methods exist, but they consume too much CPU resources, thus it could be useful to improve a single interesting image, just that is too much of a guzzler for autonomous navigation [17].

5.3. Extracting Interesting Data

The two main properties of objects in an image are color and shape. Color recognition is easy to implement, but not discriminating enough (different objects can have the same color). On the contrary, shape recognition is reliable, but not so easy to program.

We can combine advantages with these two methods: first, color recognition only keeps a little data corresponding to the right color, and then shape recognition processes these lightened data.

5.3.1. HSV Filtering

The first step is HSV filtering. It keeps pixels whose “hue” corresponds to the right color, “saturation” and “value” above a threshold to eliminate gray and dark areas. The result is a binary image, in which white pixels are kept, and blue pixels are rejected. The issue is that many good pixels are rejected, which creates little holes. The solution is to “dilate” the image to recover these pixels, and “erode” to remove the border effect of dilate (see Figure 6).

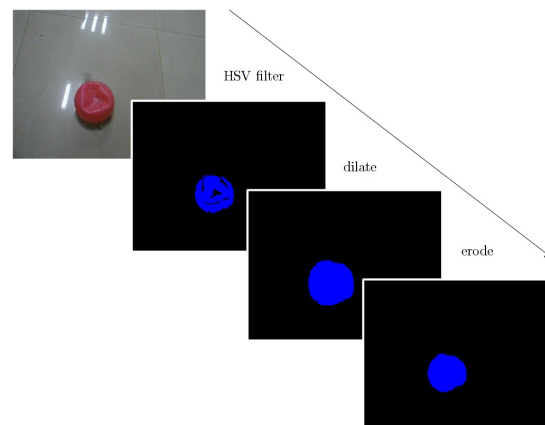


Figure 6. Extract interesting color with HSV filter.

5.3.2. Blob Detection

Blob detection consists of linking adjacent pixels to create groups—each one corresponding to an object. This algorithm [18] is implemented in the “findContours” function of OpenCV. Then, each blob has its centroid computed, whose coordinates are understandable data for the navigation algorithm (see Figure 7).

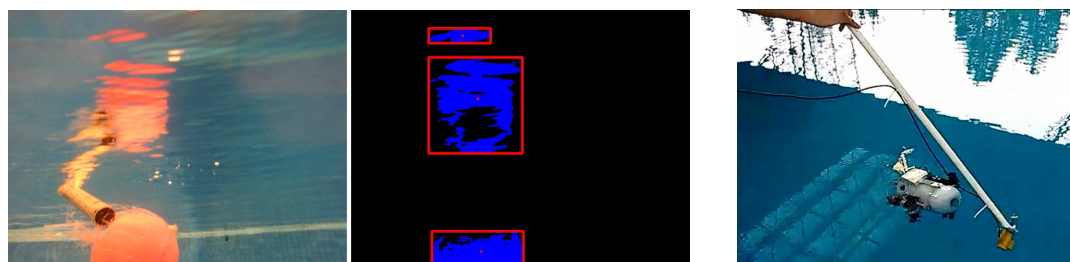


Figure 7. Segmentation to different blobs and the detection of another object.

5.4. Take Decision

Orders for motors are generated depending on the state of the FSM (Finite State Machine). Main states are:

- Remote control (no autonomous, the user sends orders).
- Stabilize (keep same position and posture).
- Go up and Go down (change only depth and stabilize).
- Explore (follow a planned path).
- Follow an object by the Raspberry Pi camera.

Ball Following through the Algorithm Vision

Blob detection gives the coordinates and size of each blob by giving HSV boundaries of interesting objects. It is required to know before the mission what the robot is looking for. The first step is to show to the robot an OPI (Object of Potential Interest) and manually select HSV boundaries until it is well detected. It must be done underwater because colors are not the same in the laboratory (Figures 6 and 7 show pictures of the same OPI, outside and inside water). On the other hand, objects that avoid its reflection on the surface due to the surrounding light should be considered to obtain a better detection of the object. In addition, another object was detected as shown in Figure 7. However, the detection of the object was fine while the sun is neglected on the surface, due to the yellow color of this object. Therefore, the detection of the objects is good until they are not close to the surface when their reflection appears or they are confused with others of the same color.

Then, the robot is able to find the coordinate (x_{opi}, y_{opi}) of the biggest object seen by its camera, which it will try to follow. Due to the shape of AR2D2, a simple proportional law control is enough. By comparing to the center (x_c, y_c) of the image, and its size to a threshold size $thresh$, we obtain this kind of control law:

$$\begin{bmatrix} mot_1 \\ mot_2 \\ mot_3 \\ mot_4 \end{bmatrix} = \begin{bmatrix} +K_{py} \\ +K_{py} \\ +K_{px} \\ -K_{px} \end{bmatrix} * \begin{bmatrix} y_{opi} - y_c \\ y_{opi} - y_c \\ x_{opi} - x_c \\ x_{opi} - x_c \end{bmatrix} + K_{size} * \begin{bmatrix} 0 \\ 0 \\ size_{tresh} - size \\ size_{tresh} - size \end{bmatrix}. \quad (49)$$

K_{px} and K_{py} try to minimize horizontal and vertical error of the OPI's centroid compared to the center of the image. Meanwhile, K_{size} keeps the robot to a good distance from the object (here, the size gives an approximation of distance). In Figure 8, we can observe an experimental test of the detection of the ball.

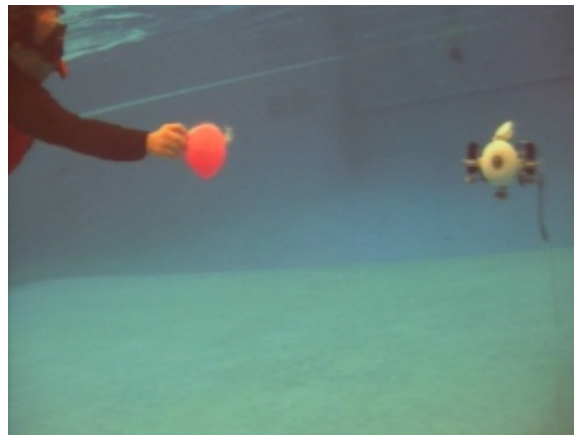


Figure 8. Ball following test into the swimming pool.

6. Simulation and Results

Matlab Simulink (R2016a, MathWorks) has been used to apply the control law chosen in order to stabilize the surge and heave movements on the dynamic model equations.

Simulation results are presented in order to observe the performance of the translation movements by the proposed control law. Nonlinear PD control was tuned to observe the best behavior versus disturbances, where the proportional and derivative gains on the saturation functions relative to surge and heave control, belong to values; $b = +/ - 4$ and $d = +/ - 2$, thus the time evolution of the “x” vehicle's position is shown in Figure 9, while its “z” position of the closed-loop system are plotted in Figure 10. Observe that the state positions (x, z) are externally perturbed and the nonlinear

PD control is again able to stabilize the engine position. The initial conditions used for this simulation are $x(0) = 0, z(0) = 0, \dot{x}(0) = 0, \dot{z}(0) = 0$ and the desired values are $x_d = 1.5$ and $z_d = 1$.

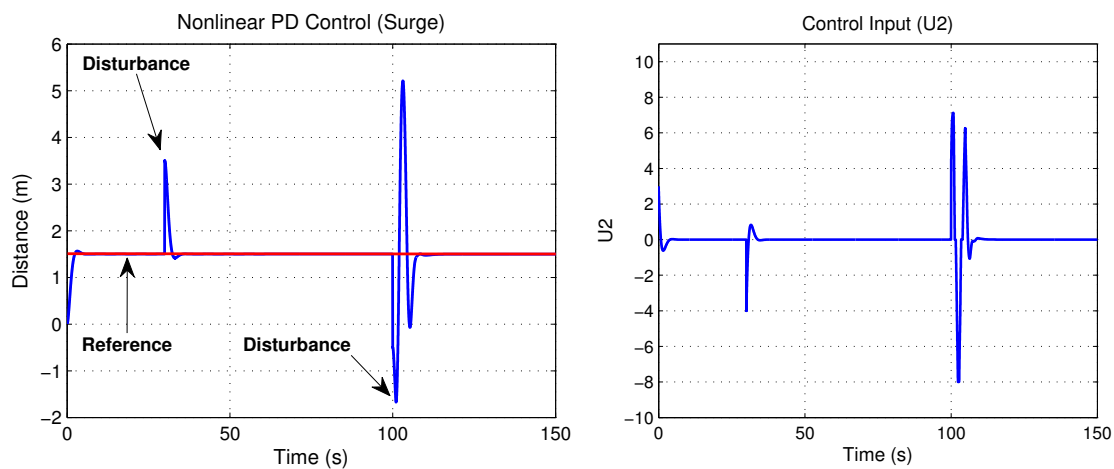


Figure 9. Surge movement and its control input U2.

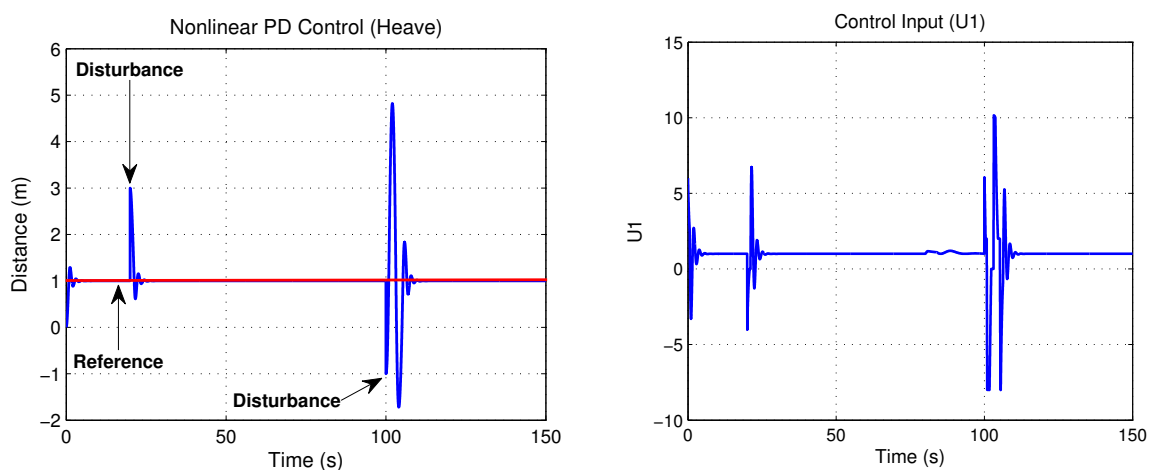


Figure 10. Heave movement and its control input U1.

Control input U1 is directly applied in vertical forces and control input U2 is applied in horizontal forces. Figures 9 and 10 show both control inputs. Observe that control input U1 converges to the weight value $f_W = 1$, whilst that control input U2 converges to zero, due to horizontal forces' combination.

In order to characterize the thrusters, we did some tests in the fish tank, considering the reaction of the thruster on the z-axis through a PWM signal (Pulse Width Modulation) for the maneuverability index (see Figure 11). Therefore, $F = (\text{PWM}/50) - 1$ is the linear equation obtained by the adjustment of the curve. On the other hand, in recent works, such as in [19], it is important to consider a greater knowledge of the dynamic of all the thrusters to obtain the best performance and efficiency with respect to the maneuvers of the vehicle in control applications. In this case, the Micro AUV has the layout configuration of the thrusters near the center of gravity to obtain better maneuverability.

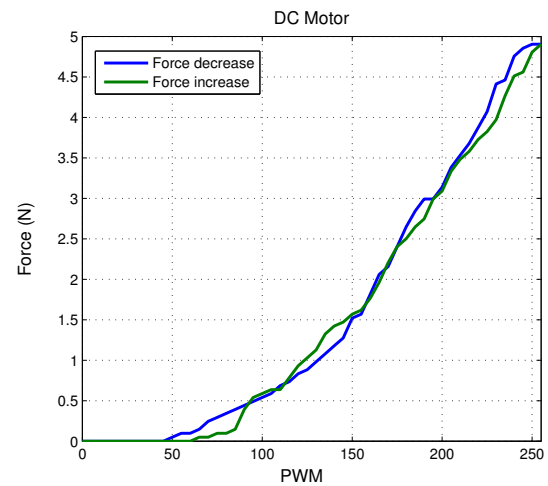
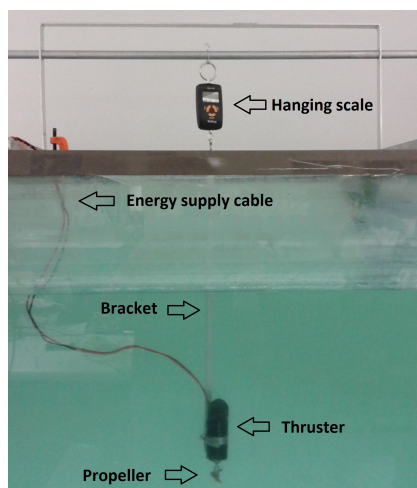


Figure 11. Thruster test.

Figures 12 and 13 show the experimental results relative to surge and heave movements and its respective error signal in the regulation, while the torques of the thrusters on both axes are depicted in Figure 14. Notice that the control law proposed presents a good performance. The gains tuning was fixed in order to get the best behaviour of the micro submarine exposed to external disturbs. Concerning the stability improvement, it is possible to fix the gains tuning with more accuracy, considering the controller characteristic, being able to saturate the control input and protect the actuators.

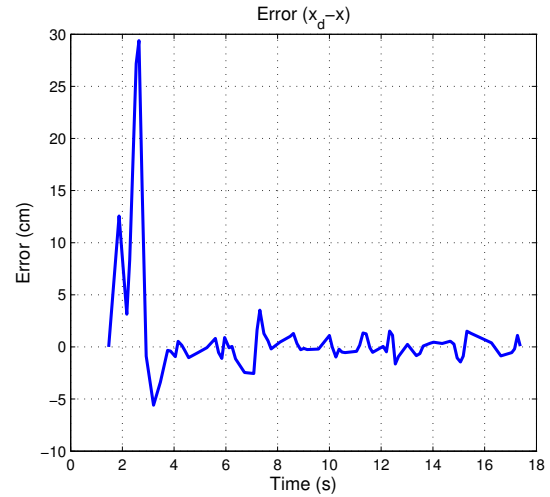
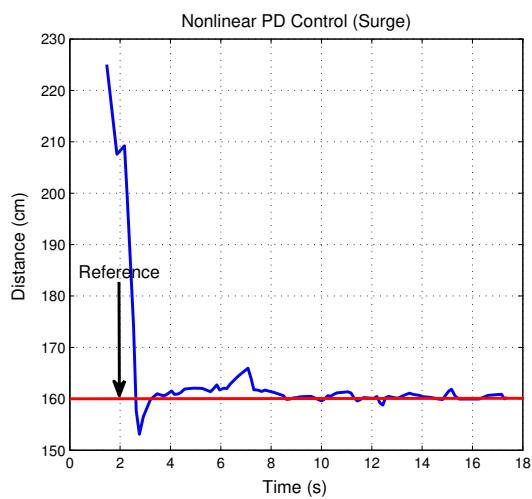


Figure 12. Experimental surge movement and its signal of error in regulation.

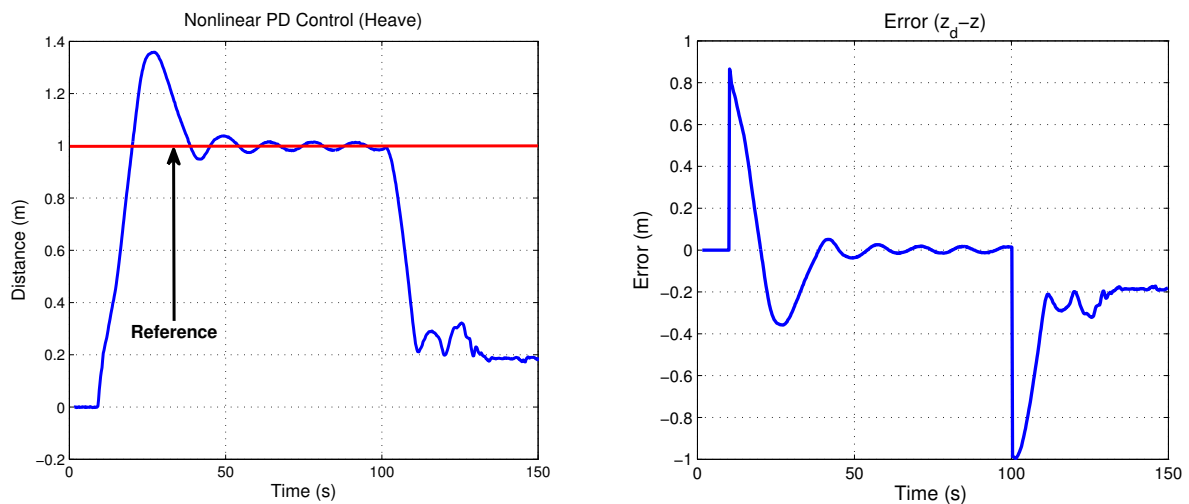


Figure 13. Experimental heave movement and its signal of error in regulation.

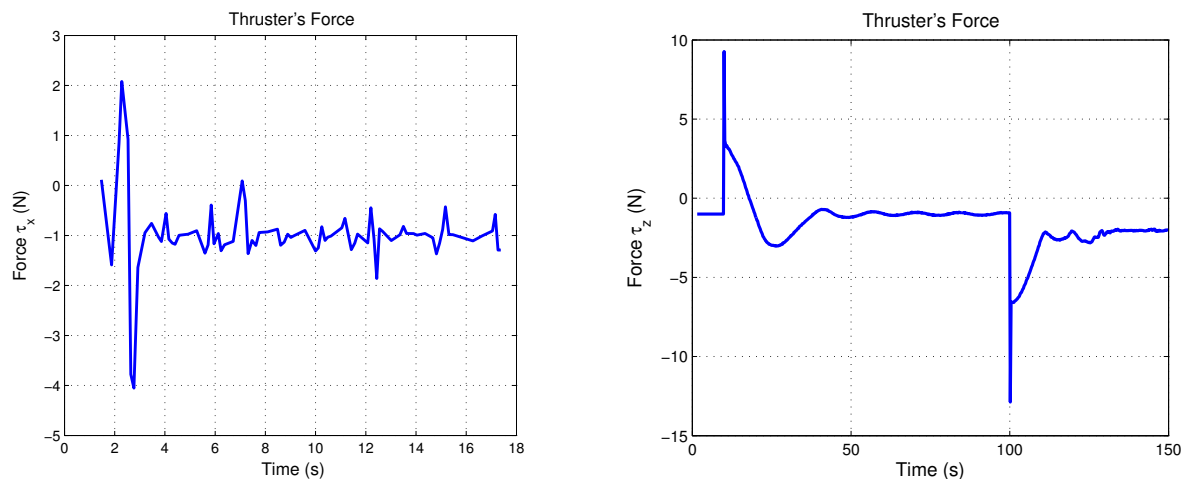


Figure 14. Experimental reaction of the torques on the x - and z -axes.

The ball followed a random path (it is not a stabilization test) with random velocity. The robot managed to follow it, even when the object goes out of image. Indeed, inertia of the robot into water preserves its movement, making the robot able to find the OPI again.

7. Conclusions

The development of this prototype was motivated by the need for having an autonomous small vehicle for operation in closed environments. We developed and presented the embedded control system of the AR2D2 Micro AUV robot. In this initial work, we have considered the problem of set-point regulation on heave and surge movement. The vehicle was designed with the aim of having a small platform able to make complex maneuvers; to this end, the dynamic decoupling of surge and sway turned to be a key issue. In turn, the navigation embedded system was reduced as much as possible. On one hand, one is restricted to work in closed volumes, and, on the other hand, we are looking for cooperative tasks as the major goal of our future research work [20]. In this paper, we have used a typical nonlinear controller with saturation functions, thus the closed loop stability was demonstrated on the basis of Lyapunov's theory. The desired behavior was validated by Matlab's simulation and experiments. On the other side, this vehicle performs real-time embedded image processing in order to recognize and follow an object of interest. The results show a good ability to do

it, without losing the OPI. In the future, we will implement another control strategy aiming to improve the performances along yaw angle and regulation in x , in order to be close to the object for further inspection. Introducing some filters along time can also improve behavior of the robot, and stabilize recognition of object (by processing sequential video images instead of single pictures). Moreover, we will conduct experiments in a natural environment with less visibility.

Author Contributions: J.A.M.-A. and C.R. designed the prototype, implemented the algorithms of artificial vision and control strategy, obtained the experimental results and wrote this paper; E.C.-M. validated the control scheme and provided guidance for experiments, T.S.-J. guided in the dynamic model process, and provided supervision and support of the work, L.G.G.-V. performed the simulations and revised the article.

Funding: This study is part of the project number 201441 “Implementation of oceanographic observation networks (physical, geochemical, ecological) for generating scenarios of possible contingencies related to the exploration and production of hydrocarbons in the deepwater Gulf of Mexico”, granted by SENER-CONACYT Hydrocarbons Sectorial Fund.

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

References

1. Antonelli, G. *Underwater Robots*; STAR (Springer Tracts in Advance Robotics), 3rd ed.; Springer: Berlin, Germany, 2014; ISBN 978-3-319-02876-7.
2. Carreras, M.; Ridao, P.; García, R.; Ribas, D.; Palomeras, N. Inspección visual subacuática mediante robótica submarina. *Rev. Iber. Aut. Inf. Ind.* **2012**, *9*, 34–45. [[CrossRef](#)]
3. Clark, C.; Olstad, C.; Buhagiar, K.; Gambin, T. Archaeology via Underwater Robots: Mapping and Localization within Maltese Cistern Systems. In Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision, Hanoi, Vietnam, 17–20 December 2008; pp. 662–667. [[CrossRef](#)]
4. Monroy, J.; Campos, E.; Torres, J. Attitude Control of a Micro AUV Through an Embedded System. *IEEE Lat. Am. Trans.* **2017**, *15*, 603–612. [[CrossRef](#)]
5. Watson, S.; Green, P. Depth Control for Micro-Autonomous Underwater Vehicles (μ AUVs): Simulation and Experimentation. *Int. J. Adv. Robot. Syst.* **2014**, *11*, 1–10. [[CrossRef](#)]
6. Fechner, S.; Kerdels, J.; Albiez, J. Design of a μ AUV. In Proceedings of the 4th International Symposium on Autonomous Minirobots for Research and Edutainment, Buenos Aires, Argentina, 2–5 October 2007; pp. 99–106; ISBN 978-3-939350-35-4.
7. Rodríguez, P.; Piera, J. Mini AUV, a platform for future use on marine research for the Spanish Research Council? *Instrum. ViewPo.* **2005**, *4*, 14–15; ISSN 1697-2562.
8. Fossen, T. *Guidance and Control of Ocean Vehicles*; John Wiley and Sons Ltd., University of Trondheim: Trondheim, Norway, 1999; ISBN 0471941131.
9. Goldstein, T.; Poole, C.; Safko, J. *Classical Mechanics*; 2nd ed.; Addison-Wesley: Boston, MA, USA, 1983; ISBN 8185015538.
10. Marsden, J. *Elementary Classical Analysis*; W.H. Freeman and Company: San Francisco, CA, USA, 1974.
11. Fossen, T. *Marine Control Systems: Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles*; Marine Cybernetics: Trondheim, Norway, 2002; ISBN 8292356002.
12. SNAME, Technical and Research Committee. *Nomenclature for Treating the Motion of a Submerged Body Through a Fluid*; The Society of Naval Architects and Marine Engineers: New York, NY, USA, 1950; pp. 1–5.
13. Campos, E.; Monroy, J.; Abundis, H.; Chemori, A.; Creuze, V.; Torres, J. A nonlinear controller based on saturation functions with variable parameters to stabilize an AUV. *Int. J. Nav. Arch. Ocean Eng.* **2018**, *10*, in press. [[CrossRef](#)]
14. Kelly, R.; Carelli, R. A Class of Nonlinear PD-type Controller for Robot Manipulator. *J. Field Robot.* **1996**, *13*, 793–802. [[CrossRef](#)]
15. Basile, B.; Blake, A.; Zisserman, A. Motion Deblurring and Super-Resolution from an Image Sequence. In Proceedings of the 4th European Conference on Computer Vision, Cambridge, UK, 15–18 April 1996; pp. 571–582; ISBN 9783540611233.
16. Pizer, S.; Amburn, E.; Austin, J.; Cromartie, R.; Geselowitz, A.; Greer, T.; Romeny, B.; Zimmerman, J. Adaptive Histogram Equalization and Its Variations. *Comp. Vis. Graph. Im. Proc. Elsevier* **1987**, *39*, 355–368. [[CrossRef](#)]

17. Bazeille, S.; Quidu, I.; Jaulin, L.; Malkasse, J. Automatic Underwater Image Pre Processing. In Proceedings of the SEA TECH WEEK Characterisation Du Milieu Marin, Brest, France, 16–19 October 2006.
18. Suzuki, S.; Abe, K. Topological Structural Analysis of Digitized Binary Images by Border Following. *Comp. Vis. Grap. Im. Proc. Elsevier* **1985**, *30*, 32–46. [[CrossRef](#)]
19. Pugi, L.; Allotta, B.; Pagliai, M. Redundant and reconfigurable propulsion systems to improve motion capability of underwater vehicles. *Ocean Eng.* **2018**, *148*, 376–385. [[CrossRef](#)]
20. Yoon, S.; Qiao, C. Cooperative Search and Survey Using Autonomous Underwater Vehicles (AUVs). *IEEE Trans. Par. Dist. Syst.* **2011**, *22*, 364–379. [[CrossRef](#)]



© 2018 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 (<http://creativecommons.org/licenses/by/4.0/>).