Advanced control and navigation of ROV/AUVs

Eduardo Coelho Martins de Almeida Cunha
eduardo.m.a.cunha@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2022

Abstract

In the last decades, Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs) have become tools par excellence for the execution of scientific, commercial and military missions at sea, due to their versatility and relatively low cost. This work addresses the problem of endowing such vehicles with the capability to follow specified paths automatically, while being robust to model uncertainty and external disturbances. This thesis starts with a brief overview of motion control problems, followed by the solutions adopted for the guidance outer-loop, inner-loop control using Sliding Mode Control (SMC) and thrust allocation methods. Two marine robotic vehicles take central stage in this work, MEDUSA and BlueROV, for which kinematic and dynamic models are introduced. This is followed by the design of two types of inner-loop controllers resorting to SMC strategies. These controllers are first implemented in *MATLAB* software for rapid iteration and finally coded in *C++*/ROS framework, enabling their application in real vehicles. Simulation results are performed with the objective of assessing the performance of the controllers designed and how it compares with the performance of existing Proportional-Integral-Derivative (PID) control laws. The results obtained show that SMC laws exhibit a good degree of robustness against model uncertainty, together with external disturbance attenuation. Furthermore, two thrust allocation methods are also compared and analysed. The SMC controllers are tested in a real environment with BlueROV, attesting to their adequacy as a robust control approach for ROVs/AUVs. Finally, future work is described, focusing on promising developments in this field of work. **Keywords:** Nonlinear Control, Autonomous Underwater Vehicle (AUV), Remotely Operated Vehicle (ROV), Sliding Mode Control (SMC)

1. Introduction

The ocean is an immensely vast body of water, covering up to 70.8% of the Earth's surface [15]. The ocean has been a crucial resource for Mankind ever since the dawn of Humanity and there is evidence that it played a significant role in the emergence of life on Earth, 3.77 billion years ago [3]. Even though ocean exploration dates as far back as 4500 BC [13], it is a fact that more than 80% of it is still unmapped, unobserved and unexplored [1]. In order to overcome these hurdles, tremendous progress has been made over the past decades towards the development and operation of Remotely Operated Vehicles (ROVs) for seafloor observation and inspection and intervention on underwater infrastructures and Autonomous Underwater Vehicles (AUVs) for marine habitat mapping and oceanographic studies. Over the years, ROVs/AUVs have grown in popularity due to their operational versatility, ranging from military applications to scientific uses [7, 4, 8]. There is a wide variety of ROVs, ranging from small vessels (Class I/II), with the sole purpose of observation and gathering data using sensors, to large vehicles (Class III/IV), with huge capability to interact with the environment. Class I/II vehicles, being much cheaper and easier to transport, are simpler to deploy and use in most real-world situations, without the need for large ships or a numerous crew, which is needed for most Class III/IV vehicles.

Encouraged by the importance of this topic, the main goal of the present manuscript is to design a method capable of autonomously operating an ROV/AUV using state of the art techniques that are rooted in a solid mathematical foundation.

1.1. Problem Description and Objectives

In the last decades, considerable progress has been done in the field of marine robotics, in an effort to make these vehicles more accessible, cheaper and useful in underwater inspections tasks and empower scientists with the tools to unrestrainedly explore and discover the Earth's ocean. Nonetheless, considerable work remains to be done in the field of marine vehicle motion control, to endow underwater robots with the capability to perform in a

reliable and efficient manner in extremely harsh environments.

1.1.1 AUV uses in extreme environments

AUVs have been used in several exploration missions in severe environments such as the Arctic. For instance, the Arctic Gakkel Vents Expedition (AGAVE) had the objective of exploring and understanding the marine biology, chemistry and geology of a hydrothermal section of the Gakkel Ridge [10]. Furthermore, the deepest place in Earth's surface, which is situated almost 11 kilometres below sea level, has been explored by ROV/AUVs since 1995, such as KAIKO, a japanese ROV - [17]. This vehicle can be observed in Figure 1.

Figure 1: KAIKO (かいこう) ROV before a mission in the Pacific Ocean.

As stated before, it is clear that this type of missions will greatly benefit from accurate and robust motion control techniques, since effective tracking and good external disturbance rejection facilitates ROV operators.

1.1.2 Motion Control in the presence of Model Uncertainty and External Disturbances

There are two key motion control building blocks at the core of any advanced underwater vehicle, commonly known as Trajectory Tracking and Path Following systems. In order to make sure that AUV missions are executed successfully, their motion control systems should be designed in such a way as to reduce the closed-loop impact of external disturbances (currents and waves) and sensor noise. They must also exhibit robustness against model uncertainty associated with a number of parameters, of which added-mass and damping terms are representative examples.

1.1.3 Motion Control Solutions

One of the most widely used type of controllers is Proportional-Integral-Derivative (PID) control, which has become quite popular in the case of linear plant models. However, it has been shown by Pretagostini et al. in [16] that this type of controllers is highly dependent on gain tuning and may exhibit poor performance in the presence of external disturbances and process noise.

In the past decades, a new breed of controllers that fall in the scope of Sliding Mode Control (SMC) theory have emerged [12] that show good potential to explicitly reject external disturbances and model uncertainty, even in the case of nonlinear plant models.

The aim of this work is to analyse and develop a state of the art control method called SMC, so as to operate at the system's dynamic level and also design, test and simulate motion control approach for a ROV/AUV both virtually and in the real world, using SMC theory for the inner-loop controllers and Path Following methodologies for the outer-loop controllers.

2. State of the Art

2.1. Path Following

In general, autonomously manoeuvring a marine vehicle like an Autonomous Underwater Vehicle (AUV) is achieved by ensuring the craft can follow a specific path and encompasses two main motion control problems: Trajectory Tracking and Path Following, the latter being the chosen strategy.

Path Following is defined as the design of control laws such that a vehicle converges and follows a determined path without an associated temporal law [2]. This path can be represented as $p_{\text{ref}}(\gamma)$, where γ is the vehicle's progression along it and $\dot{\gamma}$ is the referred speed law. A geometric illustration of path following can be observed in 2.

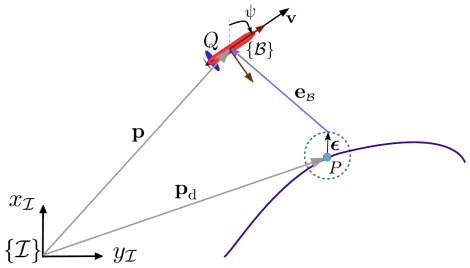


Figure 2: Geometric illustration of path following (adapted from [9]).

2.2. Control Methods

Control Theory studies the task of controlling dynamic systems, i.e., systems which change through time. This consists in designing adequate input signals such that the system's state is driven towards and stabilises at a desired state.

2.2.1 Sliding Mode Control

Variable Structure Control System (VSCS) is family of dynamical control systems whose structure changes according with the present value of the state of the system it is controlling [18], making it a discontinuous nonlinear function of time. Sliding Mode Control (SMC) is a type of VSCS - it utilises a

high-speed switching control law to ensure two main objectives. First, in the so-called *reaching phase*, the plant's state trajectory is driven onto a user-chosen hypersurface in the state space, called the sliding surface. Second, it keeps the plant's state in this hypersurface for all succeeding time (sliding phase). During this phase, the plant's state constantly switches from one side of the surface to the other, *sliding* along its vicinity until it reaches the desired equilibrium [12]. This behaviour is exemplified in figure 3.

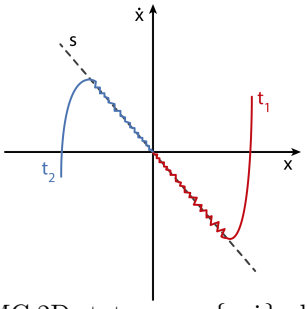


Figure 3: SMC 2D state-space $\{x\ \dot{x}\}$ phase portrait for two initial states; s is the sliding surface and t_1 and t_2 are state space trajectories.

Sliding Surface Design

As suggested by Liu et al. in [12], the sliding surface $s(\mathbf{x})$ can be described as

$$s(\mathbf{x}) = \mathbf{c}^T\mathbf{x} = \sum_{i=1}^{n} c_i x_i = \sum_{i=1}^{n-1} c_i x_i + x_n, \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector and $\mathbf{c} = [c_1\ \dots\ c_{n-1}\ 1]^T$ is a vector of coefficients. In this representation, the parameters in $\mathbf{c}$ should be selected such that the polynomial $p^{n-1} + c_n p^{n-2} + \dots + c_2 p + c_1$ is a Hurwitz polynomial, where p is a Laplace operator, i.e., the eigenvalues of the polynomial must have a negative real part. This ensures that, when the plant's state $\mathbf{x}$ reaches the sliding surface ($s(\mathbf{x}) = 0$), the equilibrium point is reached at least exponentially, i.e., the origin is exponentially stable.

In an improvement on (1), as suggested in [14], the sliding surface s can be expressed as

$$s(\mathbf{x}) = s_0(\mathbf{e}) + z(\mathbf{e}, t), \quad (2)$$

where $s_0(\mathbf{e}(t))$ is the sliding surface just described and $z(\mathbf{e}(t), t)$ an added term which ensures an exponential reaching law, $z(\mathbf{e}, t) = s_0(\mathbf{e}(0))e^{-k_c t}$, $k_c > 0$. Reaching laws will be described in more detail later. The conventional sliding surface $s_0(\mathbf{e})$ is computed using (1), where $\mathbf{x}$ is substituted by $\mathbf{e}$, the vector of the tracking error,

$$e = x - x_d, \quad (3)$$

and its derivatives (considering x and x_d to be the state and the desired state, respectively), up to the $(n-1)$th degree, such that

$$\mathbf{e} = [e\ e^{(1)}\ \dots\ e^{(n-1)}]^T, \quad e^{(i)} = \frac{d^i}{dt^i}e, \quad (4)$$

and n is the degree of the sliding surface.

Stability

In order to ensure stability [18], all trajectories in the state space must be attracted by the sliding surface s, which can be summed up as (1) all trajectories starting on the surface s must remain there (s is an invariant set) and (2) all trajectories starting outside of the surface s tend to it at least asymptotically, which can be translated as the following condition,

$$s\dot{s} < 0. \quad (5)$$

Reaching Laws

SMC, compared with other families of controllers, has the advantage of enforcing desired dynamics on the system. This is achieved by choosing the input control in a way that imposes the desired dynamic on the time derivative of the sliding surface, $\dot{s}$, while ensuring stability. A generic reaching law [12] is:

$$\dot{s} = -\epsilon\,\mathrm{sgn}\,(s) - f(s), \quad \epsilon > 0. \quad (6)$$

Moreover, the value of ϵ should be chosen, such that it is larger than the upper bound of the absolute value of the disturbances affecting the system, which makes sure that it can oppose these current perturbations [14]. Furthermore, the main challenge of SMC concerns the discontinuous property of the reaching law, which can lead to chattering, high amplitude and high frequency control inputs.

Control Law Definition

Considering a generic non-linear system with state $\mathbf{x}$, the dynamics of one of its state variables $x_i(t)$ can be described as $\ddot{x}_i(t) = g(\mathbf{x}(t), \tau_i(t))$ and its tracking error as $e_i(t) = x_i(t) - x_{i,d}(t)$. Considering a second order sliding surface (1) $s(t) = \dot{e}_i(t) + c_i e_i(t)$ and assigning a generic reaching law (6), the control law $\tau_i(t)$ can be yielded from $g(\mathbf{x}(t), \tau_i(t)) - \ddot{x}_{i,d}(t) + c_i\dot{e}_i(t) = -\epsilon\,\mathrm{sgn}\,(s(t)) - f(s(t))$.

2.2.2 SMC Smoothing Solution

SMC's high-speed discontinuous switching law has undesirable effects. As stated in [14], one solution to the damaging chattering in the control inputs is using a low-pass first-order linear filter. As such, let the control input τ be defined as

$$\tau = \tau_0 + \tau_1, \quad (7)$$

such that τ_0 is the ideal control input (continuous term) and τ_1 is designed to reject perturbations (discontinuous term), i.e., the term which contains the sgn (.) function. Hence, the low-pass filter is

applied to the latter term, such that its output τ_{1av} is defined as

$$\mu\dot{\tau}_{1av}(t) = -\tau_{1av}(t) + \tau_1(t), \tag{8}$$

where $\mu \in [0, 1]$ is a constant such that $1/\mu$ is the filter's cutting frequency.

2.3. Actuation on Underwater Vehicles

Actuation on underwater vehicles is related to their own actuators, which are devices capable of generating forces. These forces produce a total force and torque on the vehicle, based on their position.

2.3.1 Allocation Methods for Underwater Vehicles

Following the notation used by Fossen et. al in [6], let $\boldsymbol{\tau}_i \in \mathbb{R}^n$ be the vector of forces and moments applied to the AUV produced by thruster i and $\mathbf{f}_i = [F_x \ F_y \ F_z]^T$ be the vector of forces produced by that same thruster along the three translational axis. It is possible to relate these physical quantities through (9),

$$\boldsymbol{\tau}_i = \begin{bmatrix} \mathbf{f}_i \\ \ell_i \times \mathbf{f}_i \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ F_z\ell_y - F_y\ell_z \\ F_x\ell_z - F_z\ell_x \\ F_y\ell_x - F_x\ell_y \end{bmatrix}_i, \tag{9}$$

where $\ell_i = [\ell_x \ \ell_y \ \ell_z]_i^T$ is the vector of moment arms, i.e., the orthogonal distance between the thruster and each axis of rotation.

When dealing with more than one thruster, a more compact notation is used. The vector of forces and moments applied to the AUV $\boldsymbol{\tau} \in \mathbb{R}^n$ are written as

$$\boldsymbol{\tau} = T\mathbf{f}, \tag{10}$$

where $T \in \mathbb{R}^{n\times r}$ is the thrust configuration matrix and $\mathbf{f} \in \mathbb{R}^r$ is the vector of forces produced by all r thrusters present in the AUV. Now, the design challenge lies in computing $\mathbf{f}$ given a desired $\boldsymbol{\tau}$. Underactuated vehicles' thrusters can not produce an arbitrary force and torque on the vehicle's body, whereas for overactuated it is possible, leading to an infinite number of solutions for $\mathbf{f}$, from which solutions that meet certain requirements can be searched for. Thus, several methods have been proposed to approach this issue, where an optimal vector $\mathbf{f}$ is computed:

- **Linear Quadratic Unconstrained Control Allocation**

  This approach solves the unconstrained Least-Squares (LS) Optimisation Problem in (11), considering $\mathbf{f}$ and $\mathbf{u}$ as unbounded,

$$\begin{aligned} \min_{\mathbf{f}} \quad & \{J = \mathbf{f}^T W \mathbf{f}\} \\ \text{subject to} \quad & \boldsymbol{\tau} - T\mathbf{f} = \mathbf{0}, \end{aligned} \tag{11}$$

  where $W \in \mathbb{R}^{r\times r}$ is a positive definite matrix. In a situation where all actuators equally contribute to the overall energy expense, matrix W is the identity matrix I. If TT^T is invertible, the solution for this optimisation problem is

$$\mathbf{f} = T^+\boldsymbol{\tau}, \tag{12}$$

  where T^+ is the Moore-Penrose pseudo inverse,

$$T^+ = T^T(TT^T)^{-1}. \tag{13}$$

- **Linear Quadratic Constrained Control Allocation**

  Going beyond the unconstrained approach, (14) is formulated in order to also accommodate actuator saturation, wear and tear and power system overload. Here, we keep $\mathbf{f} = [f_1 \ f_2 \ \cdots \ f_r]$ as the vector of forces of the actuators and W solves a positive definite matrix weighting these forces. In addition, $\mathbf{s} \in \mathbb{R}^n$ is a vector of slack variables, $\bar{f} = \max_i |f_i|$ and $\mathbf{f}_{min}$ and $\mathbf{f}_{max}$ are, respectively, the vectors with the minimum and maximum values reachable by each one of the actuators.

$$\begin{aligned} \min_{\mathbf{f},\mathbf{s},\bar{f}} \quad & \{J = \mathbf{f}^T W \mathbf{f} + \mathbf{s}^T Q \mathbf{s} + \beta\bar{f}\} \\ \text{subject to} \quad & T\mathbf{f} = \boldsymbol{\tau} + \mathbf{s} \\ & \mathbf{f}_{min} \le \mathbf{f} \le \mathbf{f}_{max} \\ & -\bar{f} \le f_1, \ldots, f_r \le \bar{f} \end{aligned} \tag{14}$$

  This problem can be reconstructed as a Quadratic Programming (QP) problem, which is solvable in real time using Nonlinear Programming (NLP) solvers, such as the primal-dual interior point method.

3. Vehicle Model

3.1. Reference Frames and General Notation

The adopted notation for coordinates and reference frames follows [5], where a general marine craft experiences motion in all 6 DOF, i.e., motion in all three axes and rotation around these same axes, as seen in figure 4.

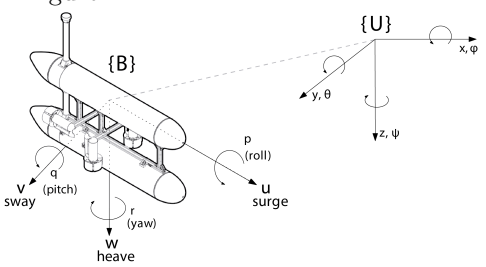


Figure 4: Reference Frames used, according to [5].

Hence, the adopted notation considers $\{U\}$ the body frame, usually attached to the geometric centre of mass of the vehicle; $\{B\}$ is the inertial reference frame; $\boldsymbol{\eta}_1 = [x\ y\ z]^T$ is the position of the origin of $\{B\}$ measured in $\{U\}$; $\boldsymbol{\eta}_2 = [\phi\ \theta\ \psi]^T$ is the orientation of $\{B\}$ with respect to $\{U\}$; $\boldsymbol{\nu}_1 = [u\ v\ w]^T$ is the linear velocity of the origin of $\{B\}$ with respect to $\{U\}$, expressed in $\{B\}$; $\boldsymbol{\nu}_2 = [p\ q\ r]^T$ is the angular velocity of the origin of $\{B\}$ with respect to $\{U\}$, expressed in $\{B\}$; $F_{RB} = [X\ Y\ Z]^T$ is the external forces measured in $\{B\}$ and $N_{RB} = [K\ M\ N]^T$ is the external torques measured in $\{B\}$. As such, $\boldsymbol{\eta} = [\boldsymbol{\eta}_1^T\ \boldsymbol{\eta}_2^T]^T$ and $\boldsymbol{\nu} = [\boldsymbol{\nu}_1^T\ \boldsymbol{\nu}_2^T]^T$.

3.2. AUV Simplified Motion

Taking into consideration the work to be developed, both the kinematics and dynamics of the AUV can be simplified. In this work's context, the vehicle is only manoeuvred at constant depth, such that movement in heave (along the z axis) is neglected. Also, rotations around the x and y axes, ϕ and θ respectively, are not considered. Thus,

$$\phi = 0, \quad \theta = 0. \tag{15}$$

As such, the vehicle is reduced to a 3 DOF craft, given by $\boldsymbol{\eta} = [x\ y\ \psi]^T$. Also, the linear and angular velocity of the body frame $\{B\}$ is also redefined, as $\boldsymbol{\nu} = [u\ v\ r]^T$. Thus, the kinematics equations are simplified as

$$\begin{cases} \dot{x} = u\cos\psi - v\sin\psi \\ \dot{y} = u\sin\psi + v\cos\psi \\ \dot{\psi} = r \end{cases}. \tag{16}$$

Furthermore, considering that the control input only actuates through forces in the x and y axes and through the torque around the z axis, then $\boldsymbol{\tau} = [\tau_u\ \tau_v\ \tau_r]^T$. Hence, the dynamics equations are also simplified to

$$\begin{cases} m_u\dot{u} - m_v v r + d_u u = \tau_u \\ m_v\dot{v} + m_u u r + d_v v = \tau_v \\ m_r\dot{r} - m_{uv} u v + d_r r = \tau_r \end{cases}. \tag{17}$$

These coefficients can be computed by the following expressions

$$\begin{aligned} m_u = m - X_{\dot{u}}, \quad m_v = m - Y_{\dot{v}}, \\ m_r = I_z - N_{\dot{r}}, \quad m_{uv} = m_u - m_v, \\ d_u = -X_u - X_{|u|u}|u|, \quad d_v = -Y_v - Y_{|v|v}|v|, \\ d_r = -N_r - N_{|r|r}|r|, \end{aligned} \tag{18}$$

where $X_{\dot{u}}$, $Y_{\dot{v}}$ and $N_{\dot{r}}$ are hydrodynamic added masses due to the acceleration of the vehicle, modelling the motion of the fluid through which the craft navigates. Also, X_u, Y_v and N_r are terms related to linear damping, $X_{|u|u}$ is the axial drag and $Y_{|v|v}$ and $N_{|r|r}$ are parameters associated with crossflow drag [11].

3.3. BlueROV vehicle

The BlueROV (Figure 5), is the chosen vehicle for this work, used both in simulation and in the real-world.

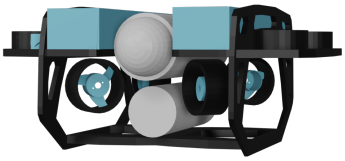


Figure 5: BlueROV vehicle: heavy configuration (3D Model).

This vehicle has four vertically oriented thrusters and other four thrusters in the horizontal plane, distributed in a 45 degrees configuration. This vehicle's parameters can be consulted in Table 1.

Table 1: BlueROV's parameters.

$X_{\dot{u}}$	X_u	$X_{\|u\|u}$
- 27.08 kg	-1.17 kg/s	- 46.27 kg/m
$Y_{\dot{v}}$	Y_v	$Y_{\|v\|v}$
- 25.95 kg	-1.17 kg/s	- 46.27 kg/m
$N_{\dot{r}}$	N_r	$N_{\|r\|r}$
1.00 kg · m^2	-0.5 kg · m/s	-1.0 kg · m

The inertia about the z axis and the vehicle's mass are also defined, such that, respectively, $I_z = 0.245$ kg · m^2 and $m = 11.5$ kg.

4. SMC Design and Application

Following an inner-outer loop system approach, SMC controllers are devised for the inner-loop structure, such that the forces and torques $\boldsymbol{\tau} = [\tau_u\ \tau_v\ \tau_r]^T$ are computed, given the guidance outer-loop references u_d, v_d and ψ_d (desired surge and sway velocities and yaw angle, respectively), with feedback of the tracking error and its derivatives $\mathbf{e}$ for each DOF.

4.1. SMC Controllers

Initially, the controllers are developed in order to prove some of the simpler concepts of SMC. However, in an a second approach, an improved solution for the control of surge and sway velocities is devised, since previously these state variables were controlled using first order sliding surfaces. The controlled variables of interest are now α and β, instead of u and v,

$$\alpha = \int_0^t u(\rho)d\rho, \quad \beta = \int_0^t v(\rho)d\rho. \tag{19}$$

Controlling the integral of the actual variable of interest has several advantages, namely a positive impact on the robustness of the controller against constant disturbances.

Moreover, instead of a constant reaching law, an exponential reaching law is used, in order to reduce the time taken to reach the control's sliding phase.

Also, the previously described smoothing solution in Section 2.2.2 is implemented, preventing chattering and high frequency and amplitude actuation signals, eliminating the need to use a saturation function instead of the sign function in the reaching law design.

4.1.1 Surge Velocity u

This controller uses α as variable of interest, as defined in (19). The tracking error vector in (4) is expressed as

$$\begin{aligned} \mathbf{e}_\alpha = [\tilde{\alpha}\ \dot{\tilde{\alpha}}\ \ddot{\tilde{\alpha}}]^T, \quad \tilde{\alpha} = \alpha - \alpha_d, \quad \dot{\tilde{\alpha}} = u - u_d, \\ \ddot{\tilde{\alpha}} = \dot{u} - \dot{u_d} = \frac{1}{m_u}\left(m_v v r - d_u u + \tau_u\right) - \dot{u}_d, \end{aligned} \tag{20}$$

where $\alpha_d = \int_0^t u_d(\rho)d\rho$. From (2), it follows that

$$s_\alpha = s_{0\alpha} + z_\alpha, \tag{21}$$

where $s_{0\alpha}$ is computed from (1) such that

$$s_{0\alpha} = \dot{\tilde{\alpha}} + \lambda_\alpha \tilde{\alpha}, \tag{22}$$

where z_α is defined as

$$z_\alpha(t) = -s_{0\alpha}(\mathbf{e}_\alpha(0))e^{-k_c t}, \tag{23}$$

so as to ensure an exponential reaching law. Furthermore, computing the derivative of s_α, $s_{0\alpha}$ and z_α yields

$$\begin{aligned} \dot{s}_\alpha = \frac{1}{m_u}\left(m_v v r - d_u u + \tau_u\right) - \dot{u}_d \\ + \lambda_\alpha(u - u_d) - k_c z_\alpha. \end{aligned} \tag{24}$$

Finally, according to (7) and (8),

$$\tau_u = \tau_{u0} + \tau_{u1_{av}}, \tag{25}$$

where τ_{u0} is the ideal control input, such that

$$\begin{aligned} \tau_{u0} = -m_v v r + d_u u - m_u\left[-\dot{u}_d\right. \\ \left. + \lambda_\alpha(u - u_d) + k_c s_{0\alpha}, \right. \end{aligned} \tag{26}$$

which implies that, according to (24),

$$\dot{s}_\alpha = -k_c s_\alpha + \frac{1}{m_u}\tau_{u1_{av}}, \tag{27}$$

where $\tau_{u1_{av}}$, according to (8), is the result of low-pass filtering τ_{u1} with a cutting frequency $1/\mu_\alpha$ Hz and is ruled by

$$\mu_\alpha \dot{\tau}_{u1_{av}}(t) = -\tau_{u1_{av}}(t) + \tau_{u1}(t), \tag{28}$$

where τ_{u1} follows from

$$\tau_{u1} = -m_u \epsilon_u \operatorname{sgn}(s_\alpha). \tag{29}$$

Hence, applying (26) to (25), yields

$$\begin{aligned} \tau_u = -m_v v r + d_u u - m_u\left[-\dot{u}_d\right. \\ \left. + \lambda_\alpha(u - u_d) + k_c s_{0\alpha} + \tau_{u1_{av}}. \right. \end{aligned} \tag{30}$$

4.1.2 Sway Velocity v

As stated previously, this controller uses β as variable of interest, as defined in (19). Following the same mathematical process as for the computation of τ_u, it is possible to derive the following expression for τ_v,

$$\begin{aligned} \tau_v = m_u u r + d_v v - m_v\left[-\dot{v}_d\right. \\ \left. + \lambda_\beta(v - v_d) + k_c s_{0\beta} + \tau_{v1_{av}}, \right. \end{aligned} \tag{31}$$

where the sliding surface $s_{0\beta}$ is defined as

$$s_{0\beta} = v - v_d + \lambda_\beta(\beta - \beta_d) \tag{32}$$

and $\tau_{v1_{av}}$ is ruled by the following differential equation, which represents a low-pass filter,

$$\mu_\beta \dot{\tau}_{v1_{av}}(t) = -\tau_{v1_{av}}(t) + \tau_{v1}(t), \tag{33}$$

such that

$$\tau_{v1} = -m_v \epsilon_v \operatorname{sgn}(s_\beta). \tag{34}$$

Finally, the complete sliding surface s_β is defined as

$$s_\beta = s_{0\beta} - s_{0\beta}(\mathbf{e}_\beta(0))e^{-k_c t}. \tag{35}$$

4.1.3 Yaw Angle ψ

Following the same mathematical process as before, it is possible to derive the following expression for τ_r

$$\begin{aligned} \tau_r = -m_{uv} u v + d_r r - m_r\left[-\ddot{\psi}_d\right. \\ \left. + \lambda_\psi(r - \dot{\psi}_d) + k_c s_{0\psi} + \tau_{r1_{av}}, \right. \end{aligned} \tag{36}$$

where the sliding surface $s_{0\psi}$ is defined as

$$s_{0\psi} = r - \dot{\psi}_d + \lambda_\psi(\psi - \psi_d) \tag{37}$$

and $\tau_{r1_{av}}$ is governed by the following differential equation, which represents a low-pass filter,

$$\mu_\psi \dot{\tau}_{r1_{av}}(t) = -\tau_{r1_{av}}(t) + \tau_{r1}(t), \tag{38}$$

such that

$$\tau_{r1} = -m_r \epsilon_\psi \operatorname{sgn}(s_\psi). \tag{39}$$

Finally, the complete sliding surface s_ψ is defined as

$$s_\psi = s_{0\psi} - s_{0\psi}(\mathbf{e}_\psi(0))e^{-k_c t}. \tag{40}$$

5. Simulation

5.1. *C++*/ROS Implementation

The SMC controller developed is implemented using the *C++* language, within the Robot Operating System (ROS) framework. This is done using the Farol stack, which is a set of ROS packages in *C++* and *Python* created and developed by the research team of DSOR, at ISR/IST, and by companies/developers of the robotics community. This stack is public and available to the

whole scientific community in `https://github.com/dsor-isr/farol`. Moreover, the usage of ROS in the stack makes it possible to use the same code both for simulation and real experiments. Furthermore, the stack follows an inner-outer loop structure, which isolates the controller's kinematic and dynamic models, simplifying the development of the software, as follows in Figure 6.

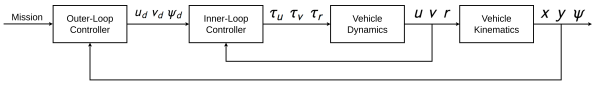


Figure 6: Inner-outer loop structure, considering Simplified Motion as in Section 3.2.

The inner-loop block computes the forces and torques required for the vehicle to follow given certain references, such as surge, sway and yaw. Also, the outer-loop block generates said references, given a certain mission to be accomplished.

5.2. SMC

In order to compare and test the SMC controller in an environment similar to the real world, the controllers presented in Section 4.1 are implemented in the Farol stack as inner-loop controllers, such that this new type of controller is available to all the users in the scientific community in addition to the existing PID controllers.

5.3. Thrust Allocation with Optimisation Methods

The Linear Quadratic Unconstrained Control Allocation method (mentioned in Section 2.3.1) is already implemented in the Farol stack by the research team at DSOR, providing a standard thrust allocation method, used in all subsequent simulations, unless stated otherwise.

As an additional contribution, the Linear Quadratic Constrained Control Allocation method, presented in the aforementioned Section, is also implemented in the Farol stack, making use of CasADi (`https://web.casadi.org/`), an open-source tool for nonlinear optimisation and algorithmic differentiation.

5.4. Results and Discussion - SMC and PID

Several tests in simulation were conducted in order to analyse SMC performance in comparison to PID control. Parameter tuning of the PID controllers used was performed by the research team at DSOR-ISR, which have been used for all vehicle control tests and have proved to be reliable. SMC controllers' parameters are shown in Table 2.

Table 2: SMC's parameters for the *C++*/ROS implementation.

	Yaw - ψ	Surge - u	Sway - v
λ	1.2	0.3	0.3
ϵ	0.1	0.02	0.02
k_c	1.8	0.8	0.8
μ	1 s	1 s	1 s

In order to compare yaw control between both types of controllers, a yaw reference signal $\psi_d(t)$ was devised, as seen in Figure 7, where $\psi_{\text{PID}}(t)$ and $\psi_{\text{SM}}(t)$ are the vehicle's yaw when using PID control and SMC, respectively.

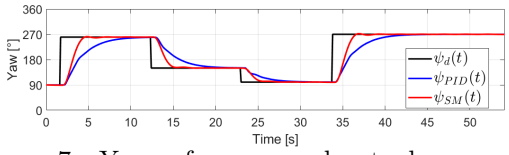


Figure 7: Yaw references and actual responses - comparison between SMC and PID.

Through analysis of Figure 7, it is clear that both controllers succeed in driving the vehicle's yaw to the reference. However, SMC achieves the objective in less time with little to no overshooting, accomplishing better performance.

Now, a surge reference signal u_d was created, as seen in Figure 8, where $u_{\text{PID}}(t)$ and $u_{\text{SM}}(t)$ are the vehicle's surge velocity when using PID control and SMC, respectively.

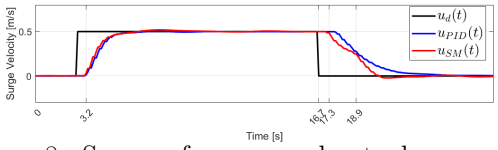


Figure 8: Surge references and actual responses - comparison between SMC and PID.

After thorough analysis of the figure presented, it is clear that both controllers achieve the desired reference for surge velocity. Also, SMC and PID control show similar behaviour when the vehicle is accelerating from 0 m/s to 0.5 m/s, reaching this objective with little overshoot and in a similar time interval.

Moreover, sway velocity controllers are also compared in Figure 9, where $v_d(t)$ is the reference signal and $v_{\text{PID}}(t)$ and $v_{\text{SM}}(t)$ are the vehicle's sway velocity when using PID control and SMC, respectively, achieving similar results to surge's. This is expected since the dynamics of the BlueROV vehicle are similar in the surge and sway axis, as seen in Table 1.

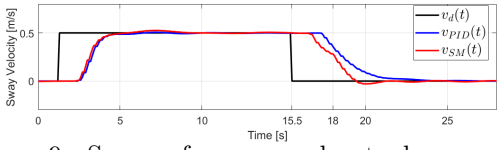


Figure 9: Sway references and actual responses - comparison between SMC and PID.

Furthermore, it is also important to assess both controllers' performance in the presence of external disturbances. Thus, as seen in Figure 10, the yaw controllers' ability to drive the vehicle's heading to a certain reference is tested, by applying an external torque of 1 Nm in the Z axis from $t = 20$ s to $t = 40$ s. Through analysis of Figure 10, it is clear that the PID controller is unable to maintain the vehicle's yaw near the desired reference while it is affected

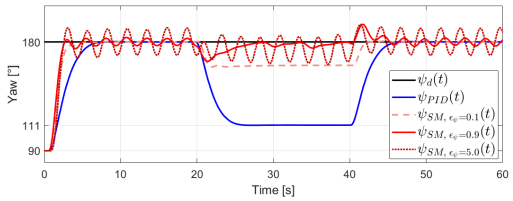


Figure 10: Yaw Controller tracking a constant reference of 180 degrees in the presence of a constant disturbance - comparison between SMC for different values of ϵ_ψ and PID.

by the external disturbance, i.e., between $t = 20$ s and $t = 40$ s, leading to a constant error of 69 degrees.

When it comes to SMC's performance in this case, it is highly affected by its parameter ϵ, which affects the scale of the discontinuous control input of the system, which is designed based on a trade-off between the ability to oppose external disturbances and SMC's natural chattering effect, since both increase with the parameter ϵ. After analysing all three cases, $\psi_{\text{SM},\epsilon_\psi=0.9}$ seems to be the best compromise between the vehicle's behaviour while affected and not affected by an external disturbance.

Now, a star-shaped path is executed with a fixed surge velocity reference of 1.0 m/s, using Aguiar's algorithm (as reviewed by Hung et. al in [9]), which can be observed in Figure 11.

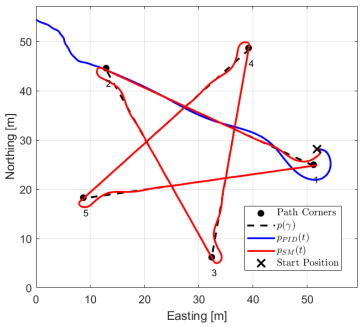


Figure 11: Star Path Following with $u_d = 1.0$ m/s - comparison between SMC and PID.

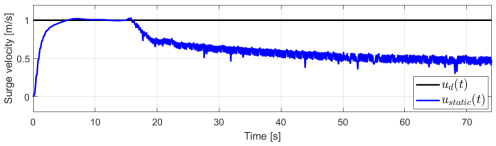
As seen in the figure, there is a significant change of behaviour between both controllers. Besides the better approximation to the path manifested by SMC, it can be observed that, after approaching the second corner of the path, the PID controller is not able to drive the vehicle to the desired path, leaving the vehicle uncontrolled, leading to the end of this mission. Nonetheless, SMC secures its good performance, reaching the end of the path and *turning* around the path's corners with relative small cross-track error.

5.5. Results and Discussion - Unconstrained and Constrained Allocation

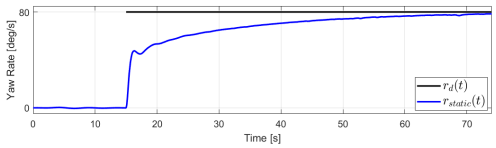
A test is developed to highlight the differences in behaviour between the Linear Quadratic Unconstrained Control Allocation and its constrained counterpart, as previously defined in Section 2.3.1. These two thruster allocation methods are referred to as *static* and *optimised* from now on, respectively, since the former has a closed-form solution and the latter results from an optimisation method.

The developed test consists of using the SMC surge velocity controller and a yaw rate PID controller, the latter previously tuned by the research team at DSOR, in order to reach a surge velocity u of 1.0 m/s and a yaw rate r of 80 deg/s at the same time, which leads to thruster saturation. Essentially, from $t = 0$ s to $t = 15$ s only the surge velocity is being controlled and from $t = 15$ s to $t = 75$ s both the surge velocity and the yaw rate are being controlled.

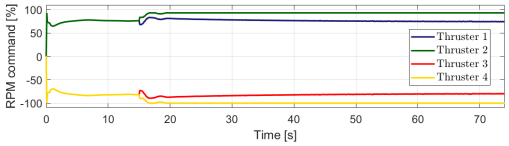
In Figure 12, both surge velocity and yaw rate can be observed when the static thrust allocation method is tested.



Surge velocity.



Yaw Rate.



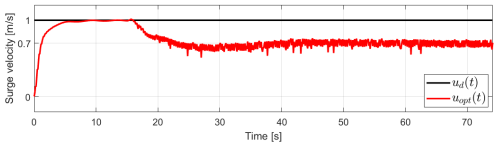
Thrusters' RPM command, in percentage.

Figure 12: Thruster saturation and its impact on the vehicle's behaviour - static thruster allocation.

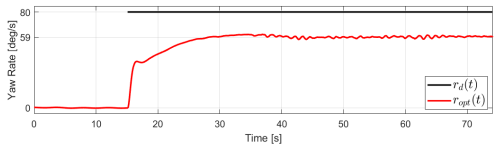
After $t = 15$ s, due to thruster saturation, it can be seen that the vehicle is unable to reach the desired surge velocity, stabilising with an error of approximately 0.5 m/s, while the desired yaw rate is slowly reached. This means that, under these circumstances, the static method leads to prioritising yaw rate tracking performance, in detriment of surge velocity.

As a way of choosing which vehicle DOF to prioritise, the optimised method is also tested under the same circumstances, with results presented in Figure 13. With the purpose of illustrating this property, matrix $Q = \text{diag}\{2000, 200, 200, 200, 200, 20\}$ is devised in order to prioritise the vehicle's surge velocity instead of its yaw rate.

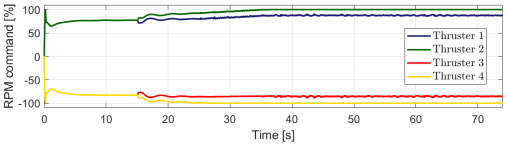
After thorough analysis, it can be seen that, after $t = 15$ s, the yaw rate does not reach its de-



Surge velocity.



Yaw Rate.



Thrusters' RPM command, in percentage.

Figure 13: Thruster saturation and its impact on the vehicle's behaviour - optimised thruster allocation.

sired value and surge velocity ends up stabilising at an error of approximately 0.3 m/s, which is lower compared to the static method. Thus, confirming that optimised thrust allocation gives the ability to choose which vehicle DOF to prioritise when it comes to situations where the vehicle's thrusters saturate.

6. Experimental Results

6.1. Constant References

Several tests are conducted with multiple constant references for yaw and surge and sway velocities. Firstly, SMC's yaw controller is tested, as seen in Figure 14, where ψ_d is the yaw reference signal and ψ_{SM} is the vehicle's yaw.

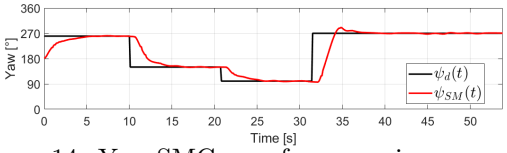
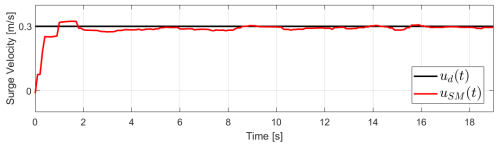


Figure 14: Yaw SMC - performance in response to constant references.

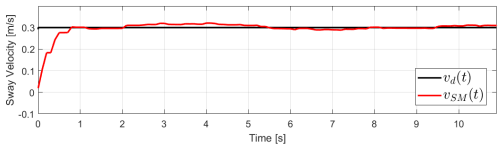
The controller succeeds in driving the vehicle to the desired yaw reference, although the convergence is quite slow, taking around 3 to 4 s to reach it, due to insufficient parameter tuning. Also, the overshoot at $t \approx 35$ s is significant, surpassing the desired reference by 20 degrees.

Moreover, the surge and sway controllers are individually tested, as seen in Figure 15, where u_d and v_d are the references and u_{SM} and v_{SM} are the vehicle's velocities, for surge and sway, respectively.

It can be observed that both controllers achieve the desired references for surge and sway in about 1 s. It is evident that there is some oscillation around the reference after the initial convergence, which



Surge velocity.



Sway velocity.

Figure 15: Surge and sway SMC - performance in response to constant references.

can be associated to the natural SMC chattering, ineffective parameter tuning and environmental disturbances.

6.2. Path Following Missions

Since all three controllers responsible for the system's inner loop demonstrate satisfactory performance, a path following mission is now devised and executed using the Aguiar algorithm. In Figure 16, a lawnmower path is executed by the BlueROV vehicle, where $p(\gamma)$ is the desired path and p_{SM} is the position of the vehicle itself.

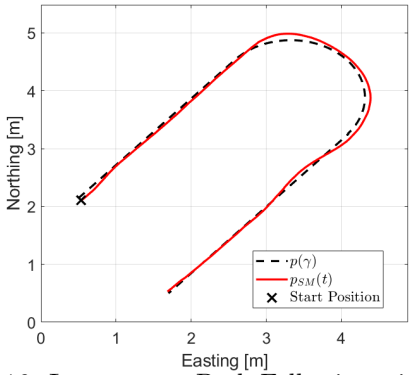


Figure 16: Lawnmower Path Following with profile velocity of 0.1 m/s.

It can be observed that the vehicle is able to follow the desired path with minor cross-track error during the straight sections of the path, while it has a maximum cross-track error of 0.1 m during its semicircular sections, resulting in a natural overshoot afterwards, since there is no sway control in this test. With it, it is predicted that the cross-track error during semicircular sections would be reduced and overshoot severely minimised.

7. Conclusion

To summarise, this work addressed the development of a nonlinear control method for ROV and AUV called SMC, focusing on the craft's surge velocity, sway velocity and yaw. Furthermore, an improved controller formulation was also devised,

targeting SMC's main disadvantage (control input chattering) and fast convergence.

A broad range of simulations was conducted, in order to test SMC controllers in comparison to PID's and equate the performance of two different thrust allocation methods. Firstly, the main objectives of SMC, when compared to the traditionally used PID control, focuses on the rejection of vehicle modelling uncertainty and environmental disturbances, both of which have been proved to be successfully tackled through simulations in *MATLAB* and ROS/*C++*. Likewise, it was also demonstrated that using Constrained Control Allocation over its unconstrained counterpart reflects on an ability to prioritise some vehicle's DOF over others when its thrusters are saturated.

References

[1] Noaa. historical maps and charts audio podcast. national ocean service website. `https://oceanservice.noaa.gov/podcast/july17/nop08-historical-maps-charts.html`.

[2] A. P. Aguiar and J. P. Hespanha. Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *IEEE transactions on automatic control*, 52(8):1362–1379, 2007.

[3] M. S. Dodd, D. Papineau, T. Grenne, J. F. Slack, M. Rittner, F. Pirajno, J. O'Neil, and C. T. Little. Evidence for early life in earth's oldest hydrothermal vent precipitates. *Nature*, 543(7643):60–64, 2017.

[4] B. Ford, A. Borgens, and P. Hitchcock. The 'mardi gras' shipwreck: Results of a deep-water excavation, gulf of mexico, usa. *International Journal of Nautical Archaeology*, 39(1):76–98, 2010.

[5] T. I. Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[6] T. I. Fossen, T. A. Johansen, T. Perez, and A. Inzartsev. A survey of control allocation methods for underwater vehicles. *Underwater vehicles*, pages 109–128, 2009.

[7] H. G. Greene, D. S. Stakes, D. L. Orange, J. P. Barry, and B. H. Robison. Application of a remotely operated vehicle in geologic mapping of monterey bay, california, usa. 1993.

[8] C. Harrold, K. Light, and S. Lisin. Distribution, abundance, and utilization of drift macrophytes in a nearshore submarine canyonsystem. In *Diving for Science... 1993. Proceedings of the American Academy of Underwater Sciences (13th annual Scientific Diving Symposium). Retrieved*, pages 07–11, 2008.

[9] N. Hung, F. Rego, J. Quintas, J. Cruz, M. Jacinto, D. Souto, A. Potes, L. Sebastiao, and A. Pascoal. A review of path following control strategies for autonomous robotic vehicles: theory, simulations, and experiments. *arXiv preprint arXiv:2204.07319*, 2022.

[10] C. Kunz, C. Murphy, R. Camilli, H. Singh, J. Bailey, R. Eustice, M. Jakuba, K.-i. Nakamura, C. Roman, T. Sato, et al. Deep sea underwater robotic exploration in the ice-covered arctic ocean with auvs. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3654–3660. IEEE, 2008.

[11] T.-H. Le, H. L. N. N. Thanh, T.-T. Huynh, M. Van, Q.-D. Hoang, T. D. Do, et al. Robust position control of an over-actuated underwater vehicle under model uncertainties and ocean current effects using dynamic sliding mode surface and optimal allocation control. *Sensors*, 21(3):747, 2021.

[12] J. Liu and X. Wang. *Advanced sliding mode control for mechanical systems*. Springer, 2012.

[13] T. R. Martin. *Ancient Greece: from prehistoric to Hellenistic times*. Yale University Press, 2013.

[14] Y. Pan, Y. H. Joo, and H. Yu. Discussions on smooth modifications of integral sliding mode control. *International Journal of Control, Automation and Systems*, 16(2):586–593, 2018.

[15] M. Pidwirny. Introduction to the oceans. fundamentals of physical geography, date viewed: 2013 march, 2006. `http://www.physicalgeography.net/fundamentals/8o.html`.

[16] F. Pretagostini, L. Ferranti, G. Berardo, V. Ivanov, and B. Shyrokau. Survey on wheel slip control design strategies, evaluation and application to antilock braking systems. *IEEE Access*, 8:10951–10970, 2020.

[17] R. J. Stern. Commentary on jgr-sold earth paper "deep seismic structure across the southernmost mariana trench: Implications for arc rifting and plate hydration" by wan et al. 2019. `https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2019JB017864`.

[18] S. H. Zak. *Systems and control*, volume 198. Oxford University Press New York, 2003.