



Aalborg Universitet

AALBORG UNIVERSITY  
DENMARK

## An Open-Source Benchmark Simulator: Control of a BlueROV2 Underwater Robot

von Benzon, Malte; Liniger, Jesper; Pedersen, Simon; Sørensen, Fredrik Fogh; Jouffroy, Jerome; Uth, Esben Thomsen

*Published in:*  
Journal of Marine Science and Engineering

*DOI (link to publication from Publisher):*  
[10.3390/jmse10121898](https://doi.org/10.3390/jmse10121898)

*Creative Commons License*  
CC BY 4.0

*Publication date:*  
2022

*Document Version*  
Publisher's PDF, also known as Version of record

[Link to publication from Aalborg University](#)

*Citation for published version (APA):*  
von Benzon, M., Liniger, J., Pedersen, S., Sørensen, F. F., Jouffroy, J., & Uth, E. T. (2022). An Open-Source Benchmark Simulator: Control of a BlueROV2 Underwater Robot. *Journal of Marine Science and Engineering*, 10(12), [1898]. <https://doi.org/10.3390/jmse10121898>

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

### Take down policy

If you believe that this document breaches copyright please contact us at [vbn@aub.aau.dk](mailto:vbn@aub.aau.dk) providing details, and we will remove access to the work immediately and investigate your claim.

Article

# An Open-Source Benchmark Simulator: Control of a BlueROV2 Underwater Robot

Malte von Benzon <sup>1,\*</sup> , Fredrik Fogh Sørensen <sup>1</sup> , Esben Uth <sup>1</sup>, Jerome Jouffroy <sup>2</sup>, Jesper Liniger <sup>1</sup>   
and Simon Pedersen <sup>1</sup> 

<sup>1</sup> AAU Energy, Aalborg University, 6700 Esbjerg, Denmark

<sup>2</sup> Department of Mechanical and Electrical Engineering, University of Southern Denmark, SDU, 6400 Sønderborg, Denmark

\* Correspondence: msvb@energy.aau.dk; Tel.: +45-9940-9230

**Abstract:** This paper presents a simulation model environment for the popular and low-cost remotely operated vehicle (ROV) BlueROV2 implemented in Simulink™ which has been designed and experimentally validated for benchmark control algorithms for underwater vehicles. The BlueROV2 model is based on Fossen's equations and includes a kinematic model of the vehicle, the hydrodynamics of vehicle and water interaction, a dynamic model of the thrusters, and, lastly, the gravitational/buoyant forces. The hydrodynamic parameters and thruster model have been validated in a test facility. The benchmark model also includes the ocean current, modeled as constant velocity. The tether connecting the ROV to the top-site facility has been modeled using the lumped mass method and is implemented as a force input to the ROV model. At last, to show the usefulness of the benchmark model, a case study is presented where a BlueROV2 is deployed to inspect an offshore monopile structure. The case study uses a sliding mode controller designed for the BlueROV2. The controller fulfills the design criteria defined for the case study by following the provided trajectory with a low error. It is concluded that the simulator establishes a benchmark for future control schemes for position control and trajectory tracking under the influence of environmental disturbances.

**Keywords:** underwater vehicles; BlueROV2; modeling and control; benchmarking control algorithms



**Citation:** von Benzon, M.; Sørensen, F.F.; Uth, E.; Jouffroy, J.; Liniger, J.; Pedersen, S. An Open-Source Benchmark Simulator: Control of a BlueROV2 Underwater Robot. *J. Mar. Sci. Eng.* **2022**, *10*, 1898. <https://doi.org/10.3390/jmse10121898>

Academic Editors: Mai The Vu and Hyeung-Sik Choi

Received: 28 October 2022

Accepted: 20 November 2022

Published: 5 December 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

The area of remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs) has rapidly developed, with several decades of experience, both in research (see, for example, the study described in [1]) and industrial applications of increasing complexity, both in terms of missions and levels of automation (see for example [2] on cooperative navigation). Regarding the latter, many control algorithms have been proposed, both for simulated and actual vehicles. However, it is generally not immediate to compare control algorithms between them. Indeed, while some articles comparing newly-proposed control laws for ROV stabilization to previous algorithms can easily be found [3–8], it is primarily based on simulation models for which the actual physical replica is not easily accessible or affordable, thereby limiting the comparability between control algorithms and the extent of the benchmarking attempt.

More recently, the industry has also evolved towards more affordable products meant for researchers, students, or hobbyists, several of which are proposed under the open-source paradigm (see, for example, OpenROV, OpenROV Trident, BlueROV2) [9]. The advantages of open software and hardware solutions lie in their cost-effectiveness compared to conventional solutions and that they allow increasingly faster development, modifications, and improvements. Among these solutions, the widely-popular BlueROV2 [10] combines an open-source architecture with components of sufficiently good quality, allowing relatively advanced missions (see, for example, [11–13]).

Several robot simulators have previously been developed, some of which are specific for underwater robots [14–16]. However, the simulators for underwater robots are generally based on a user-defined model, such as the UUV Simulator [14] package for Gazebo or the Simu2VITA [15] block for Simulink™, which makes it challenging to use as a benchmark for control algorithms. Furthermore, these simulators neglect the tether force many underwater robots use for top-site communications or power [17]. The present paper aims at bridging the gap between simulation for control systems and experimental testing by proposing a MATLAB™ and Simulink™ open-source package of the tethered BlueROV2 for the control community as a benchmark in simulation toward full deployment on the same well-known platform. MATLAB™ and Simulink™ have been chosen as they are well-known to the control community and offer various toolboxes to implement advanced control and missions. These missions could be measuring reefs [18], monitoring and removal of marine growth on offshore structures [3,19,20], water quality and ecology monitoring [21], detection of oil plumes [22], fish farm net monitoring [23], etc.

Section 2 will be dedicated to the presentation of the BlueROV2 platform in terms of hardware (frame, thrusters, sensors). In Section 3, we recall the dynamic equations of motions of underwater vehicles propelled by thrusters using the well-known  $\eta$ - $v$  Fossen notation [24]. The parameters of this model are then identified for the particular case of the BlueROV2 in Section 4, and the model is then validated using experimental data. Since the tether attached to the ROV has a significant impact on the dynamics, it is included in the overall model, see Section 6. Section 7 proposes a sliding-mode controller meant as a benchmark for the community to compare against subsequent controllers. Section 8 then starts with the presentation of a modular simulator fully implemented in MATLAB™ and Simulink™ and based on the model from Section 3 and the parameters from Section 4. Appendix B provides a detailed guide to using the simulator. Section 9 examines a case study on offshore monopile inspection to demonstrate the use of the proposed simulator. Finally, a few concluding remarks end the paper. The proposed simulator can be found at [Github](#) in Supplementary Materials, accessed on 1 October 2022.

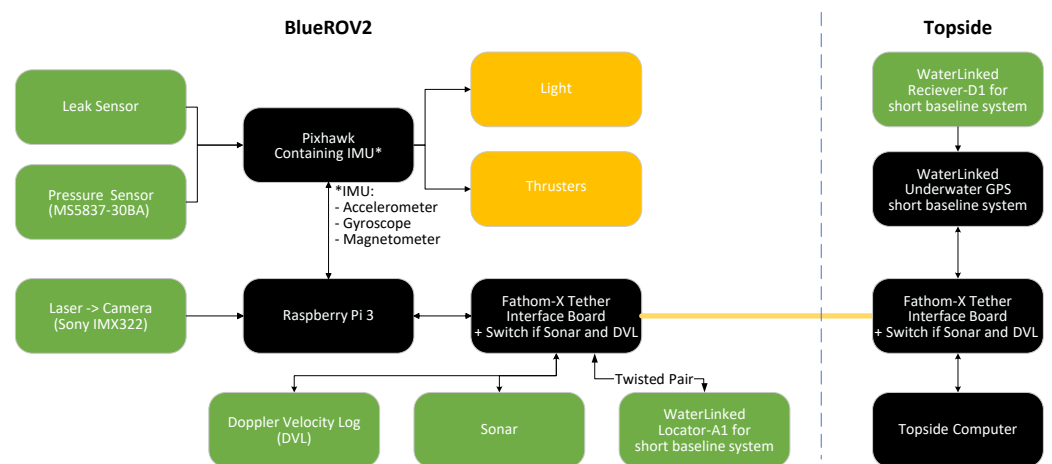
## 2. BlueROV2: An Open-Source Underwater Vehicle

The BlueROV2 is a tethered ROV. The tether is used for communication between the vehicle and a top-side unit. The vehicle can be powered either through the tether or by an onboard battery. The ROV comes with various onboard sensors, including a pressure transmitter, leak detection sensors, a power sense module, an Inertial Measurement Unit (IMU), and a magnetometer. The IMU consists of an accelerometer and a gyroscope. The IMU and magnetometer are integrated in a Pixhawk. The open-source software architecture allows adding additional sensors such as an Underwater Acoustic Positioning System (UAPS), Doppler Velocity Log (DVL), camera vision, sonar, lasers, etc. Software implementation of the sensors on the physical hardware can be done on the Pixhawk through ArduSub or on the Raspberry Pi using the MAVLink protocol. However, the simulator is based on MATLAB™ and Simulink™, and therefore implementation on the physical hardware is not considered. A list of common sensors and their properties is shown in Table 1, where the associated state measurements are given according to the notation used in the model presented in the next section.

The main control board on the BlueROV2 is a Raspberry Pi 3, which is connected to a top-side computer through an Ethernet connection. Besides the Raspberry Pi 3, the electronic tube contains a Pixhawk. The Ethernet communication is transferred top-side through a Fathom-X Tether Interface, which provides a long-distance Ethernet connection over a single twisted pair of wires. The hardware and interfaces of the BlueROV2 is shown in Figure 1.

**Table 1.** List of common sensors used for the BlueROV2.

Sensor Technology	State	Sample Rate	Delay	Variance (Filtered)
Pressure Sensor	D	20 Hz	-	$1.5 \times 10^{-5}$ m
Inertial Measurement Unit	$\phi, \theta$	1000 Hz	-	$1 \times 10^{-5}$ rad to $2.5 \times 10^{-5}$ rad
Magnetometer/compass	$\psi$	1000 Hz	-	
Laser-Camera	x,y	20 Hz	-	$2 \times 10^{-6}$ m
Sonar	x,y	40 Hz	-	
Short Baseline	N,E	4 Hz	1.2 s	$1 \times 10^{-3}$ m to $1.3 \times 10^{-3}$ m
Doppler Velocity Log	u,v	15 Hz	0.1 s	0.3 cm to 0.8 cm



**Figure 1.** Hardware and interfaces on BlueROV2, green boxes represents sensors, black boxes represents processing units/control units, the yellow boxes represents actuators, and the dashed blue line illustrates the cut between onboard hardware and topside hardware.

Blue Robotics Inc.’s heavy configuration kit is installed as additional equipment on the BlueROV2, which includes two additional T200 thrusters, such that the thrusters can be paired to move in all degrees of freedom. This modification is illustrated in Figure 2. The difference in thruster allocation between the normal and the heavy configuration is shown in Figure 3 and Table 2, the most noticeable difference is that the heavy configuration is controllable in pitch. Tools can be installed for specific operations, such as a gripper (purchased together with the BlueROV2), a water jet [3], etc.

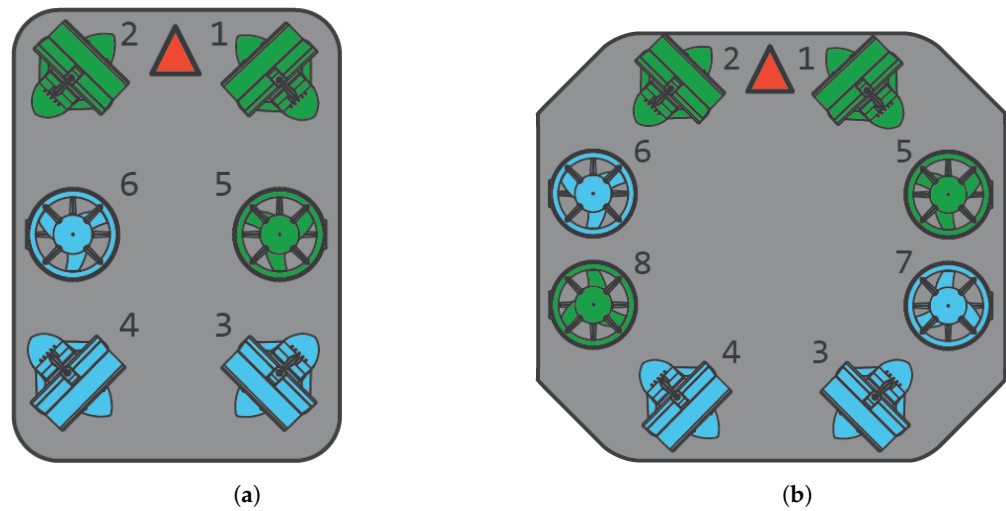


BlueROV2 Normal Configuration



BlueROV2 Heavy Configuration

**Figure 2.** Two illustrations of the BlueROV2 with and without heavy configuration, respectively.



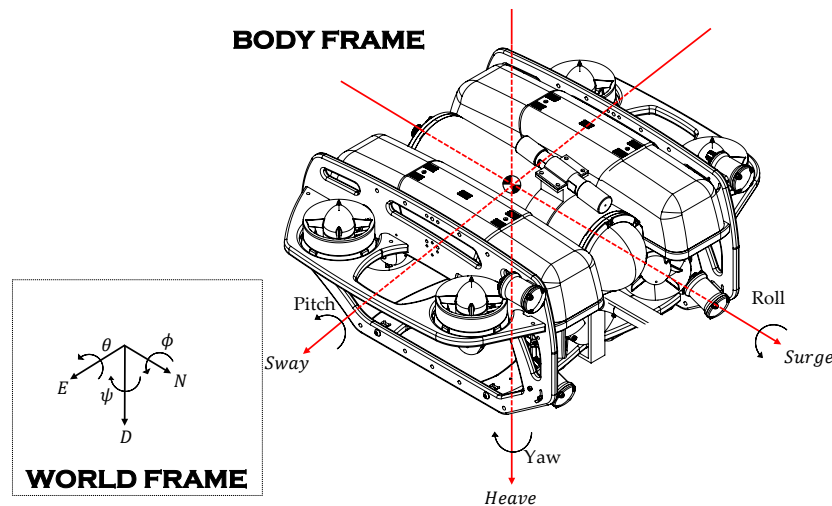
**Figure 3.** A comparison of the thruster configuration of the normal and heavy BlueROV2. The blue thrusters are the clockwise thrusters, and the green are the counterclockwise thrusters. The red arrow is the positive surge direction [25]. (a) BlueROV2 normal configuration. (b) BlueROV2 heavy configuration.

**Table 2.** Difference between thruster combination to motion on normal and heavy configuration.

Motion	Normal BlueROV2	Heavy BlueROV2
Surge	T1, T2, T3 and T4	T1, T2, T3 and T4
Sway	T1, T2, T3 and T4	T1, T2, T3 and T4
Heave	T5 and T6	T5, T6, T7 and T8
Roll	T5 and T6	T5, T6, T7 and T8
Pitch	Uncontrollable	T5, T6, T7 and T8
Yaw	T1, T2, T3 and T4	T1, T2, T3 and T4

### 3. Mathematical Modeling

In the following, we will distinguish between the body or *b*-frame and world or *n*-frame as illustrated in Figure 4, where the respective degrees of freedom are visualized in each frame.



**Figure 4.** The reference and world frame with all degrees of freedom in each respective frame.

In this section, we recall the standard mathematical model of an underwater vehicle, to which the BlueROV2 applies. This model is based on the following few assumptions:

**Assumption 1.** The vehicle is assumed to be rigid, and 6 degrees of freedom are considered.

**Assumption 2.** The ROV is assumed symmetric around the front-back, port-starboard and the top-bottom axes.

**Assumption 3.** The body axes coincide with the principles axes of inertia.

**Assumption 4.** The origin of the  $b$ -frame is located at the center of mass of the vehicle.

**Remark 1.** The Assumptions 1–4 are common when modeling underwater vehicles [26].

**Assumption 5.** The ocean current is modeled as a constant irrotational flow in the  $n$ -frame. The current is projected on the ROV as a change in velocity. Waves are neglected.

**Remark 2.** The Assumption 5 is a simplified model of the current, as ocean currents are mainly caused by tidal movements and can be dependent on both the local climate and the geographic characteristics [27]. Furthermore, the current velocity varies both with space and time. However, from a control perspective where the current acts as a disturbance to the system, the time dependency can be neglected due to the relatively slowly varying current compared to the dynamics of the controlled system [26].

In the following, we will use the well-known compact  $\eta - v$  Fossen notation (first proposed in [28]), itself gathering in a vectorial form elements of the SNAME nomenclature (see [29,30], and a list of the variables and parameters used hereafter summarized in Table 3). It is given by

$$\begin{aligned} \dot{\eta} &= J(\eta)v & (1) \\ M\dot{v} + C(v)v + D(v)v + g(\eta) &= \tau + \tau_{tet} & (2) \end{aligned}$$

where  $v = [u, v, w, p, q, r]^T$  is the vector of body-fixed velocities (surge, sway, heave, roll, pitch and yaw) and  $\eta = [x, y, z, \phi, \theta, \psi]^T$  is the vector combining the positions and Euler angles with respect to the  $n$ -frame. The vector  $\tau = [X, Y, Z, K, M, N]^T$  contains the forces and moments applied to the vehicle.

**Table 3.** Definitions of variables.

Notation	Description
$C, C_{RB}, C_A$	System/rigid-body/added-mass Coriolis -centripetal matrix
$D$	Vector of damping forces
$F, M$	External forces and moments acting on the vehicle in the $b$ -frame
$g$	Vector of restoring forces
$J$	Coordinate transformation matrix
$M, M_{RB}, M_A$	System/rigid-body/added-mass inertia matrix
$m$	Total mass of the vehicle
$p, q, r$	Angular velocities in the $b$ -frame
$u, v, w$	Linear velocities in the $b$ -frame
$v$	Linear velocity vector in the $b$ -frame
$X, Y, K, N_{\dot{u}-\dot{r}}$	Added mass coefficients in the $b$ -frame
$x, y, z$	Translation in the $n$ -frame
$x_b, y_b, z_b$	Body-fixed reference frame ( $b$ -frame)
$\eta$	Vector of position and orientation in the $n$ -frame
$v$	Velocity vector in the $b$ -frame
$\rho$	Water density
$\tau$	Vector of control inputs
$\phi, \theta, \psi$	Euler’s angles in the $n$ -frame
$\omega$	Angular velocity vector in the $b$ -frame

Equation (1) represents the kinematic equations of motion, where

$$J(\eta) = \begin{bmatrix} J_1(\eta) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & J_2(\eta) \end{bmatrix}, \tag{3}$$

and

$$J_1(\eta) = \begin{bmatrix} \cos \psi \cos \theta & -\sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi & \sin \psi \sin \phi + \cos \psi \cos \phi \sin \theta \\ \sin \psi \cos \theta & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & -\cos \psi \sin \phi + \sin \theta \sin \psi \cos \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \tag{4}$$

$$J_2(\eta) = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix}, \tag{5}$$

where let us recall that  $J_2(\eta)$  is not defined when  $\theta = \{-\pi, \pi\} [2\pi]$ .

The constant, symmetric and strictly positive definite inertia matrix  $M = M_{RB} + M_A$  combines the rigid-body inertia matrix  $M_{RB}$  given by

$$M_{RB} = \begin{bmatrix} m\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_c \end{bmatrix} \tag{6}$$

with  $\mathbf{I}_c = \text{diag}\{I_x, I_y, I_z\}$ , and the added-mass and inertia matrix, which we assume to be diagonal containing the added-mass coefficients, so that we have

$$M_A = -\text{diag}\{X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}\}. \tag{7}$$

Skew-symmetric Coriolis matrix  $C(v) = C_{RB}(v) + C_A(v)$  also combines rigid-body and hydrodynamics effects, with

$$C_{RB} = \begin{bmatrix} 0 & 0 & 0 & 0 & m\omega & -mv \\ 0 & 0 & 0 & -m\omega & 0 & mu \\ 0 & 0 & 0 & m\omega & -mu & 0 \\ 0 & m\omega & -m\omega & 0 & -I_z r & -I_y q \\ -m\omega & 0 & mu & I_z r & 0 & I_x p \\ m\omega & -mu & 0 & I_y q & -I_x p & 0 \end{bmatrix} \tag{8}$$

and

$$C_A = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}\omega & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}\omega & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}\omega & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}\omega & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix} \tag{9}$$

At low-speed, the hydrodynamic damping matrix  $D(v) = D + D_n(v)$  is considered without coupling, so that we have the diagonal matrices containing the linear and nonlinear damping coefficients

$$D = -\text{diag}(X_u, Y_v, Z_w, K_p, M_q, N_r) \tag{10}$$

and

$$D_n(v) = -\text{diag}X_{u|u}|u|, Y_{v|v}|v|, Z_{w|w}|w|, K_{p|p}|p|, M_{q|q}|q|, N_{r|r}|r|, \tag{11}$$

the latter diagonal matrix representing quadratic damping.

Taking into account Assumption 4, the vector of restoring forces (gravitational and buoyancy forces)  $g(\eta)$  is expressed as

$$g(\eta) = \begin{bmatrix} (W - B) \sin \theta \\ -(W - B) \cos \theta \sin \phi \\ -(W - B) \cos \theta \cos \phi \\ y_b B \cos \theta \cos \phi - z_b B \cos \theta \sin \phi \\ -z_b B \sin \theta - x_b B \cos \theta \cos \phi \\ x_b B \cos \theta \sin \phi + y_b B \sin \theta \end{bmatrix}. \tag{12}$$

where  $W = mg$  and  $B = \rho g \nabla$  are the gravity and buoyancy forces, respectively ( $\rho$  the water density and  $\nabla$  the volume of fluid displaced by the vehicle), and  $(x_b, y_b, z_b)$  are the coordinates of the center of buoyancy expressed in the  $b$ -frame.

An ROV like the BlueROV2 is typically actuated by a number  $r$  of fixed thrusters. In order to link the vector  $\tau$  to the voltages  $V_i$  applied to each thruster  $i$  ( $i \in \{1, \dots, r\}$ ), we use the expression

$$\tau = TK(s)F(V) \tag{13}$$

with  $V = [V_1, \dots, V_r]^T$  denoting thruster voltages in column vector  $F(V)$ . Matrix  $T \in \mathbb{R}^{6 \times r}$  is the so-called thrust configuration matrix (see [24]):

$$T = [t_1, \dots, t_r] \tag{14}$$

thereby, we obtain the thruster mixing matrix, where each column vector  $t_i$  links the force  $F_i$  created by each thruster to vector  $\tau$ :

$$t_i = \begin{bmatrix} \varepsilon_i \\ r_i \times \varepsilon_i \end{bmatrix} \tag{15}$$

where  $\varepsilon_i$  is a unit vector representing the orientation of thruster  $i$  and  $r_i$  is the distance between the point of application of the force  $F_i$  and the center of the  $b$ -frame.

The diagonal matrix  $K(s)$  contains unity DC-gain transfer functions accounting for the dynamic relation between  $V_i$  and the force  $F_i$ , i.e., we have

$$K(s) = \text{diag}(K_1(s), \dots, K_r(s)). \tag{16}$$

#### 4. Parameter Identification and Model Validation

This section covers the identification procedure and validation of model parameters listed in Table A1. Table A1 specifies whether the parameter is determined from experiments, from a CAD-model, or obtained from other sources.

##### 4.1. Thruster Model

The BlueROV2 in heavy configuration is built with eight T200 thrusters. The numbering and loci follow the definition of the manufacturer [31]. The thruster orientation,  $\varepsilon_i$ , location,  $r_i$ , is given in Table A1. The thrusters are located in a circular pattern and the rotation matrix used is denoted

$$J_3(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

A nonlinear regression to normalized input voltages and measured force,  $F(V)$ , is obtained from experiments conducted in prior work [4]. The regression given for  $V_i \in [-1, 1]$  is

$$F_i(V_i) = -140.3V_i^9 + 389.9V_i^7 - 404.1V_i^5 + 176.0V_i^3 + 8.9V_i \tag{18}$$



The thruster dynamics is described by the diagonal transfer function matrix  $K(s) = \text{diag}(K_1(s), \dots, K_r(s))$ . The diagonal elements are equal and obtained from system identification on experimental data published by MathWorks [32]. The following third order transfer function with unity DC-gain was identified

$$K_i(s) = \frac{6136s + 108700}{s^3 + 89s^2 + 9258s + 108700} \tag{19}$$

The maximum velocities and forces of each direction of motion are shown in Table 4.

**Table 4.** Maximum velocities and forces in *b*-frame.

Movement in <i>b</i> -Frame	Maximum Velocity	Maximum Thrust Force
$\dot{x}_{max}$	0.72 ms <sup>-1</sup>	85 N
$\dot{y}_{max}$	0.63 ms <sup>-1</sup>	85 N
$\dot{z}_{max}$	0.71 ms <sup>-1</sup>	120 N
$p_{max}$	5.2 ms <sup>-1</sup>	26 Nm
$q_{max}$	4.9 ms <sup>-1</sup>	14 Nm
$r_{max}$	3.6 ms <sup>-1</sup>	22 Nm

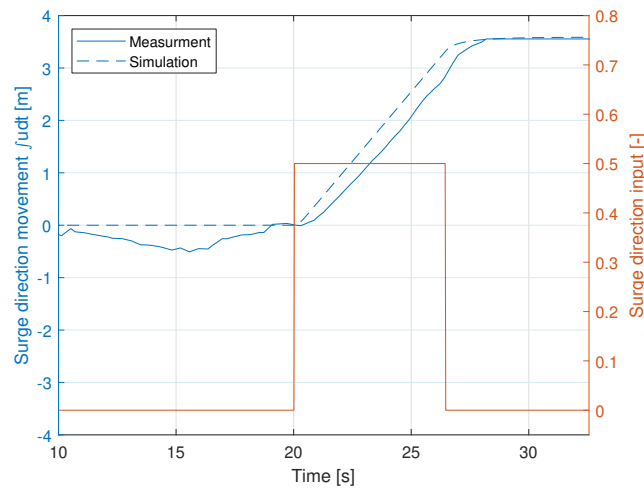
#### 4.2. Hydrodynamic Damping

The hydrodynamic damping coefficients given in Equations (10) and (11) are determined in a comparative study of the model output and from a series of experiments conducted in a 1.15 m deep 6.1 m diameter pool. The experiments are designed such that the ROV is actuated individually and in open-loop in all 6-DOF, one at a time, at an interval of several input voltages. Each experiment is conducted twice, where the first instance is used for estimating coefficients and the latter is used for validation. Rather than actuating the thrusters individually, the thrusters are actuated using the standard ArduSUB software allowing for actuating along the 6-DOF directly. The individual thruster input is logged during the experiments and used as input to the model.

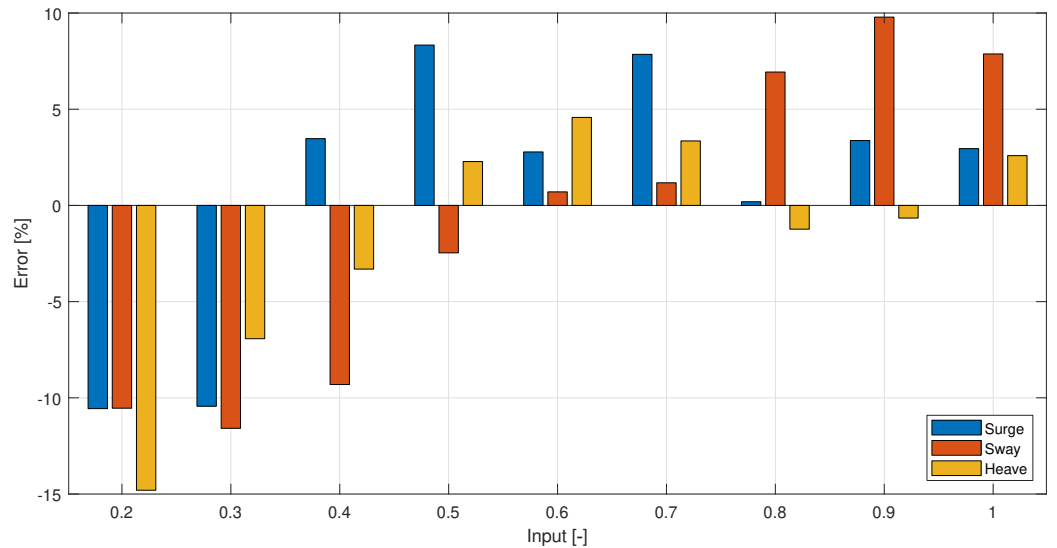
The ROV is actuated in each direction of motion from just above the deadzone of the thrusters at 20% input to 100% input, with a step-wise increase of 10%. As an example of one of these tests, Figure 5 compares the experimental data and the model output given 50% input in the surge direction. Position feedback is obtained by using a UAPS Short BaseLine (SBL) location system. The disturbance from tether forces is reduced by manually maneuvering the tether in the direction of the ROV, but it has not been possible to eliminate this disturbance entirely. As a consequence, a proportional feedback controller is used for regulating to a constant depth to ensure that the ROV does not hit the pool bottom or water surface during the surge and sway-direction experiments.

As seen in Figure 5, the thrusters are activated after 20 s. The quasi-static part of the motion is used for determining the velocity and consequently the difference between the simulation model and experimental measurements. Figure 6 shows the error for an interval of inputs from just above the deadzone of the thrusters to maximum input, from 20% to 100% input. The largest model discrepancy for both the surge and sway direction of motion is approximately 10%.

Hydrodynamic coefficients related to the heave motion are estimated and validated by performing an experiment where the ROV is submerged at the bottom of the pool and actuating thrusters in the negative heave direction. The heave motion is determined using the pressure sensor. Figure 6 shows that the heave motion model error for the thrust input of 20% is 15% and that the error reduces for larger inputs.



**Figure 5.** Simulation model and experimental measurements for the surge-direction with positive 50% open-loop input. The error between these is shown as the blue bar at an input of 0.5 in Figure 6.



**Figure 6.** Model and experimental data comparison for surge, sway, and heave. Each of the bars represents a test similar to the one shown in Figure 5.

Rotational motion in pitch, roll, and yaw is validated using the integrated Inertial Measurement Unit (IMU) for angular velocity measurement. It is necessary to fully rotate the ROV during the rotational drag tests. This requires an input of more than 30%. Therefore, for the rotational motions, the input given is from 30% to 100% with a step-wise increase of 10%. The model discrepancy for pitch, roll, and yaw is seen in Figure 7. Roll motion error is seen to increase at higher inputs to approximately 10%. The largest discrepancy between the model and experiment for pitch and yaw rotation is 10% and 12%, respectively.

The error bars shown in Figures 6 and 7 show the error between the corrected simulation model and the validation data. There is no clear pattern in the error to further optimize the second order polynomial model. If the hydrodynamic model should be further improved, this requires more precise measurements along with more tests to obtain knowledge about the tendency of the error. This has not been deemed necessary.

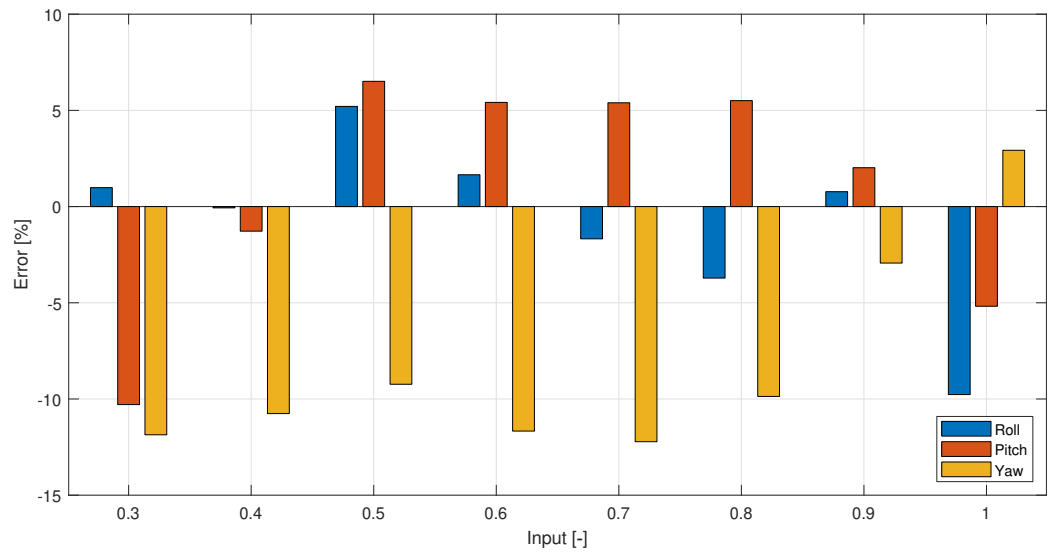


Figure 7. Model and experimental data comparison for roll, pitch, and yaw. Each of the bars represents a test similar to the one shown in Figure 5.

#### 4.3. Added Mass

Added mass coefficients presented in Equation (7) are estimated from Eidsvik method [33], which builds upon the experimental work outlined by Det Norske Veritas (DNV) [34]. Eidsvik shows that the relative error of the empirical estimations in translational and rotational degrees of freedom are 10–20% and 30–100%, respectively. The method requires overall dimensions and projected areas of the ROV. The method also requires that the difference in overall dimensions is not larger than 10%. The BlueROV2 in heavy configuration violates this requirement as the width is 20% larger than the length, and the estimated added mass coefficients may be susceptible to further error than predicted by Eidsvik. This will lead to a difference between the model and the physical robot in the acceleration phase. The overall dimensions and projected areas are given in Table 5 and are found using a CAD model of the ROV.  $L$ ,  $H$ , and  $W$ , denote the length, height, and width. The notation for the projected areas is given such that the subscript shows the normal direction to the area in the  $b$ -frame. The resulting added mass coefficients are found in Table A1.

Table 5. Overall dimensions and projected areas of the BlueROV2 heavy configuration.

Notation	Value	Unit
$L \times H \times W$	$0.46 \times 0.38 \times 0.58$	m
$A_u$	0.0877	m <sup>2</sup>
$A_v$	0.1131	m <sup>2</sup>
$A_w$	0.2049	m <sup>2</sup>

### 5. Ocean Current

The current effect is implemented as an additional velocity term that does not change in the water column. The velocity term in the equations of motions given in Equation (2) is thereby transformed to be in terms of relative velocity shown in Equation (20),

$$(\mathbf{M}_{rb}\dot{\mathbf{v}} + \mathbf{C}_{rb}(\mathbf{v})\mathbf{v} + \mathbf{g}(\boldsymbol{\eta})) + (\mathbf{M}_A\dot{\mathbf{v}}_r + \mathbf{C}_A(\mathbf{v}_r)\mathbf{v}_r + \mathbf{D}(\mathbf{v}_r)\mathbf{v}_r) = \boldsymbol{\tau} + \boldsymbol{\tau}_{tet} \quad (20)$$

where the relative velocity is given by  $\mathbf{v}_r = \mathbf{v} - \mathbf{v}_c$ . As it is assumed that the current is irrotational, the vector for the water particles velocity in  $b$ -frame  $\mathbf{v}_c$  becomes

$$\mathbf{v}_c = \mathbf{R}\mathbf{v}_c \quad (21)$$

where,

$$\mathbf{v}_c = (\dot{u}_c, \dot{j}_c, \dot{w}_c, 0, 0, 0)^\top \tag{22}$$

where  $\dot{u}_c, \dot{j}_c, \dot{w}_c$  is the  $n$ -frame water particle velocity in north, east and down direction, respectively. As the current is represented in  $n$ -frame, this velocity can be transformed into  $b$ -frame by the inverse rotation matrix such that it can be implemented in the model represented in  $b$ -frame.

From Equation (8), it is shown that  $\mathbf{C}_{rb}$  is parameterized independent of linear velocity, and it is assumed that the current is slowly varying, so that the velocity can be assumed to be constant. Therefore, Equation (20) can be reduced to

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v}_r)\mathbf{v}_r + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{tet} \tag{23}$$

### 6. Tether Model

In most underwater applications, the tether is used for communication and power from the top-side, therefore it is a necessity. However, the tether also disturbs the ROV due to the forces generated by the drag and underwater currents [35]. The proposed simulator also includes the dynamics of the tether linking the BlueROV2 underwater vehicle to the top-site base. In order to model the tether, we use a lumped-mass approach, whereby the cable is idealized as a set of  $n + 1$  lumped masses connected together by  $n$  weightless cylinders [36,37].

Let  $\mathbf{p}_i \in \mathbb{R}^3, i \in \{0, \dots, n\}$  represent the position of node/lumped mass  $i$  in the  $n$ -frame, and  $\mathbf{v}_i := \dot{\mathbf{p}}_i$  the corresponding velocity vector. In this setting,  $\mathbf{p}_0$  is directly linked to the above-mentioned base, whose position is possibly time-varying, i.e.,  $\mathbf{p}_0 = \mathbf{p}_b(t)$ , while  $\mathbf{p}_n$  is attached to the ROV. Hence, regarding the tether dynamics, let us group the remaining  $\mathbf{p}_i$ s and  $\mathbf{v}_i$ s into the vectors

$$\mathbf{p}_t = [\mathbf{p}_1^T, \mathbf{p}_2^T, \dots, \mathbf{p}_{n-1}^T]^T \tag{24}$$

and

$$\mathbf{v}_t = [\mathbf{v}_1^T, \mathbf{v}_2^T, \dots, \mathbf{v}_{n-1}^T]^T. \tag{25}$$

The dynamics associated with the tether position and velocity vectors  $\mathbf{p}_t$  and  $\mathbf{v}_t$  are summarized by the mass-spring-damper-like expression

$$\mathbf{M}_t\dot{\mathbf{v}}_t + \mathbf{D}_t\mathbf{v}_t + \mathbf{k}_t = 0 \tag{26}$$

where  $\mathbf{M}_t$  is a diagonal matrix of  $n - 1$  elements

$$\mathbf{M}_{t,i} = \frac{1}{2}(\mathbf{M}_{i-1} + \mathbf{M}_i + \mathbf{M}_{a,i-1} + \mathbf{M}_{a,i}) \tag{27}$$

where  $M_i \in \mathbb{R}^{3 \times 3}$  is the mass matrix of node  $i$  and  $M_{a,i} \in \mathbb{R}^{3 \times 3}$  is the added mass matrix of the corresponding cylinder. Similarly, the damping terms of diagonal matrix  $\mathbf{D}_t$  are given by

$$\mathbf{D}_{t,i} = \mathbf{P}_{i-1} - \mathbf{P}_i + \mathbf{F}_i \tag{28}$$

where  $\mathbf{P}_i$ , the internal damping of the tether, is expressed as

$$\mathbf{P}_i = C_t \left[ (\mathbf{v}_i - \mathbf{v}_{i-1})^T \frac{\mathbf{r}_i}{|\mathbf{r}_i|} \right] \frac{\mathbf{r}_i}{|\mathbf{r}_i|} \tag{29}$$

where  $C_t$  is a damping constant, and  $\mathbf{r}_i := \mathbf{p}_{i+1} - \mathbf{p}_i$  is the vector between two consecutive nodes.

The tether being assumed to be neutrally buoyant, the external force  $\mathbf{F}_i$  acting on element  $i$  of the tether represents only hydrodynamic drag. Then, force  $\mathbf{F}_i$  is the sum of its normal and tangential components

$$\mathbf{F}_i = \mathbf{F}_{\perp,i} + \mathbf{F}_{\parallel,i} \tag{30}$$

where

$$\mathbf{F}_{\perp,i} = \frac{1}{2} \rho d_t C_{\perp} \mathbf{v}_{\perp,i} |\mathbf{v}_{\perp,i}| |\mathbf{r}_i| \tag{31}$$

and

$$\mathbf{F}_{\parallel,i} = \frac{1}{2} \rho d_t C_{\parallel} \mathbf{v}_{\parallel,i} |\mathbf{v}_{\parallel,i}| |\mathbf{r}_i|, \tag{32}$$

where  $d_t$  is the diameter of the tether,  $\mathbf{v}_{\perp,i}$  and  $\mathbf{v}_{\parallel,i}$  are the normal and tangential components of the flow velocity of the  $i$ -th cylinder, while  $C_{\perp}$  and  $C_{\parallel}$  are their respective drag coefficients. The normal and tangential velocity components are obtained through the expressions

$$\mathbf{v}_{\perp,i} = \frac{[(\mathbf{v}_c - \mathbf{v}_i)^T \mathbf{r}_i] \mathbf{r}_i}{|\mathbf{r}_i|^2} \tag{33}$$

with  $\mathbf{v}_c$  representing the water current velocity, expressed in the  $n$ -frame and

$$\mathbf{v}_{\parallel,i} = \mathbf{v}_c - \mathbf{v}_i - \mathbf{v}_{\perp,i}. \tag{34}$$

The term  $\mathbf{k}_t$  in (26) is given by

$$\mathbf{k}_{t,i} = \mathbf{T}_i \tag{35}$$

where the axial tension  $\mathbf{T}_i$  of the  $i$ -th node is given by

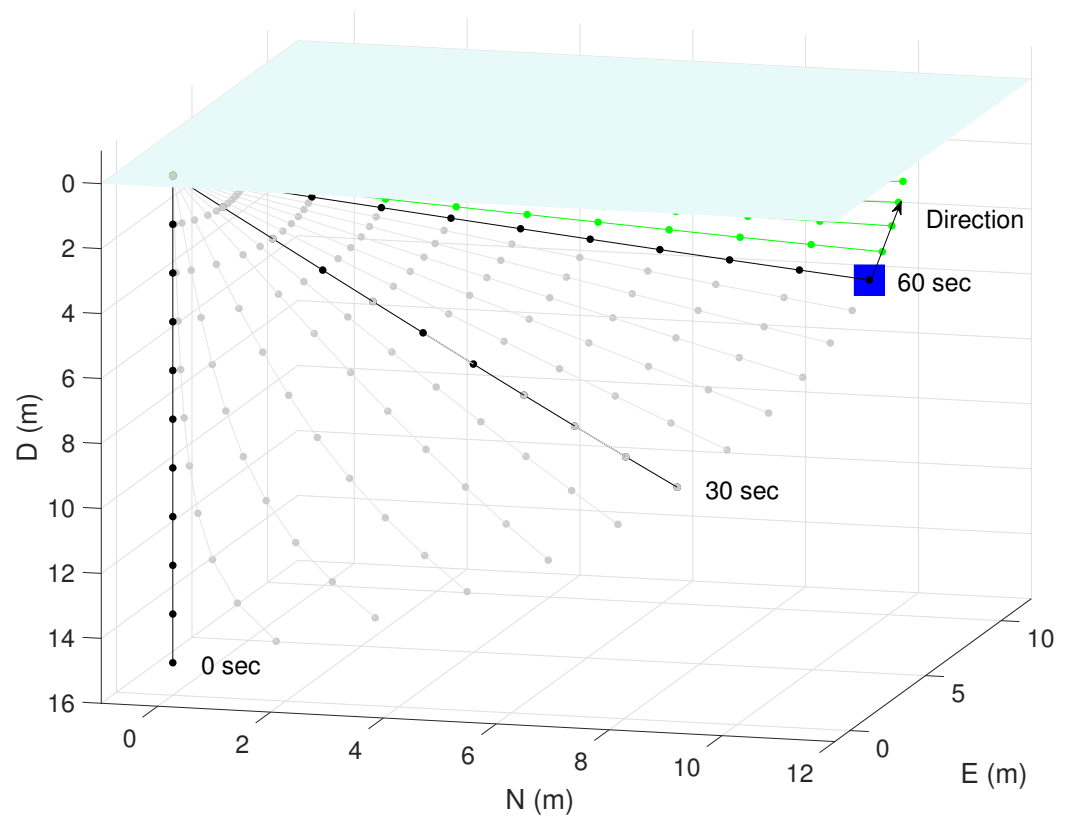
$$\mathbf{T}_i = E_t \left( \frac{A_{t,i}}{l_{0,i}} \right) \mathbf{r}_i \left( 1 - \frac{l_{0,i}}{|\mathbf{r}_i|} \right), \tag{36}$$

The force acting on the ROV  $\boldsymbol{\tau}_{tet}$  is the end node of the axial tension.

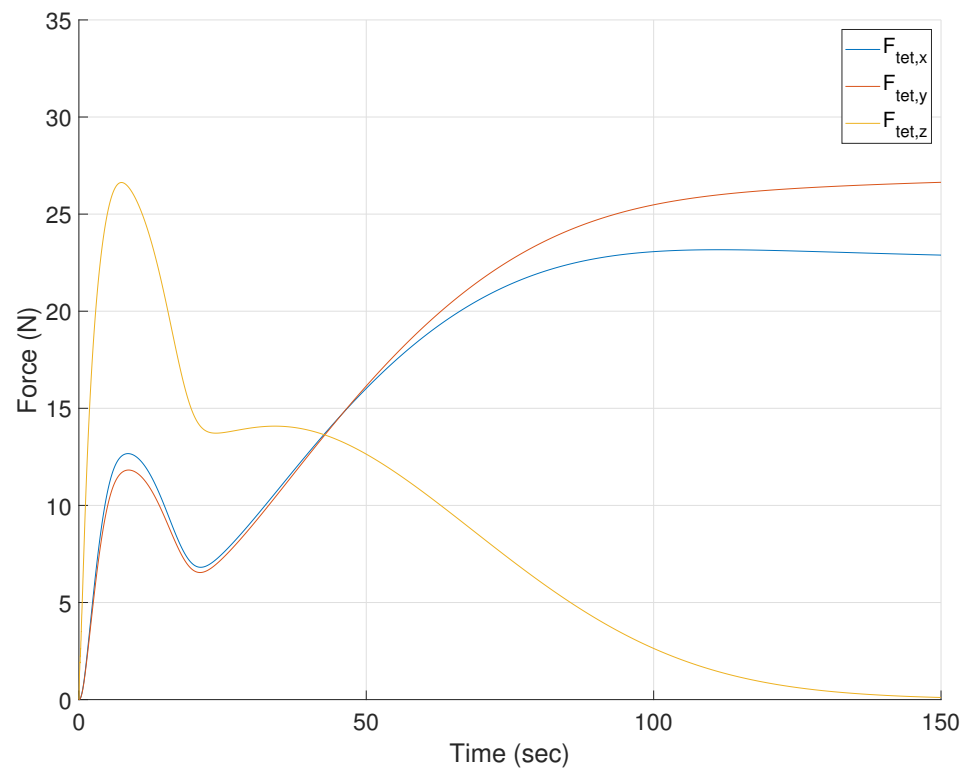
#### Tether ROV Interaction

In the following, the ROV and tether interaction is demonstrated by letting the tethered ROV be free floating with a current velocity of  $[0.3536, 0.3536, 0]$  in the  $n$ -frame and initialized at  $[0, 0, 15]$ . The tether is 15 m in length and with the tether parameters from Table A1. In Figure 8, a 3D plot of the ROV and tether is shown. This shows that the tether is initially lagging behind the ROV, but straightens out after 30 s. After 100 s, the ROV is approximately at the surface, and after 150 s, the force in  $x,y,z$  is constant, which is shown in Figure 9. When comparing the forces from Figure 9 to the maximum thrust force of the ROV seen in Table 4, it is clear that the tether has a large effect on the ROV. The movement in the Down-North and Down-East directions are shown in Figure 10. The range from 100–150 s is left out due to the change in position being minor. From Figure 10, it can be seen that the ROV moves further East compared to North, which is due to the drag force being larger.

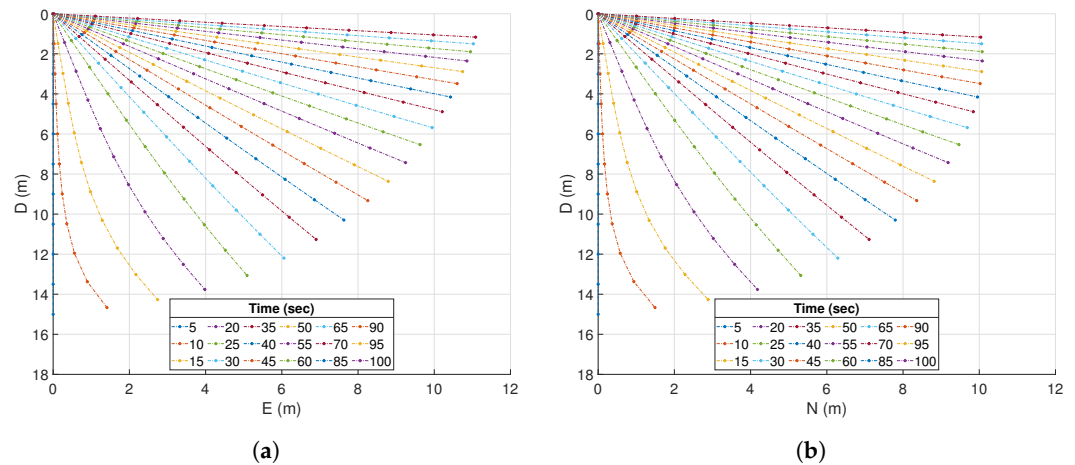
A video of the free floating ROV and tether with a constant current velocity of  $[0.3536, 0.3536, 0]$  in the  $n$ -frame can be found in the attached video named “Tether-ROV interaction” or at [Github](#) in Supplementary Materials.



**Figure 8.** Free floating ROV and tether with a current velocity of  $[0.3536, 0.3536, 0]$  in the  $n$ -frame. Initialized at  $[0, 0, 15]$  in  $x, y$  and  $z$ , respectively. The blue box represents the ROV at position  $[9.68, 9.94, 5.70]$ . The faded black lines shows the trajectory of the tether and ROV and the green lines show the future path.



**Figure 9.** Tether force in  $n$ -frame on the free floating ROV, current velocity of  $[0.3536, 0.3536, 0]$  in the  $n$ -frame.



**Figure 10.** Cross-section of the tether and ROV in the North (b), East (a) and Down direction from time 0 to 100 s.

### 7. Control Algorithms for Benchmarking

#### 7.1. Sliding Mode Control

Sliding mode control (SMC) is a popular choice for control of underwater vehicles due to its ability to deal with nonlinear systems subject to uncertainties and bounded disturbances [38]. Moreover, it requires relatively low computer power to compute. The objective, for the sliding-mode controllers implemented on the BlueROV2 (one for each DOF), is to follow a pre-defined reference path given as

$$\boldsymbol{\eta}_r = [x_r, y_r, z_r, \phi_r, \theta_r, \psi_r]^T, \tag{37}$$

where we assume that the time-derivatives of  $\boldsymbol{\eta}_r$  exist and are bounded.

In the following, we design 6 sliding-mode controllers, one for each DOF, where  $\eta_i$  is the  $i$ -th component of earth-fixed vector  $\boldsymbol{\eta}$ . Then, we rewrite in component-form the system described by Equations (1) and (2) so that we have, for each component  $\eta_i$ ,

$$\dot{\eta}_i = f_i(\eta_i, \dot{\eta}_i, t) + \bar{\tau}_i, \tag{38}$$

where  $f_i(\eta_i, \dot{\eta}_i, t)$  is an unknown but bounded time-varying nonlinear function, with the explicit time dependence representing the coupling between the different DOFs. Control input  $\bar{\tau}_i$  is the  $i$ -th component of vector  $\bar{\boldsymbol{\tau}}$ , which is in turn given by  $\bar{\boldsymbol{\tau}} = [\mathbf{J}^{-T}(\boldsymbol{\eta})\mathbf{M}\mathbf{J}^{-1}(\boldsymbol{\eta})]^{-1}\mathbf{J}^{-T}(\boldsymbol{\eta})\boldsymbol{\tau}$ . Defining now the tracking error of each of the 6 DOF as  $e_i := \eta_i - \eta_{r,i}$  (with  $\eta_{r,i}$  being the  $i$ -th component of vector  $\boldsymbol{\eta}_r$  in Equation (37)), the tracking error dynamics for each DOF is given by

$$\ddot{e}_i = f_i(\eta_i, \dot{\eta}_i, t) - \dot{\eta}_{r,i} + \bar{\tau}_i. \tag{39}$$

From there, define first, for each DOF, the sliding variable

$$\sigma_i := \dot{e}_i + c_{0,i}e_i, \tag{40}$$

where  $c_{0,i}$  is a strictly positive constant.

Computing the time-derivative of  $\sigma_i$  and using expression from Equation (39), we obtain

$$\dot{\sigma}_i = \ddot{e}_i + c_{0,i}\dot{e}_i = f_i(\eta_i, \dot{\eta}_i, t) - \dot{\eta}_{r,i} + \bar{\tau}_i + c_{0,i}\dot{e}_i. \tag{41}$$

Choose then the following sliding-mode control law:

$$\bar{\tau}_i = \bar{\tau}_{i,nom} + \bar{\tau}_{i,dis}, \tag{42}$$

where the role of the so-called nominal control law  $\bar{\tau}_{i,nom}$  is to cancel the known terms in Equation (41) and is given by

$$\bar{\tau}_{i,nom} = \ddot{\eta}_{r,i} - c_{0,i}\dot{e}_i, \tag{43}$$

so that Equation (41) is transformed into

$$\dot{\sigma}_i = \sigma_{i,dis} + f_i(\eta_i, \dot{\eta}_i, t). \tag{44}$$

Finally, given the bound on unknown dynamics  $f_i$ ,

$$|f_i(\eta_i, \dot{\eta}_i, t)| \leq \bar{f}_i, \tag{45}$$

where  $\bar{f}_i$  is a strictly positive constant, the discontinuous control law  $\bar{\tau}_{i,dis}$  reads

$$\bar{\tau}_{i,dis} = -k_i \text{sgn}(\sigma_i), \tag{46}$$

where  $k_i$  is a strictly positive tuning gain, and where  $\text{sgn}(\sigma_i)$  is the signum function. Using standard Lyapunov analysis, the sliding-mode controller in Equations (40), (42), (43) and (46) can be proven to make the sliding variable  $\sigma_i$  converge to the so-called sliding surface  $\sigma_i = 0$  in finite time as long as tuning gain  $k_i$  is chosen such that  $k_i > \bar{f}_i$ . Note that having the bound  $\bar{f}_i$  is a mild but realistic assumption, as the vehicle is generally assumed to evolve in a constraint set of operations, with limited velocities.

To avoid the well-known chattering phenomenon caused by the use of the signum function in the discontinuous control law, one typically uses continuous approximations such as a sigmoid function, i.e., we have

$$\text{sgn}(\sigma_i) \approx \frac{\sigma_i}{|\sigma_i| + \epsilon_i}, \tag{47}$$

where  $\epsilon_i$  is a strictly positive tuning parameter. Note that the right-hand side term of Equation (47) approaches the signum function as parameter  $\epsilon_i$  goes to 0. Whenever the signum function in Equation (46) is replaced by the sigmoid function of Equation (47), the sliding variable does not reach 0 in finite time, but a so-called boundary layer of thickness  $\epsilon_i$  around 0, i.e., after a finite time  $t_r$ , we have  $|\sigma_i(t)| \leq \epsilon_i$  for all  $t \geq t_r$ .

### 8. Simulator

The model of the BlueROV2 is found in Section 2, and together with the parameters found in Section 4 and the tether model found in Section 6 is implemented in a MATLAB™ and Simulink™ model. The controllers from Section 7 are also implemented for all six degrees of freedom. The behavior of the BlueROV2 is visualized with a simple 3D graphic environment, which is implemented in the simulator. The simulator will be demonstrated in a case study in Section 9. A simple block diagram of the simulator is shown in Figure 11, it shows how the equations of this paper are implemented in the simulator. A video of the simulator can be found in the attached files “Case study of simulator” or at [Github](#) in Supplementary Materials. The simulator will be updated occasionally. An implicit solver is recommended due to stiffness in the system of equations, e.g., ode15s in Simulink™. A detailed guide going through each step of setting up the simulator can be found in Appendix B.



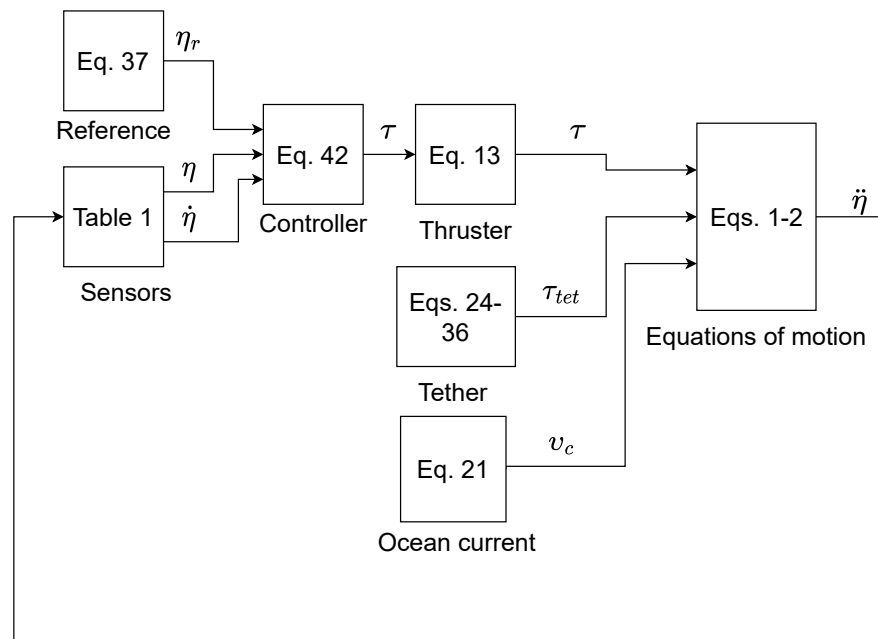


Figure 11. A simple overview of the simulator as it is implemented in Simulink™.

### 9. Case Study-Offshore Mono Pile Inspection

In this section, a case study will be carried out to demonstrate the simulator. Subsea inspections are a necessity in a variety of applications, such as in the oil and gas industry and the offshore wind industry. The case study will be based on an inspection of a monopile. The monopile is based on a realistic offshore wind turbine foundation of 6 m in diameter placed at depth of 25 m [39].

Offshore wind turbine operators have to account for a layer of marine growth when designing foundations for the offshore wind turbines, this increases the amount of steel used per installation [19]. An alternative to over-sizing the foundation is to remove the marine growth, so that it does not reach a thickness that is damaging to the structural integrity. The oil and gas operators to avoid an increase in surface area, which would decrease the fatigue life of the structure by cleaning the offshore structures periodically. However, cleaning requires annual inspections of the structure both before and after a cleaning job [40]. Before a cleaning job, inspections are needed to estimate the thickness of the marine growth to determine if it has reached a critical level, after a cleaning job inspections are carried out to verify that the marine growth has been removed and to inspect the structure for cracks to determine the structural integrity [19]. Wind turbine foundations that have been designed to withstand the marine growth layer must also be inspected for cracks, which in some cases also requires cleaning [40].

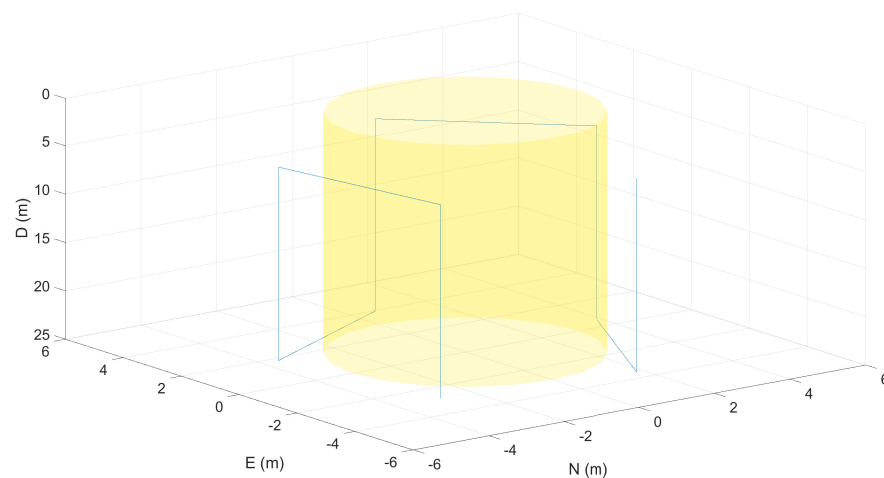
#### 9.1. Design Criteria

The criteria for a successful inspection is for the ROV to take a picture at steps of one meter along the depth of the monopile with a relative distance of one meter between the camera and the monopile. The camera will be mounted along the surge direction in the ROV's *b*-frame. The control parameters for the SMC has been designed according to Section 7.1, and can be seen in Table 6. The control parameters has been tuned initially based on simulations and then re-tuned from experimental data [3]. The noise variances and time delays in the feedback of the simulation will be based on a DVL (*u,v,w*), a laser and camera (*x,y*), an IMU (*φ,θ,p,q*), a magnetometer (*ψ,r*) and a pressure sensor (*z*), see Table 1. The thruster allocation matrix in Section 3 is implemented in the simulation, but has not been considered in the control design.

**Table 6.** Tuned control parameters for SMC.

Controller	$\alpha$	$C_0$	$\epsilon_s$	$L_b$
Surge	0.5	3	0.03	1.17
Sway	0.5	3	0.03	1.04
Heave	0.1	2	0.05	0.70
Roll	0.1	0.4	0.1	2.01
Pitch	0.1	0.4	0.1	1.83
Yaw	0.1	2	0.1	2.05

The trajectory that the ROV will follow is shown in Figure 12. It is calculated based on Table 7. The initial conditions of the ROV, water current, and tether length can be seen in Table 8. Note that the trajectory is not optimally designed, thus the ROV will not follow the shortest path.



**Figure 12.** Trajectory of the inspection campaign. The vertical lines are the inspection trajectory and the horizontal lines are the reference change.

**Table 7.** Parameters for planning the campaign.

Description	Value	Unit
Diameter of the monopile	6	m
Distance to monopile	1	m
Camera angle	2.09	rad
Height of monopile	25	m
Inspection range	5–25	m
Velocity of the ROV	0.2	ms <sup>-1</sup>

**Table 8.** Tether and ROV parameters.

Description	Value	Unit
Initial position ( $x_i, y_i, z_i$ )	[0, -5, 25]	m
Initial orientation ( $\phi_i, \theta_i, \psi_i$ )	[0, 0, 0]	rad
Current velocity	[0.21, 0.21, 0]	ms <sup>-1</sup>
Initial velocity of the ROV	[0, 0, 0]	ms <sup>-1</sup>
Initial angular velocity of the ROV	[0, 0, 0]	rads <sup>-1</sup>
Length of tether	35	m

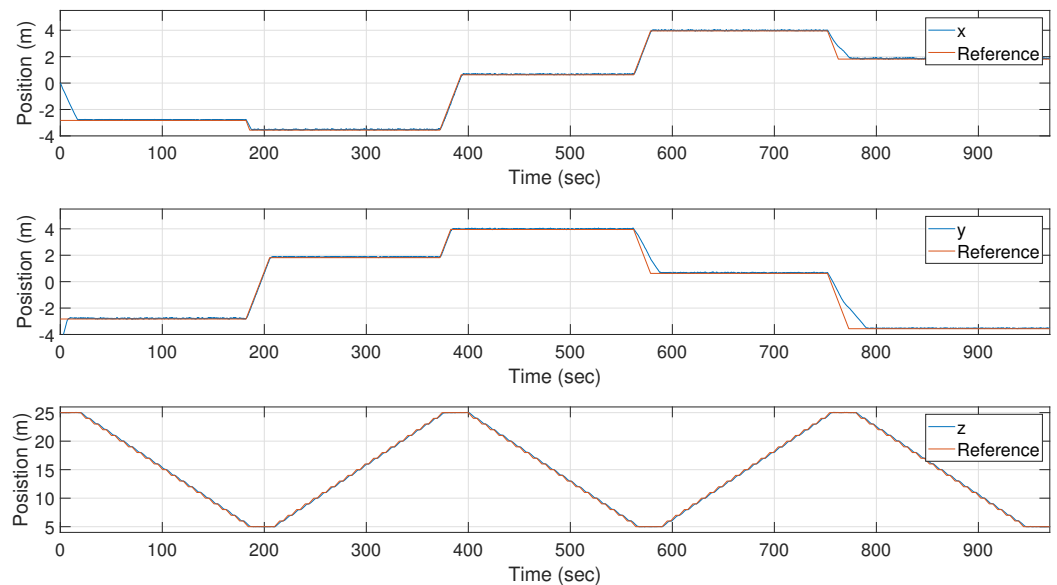
9.2. Simulation Results

In this section, a brief overview of the simulation results from the case study will be presented. The pose,  $\eta$ , of the ROV is seen in Figures 13 and 14. The controller is able follow the trajectory with a low error, even though the ROV is influenced by the tether forces shown

in Figure 15, the water current, a delay of 0.1 s in x and y from the DVL and the sensor noises. The ROV is initialized away from the trajectory to show that the controller is able to stabilize the ROV on the trajectory before it starts the mission. The starting point of the trajectory is at  $[-2.83, -2.83, 25, 0, 0, -0.75 \text{ pi}]$  with respect to the  $n$ -frame. The controller stabilizes the ROV at the reference within 20 s. Afterwards the ROV follows the trajectory, where it takes a picture at an interval of one meter in depth, from the bottom of the structure to 5 m below the surface. From Figure 13, it can be noted that the controller has a difficulty following the trajectory when the ROV is moving in the negative x and y directions, this is caused by the ocean current pushing the ROV and tether in the opposite direction. There is a similar helping effect when the ROV moves in the positive x and y direction. In Figure 15, the tether force is shown, and it should be noted that the force fluctuates due to the interaction between the actively controlled ROV and the tether. Similar fluctuations are not seen in Figure 9, where the ROV is only acting as a passive weight. In Table 9, an average of the absolute force and an average of the 5% largest forces are displayed. The impact from the tether is analyzed as statistical forces for each direction and given as a percentage of the maximum thrust force in that direction. For example, the ROV uses 38% of the thrust power to compensate for the 5% largest forces. It should be noted that this percentage is the ideal case since the ROV uses the same thrusters to surge and sway.

**Table 9.** Average and maximum tether forces during the inspection campaign. The number in parentheses is the percentage of the maximum thrust force.

Motion	Average of the Absolute Forces	Average of the 5% Largest Absolute Forces
x	21.1 N (25%)	31.9 N (38%)
y	19.3 N (23%)	28.7 N (34%)
z	12.2 N (10%)	21.2 N (18%)



**Figure 13.** Position in x,y and z.

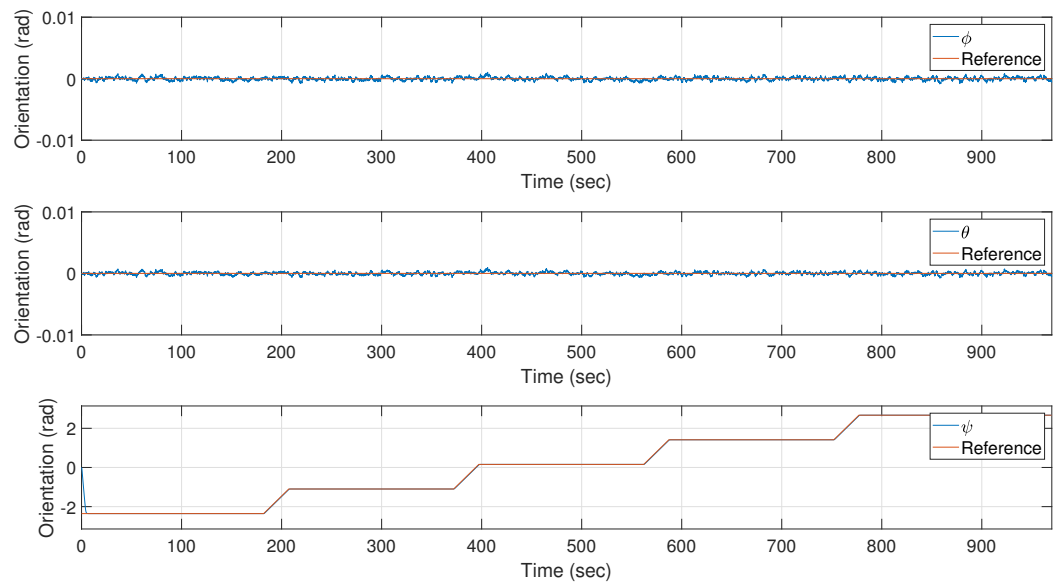


Figure 14. Orientation in  $\phi, \theta$  and  $\psi$ .

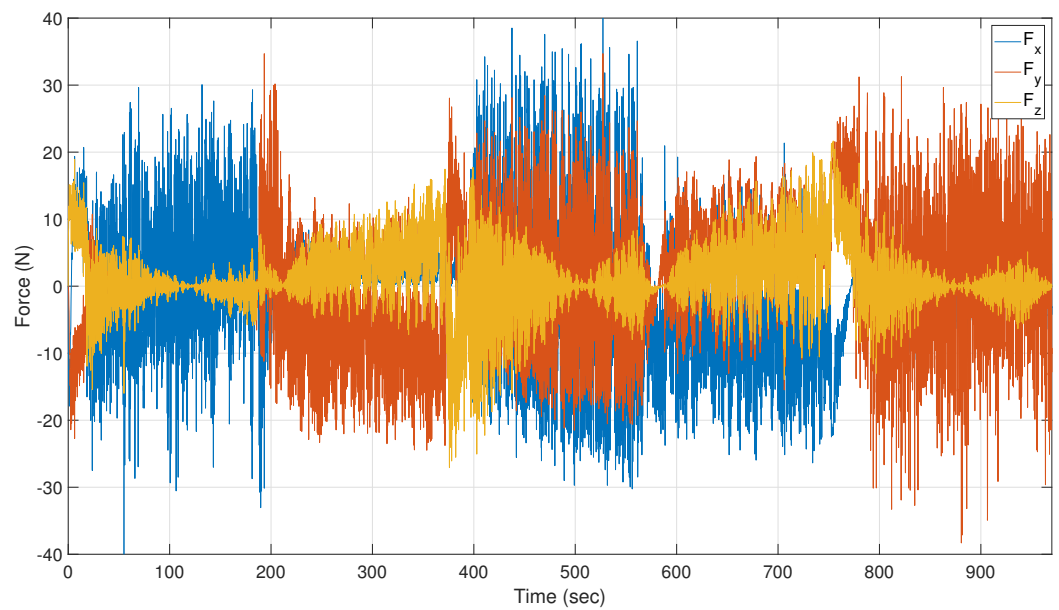


Figure 15. Tether force in x,y and z.

### 10. Concluding Remarks

This paper presented a dynamic BlueROV2 Simulator including a model of the tether connecting the ROV to the surface. The simulator is published in a MATLAB™ and Simulink™ environment. The simulator is designed to be used by the control community as a digital platform to implement control algorithms on and benchmark control algorithms. The dynamic model of the BlueROV2 is built on Fossens equations for underwater vehicles, which are widely used to model similar robotic platforms. The parameters have been found and validated using a combination of CAD software and physical experiments. The tether has been modeled using a lumped-mass approach which is an approximation of the standard finite element method. The tether model is included as it has a large impact on the dynamics of the ROV and is a necessity to do missions requiring large data transmissions, such as video feedback. A case study is presented where the ROV is deployed to inspect a monopile structure, and sliding mode control is implemented in the simulator to fulfill the task. The case study demonstrates the usefulness of the BlueROV2 simulator and the results establish a benchmark for future control schemes for positioning

and orientation trajectory tracking. The simulator has been designed to be general purpose and all parameters can be changed in the App in MATLAB™ to fit a specific vehicle. It is possible to add additional blocks in the simulator for more advanced missions such as obstacle avoidance, optimal trajectory tracing, etc. Furthermore, additional disturbances can be added by applying the force to Equation (2), this could be water waves for vehicles working close to the surface or a reaction force from a tool.

**Supplementary Materials:** The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/jmse10121898/s1>. Case study of Simulator; ROV Simulator; Tether-ROV Interaction.

**Author Contributions:** Conceptualization, M.v.B., J.L., S.P., F.F.S. and J.J.; modeling, M.v.B., J.J. and F.F.S.; experiments, M.v.B. and F.F.S.; code, M.v.B., F.F.S. and E.U.; writing—original draft preparation, M.v.B., J.L., S.P., F.F.S. and J.J.; writing—review and editing, M.v.B., J.L., S.P., F.F.S. and J.J.; visualization, M.v.B., F.F.S. and J.L.; supervision, S.P. and J.L.; funding acquisition, S.P. and J.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** The research is part of the ACOMAR project which is funded by the Energy Technology Development and Demonstration Program (EUDP), journal number 64020-1093.

**Data Availability Statement:** The data presented in this study are available on request from the authors. The proposed Simulink™ simulator can be found in the attached folder under “ROV simulator” or at [Github](#) in Supplementary Materials.

**Acknowledgments:** The authors would like to thank for the support from the Energy Technology Development and Demonstration Program (EUDP) via the “ACOMAR–Auto Compact Marine Growth Remover” project (J.No. 64020-1093). Thanks also to our project partners SubC Partner, Sihm Højtryk, Mati2ilt, Total E&P Denmark, Siemens Gamesa Renewable Energy and EIVA, and our colleagues from Aalborg University, for many valuable discussions and technical support.

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

### Appendix A

The detailed list of the values of the BlueROV2 parameters is given in Table A1.

**Table A1.** Evaluation of parameters.

Notation	Value	Unit	Origin
$g$	9.82	$\frac{m}{s^2}$	-
$\rho$	1000	$\frac{kg}{m^3}$	-
$m$	13.5	kg	Experimental
$\nabla$	0.0134	$m^3$	Experimental
$I_x$	0.26	$kg \cdot m^2$	CAD
$I_y$	0.23	$kg \cdot m^2$	CAD
$I_z$	0.37	$kg \cdot m^2$	CAD
$(x_g, y_g, z_g)$	(0, 0, 0)	m	Defined
$(x_b, y_b, z_b)$	(0, 0, -0.01)	m	CAD
$\mathbf{r}_i$ for $i = \{1, 2, 3, 4\}$	$J_3(\alpha_i) [0.156 \ 0.111 \ 0.085]^T$ for $\alpha_i = \{0, 5.05, 1.91, \pi\}$		
$\boldsymbol{\varepsilon}_i$ for $i = \{1, 2, 3, 4\}$	$J_3(\beta_i) [1/\sqrt{2} \ -1/\sqrt{2} \ 0]^T$ for $\beta_i = \{0, \frac{\pi}{2}, \frac{3\pi}{4}, \pi\}$	m	CAD
$\mathbf{r}_i$ for $i = \{5, 6, 7, 8\}$	$J_3(\gamma_i) [0.120 \ 0.218 \ 0]^T$ for $\gamma_i = \{0, 4.15, 1.01, \pi\}$		
$\boldsymbol{\varepsilon}_i$ for $i = \{5, 6, 7, 8\}$	$[0 \ 0 \ -1]^T$		
$X_u$	13.7	$\frac{Ns}{m}$	
$X_{u u }$	141.0	$\frac{Ns^2}{m^2}$	
$Y_v$	0	$\frac{Ns}{m}$	
$Y_{v v }$	217.0	$\frac{m}{Ns^2}$	
$Z_w$	33.0	$\frac{Ns}{m}$	

Table A1. Cont.

Notation	Value	Unit	Origin
$Z_{w w}$	190.0	$\frac{Ns^2}{m^2}$	Experimental
$K_p$	0	$\frac{Ns}{m}$	
$K_{p p}$	1.19	$\frac{Ns^2}{m^2}$	
$M_{\dot{q}}$	0.8	$\frac{Ns}{m^2}$	
$M_{\dot{q} q}$	0.47	$\frac{Ns^2}{m^2}$	
$N_r$	0	$\frac{Ns}{m^2}$	
$N_{r r}$	1.5	$\frac{Ns^2}{m^2}$	
$X_{\dot{u}}$	6.36	kg	[33]
$Y_{\dot{v}}$	7.12	kg	
$Z_{\dot{w}}$	18.68	kg	
$K_{\dot{p}}$	0.189	$\frac{kg\ m^2}{m^2}$	
$M_{\dot{q}}$	0.135	$\frac{kg\ m^2}{m^2}$	
$N_{\dot{r}}$	0.222	$\frac{kg\ m^2}{m^2}$	
$m_{tet}$	0.043	$\frac{kg}{m}$	Experimental
$d_{tet}$	0.0075	m	Experimental
$Cn_{tet}$	1.2	-	[41]
$Ct_{tet}$	0.01	-	[41]
$C_{tet}$	100	$\frac{Ns}{m}$	[41]
$E_{tet}$	$6.437 \times 10^{10}$	$\frac{N}{m^2}$	[41]

### Appendix B. Documentation

#### Appendix B.1. Installing the Simulation Software

The app can be found at the website [Github](#) in Supplementary Materials or in the attachments. Install the app by running the “ROV Simulator.mlappinstall” installation file. A window will appear where you have to accept the installation of the ROV Simulator app. Agree to this and the app is now installed.

#### Appendix B.2. Launching the App

To launch the app for the first time, open MATLAB and chose the APPS pane in the top of the screen, as seen in Figure A1. Then, locate and click the icon named “ROV Simulator”.

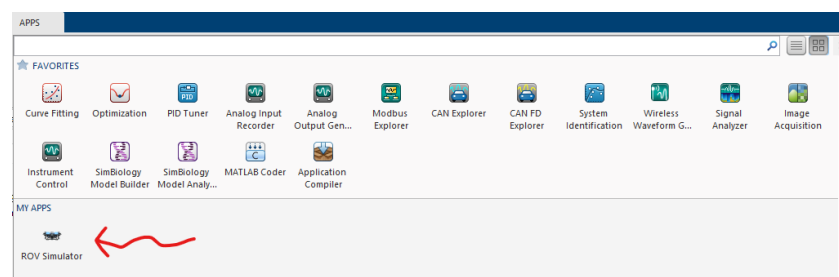


Figure A1. App launch from within MATLAB.

When the app is launched for the first time, certain initial parameters are designated. These are set to follow the structure of this report. To add or change any of these parameters, the following tabs within the graphical user interface of the app can be used.

#### Appendix B.3. Simulation Options

The first section is the simulation options pane. To quickly start the simulation, the “Open Simulation” button can be pressed in Figure A2.

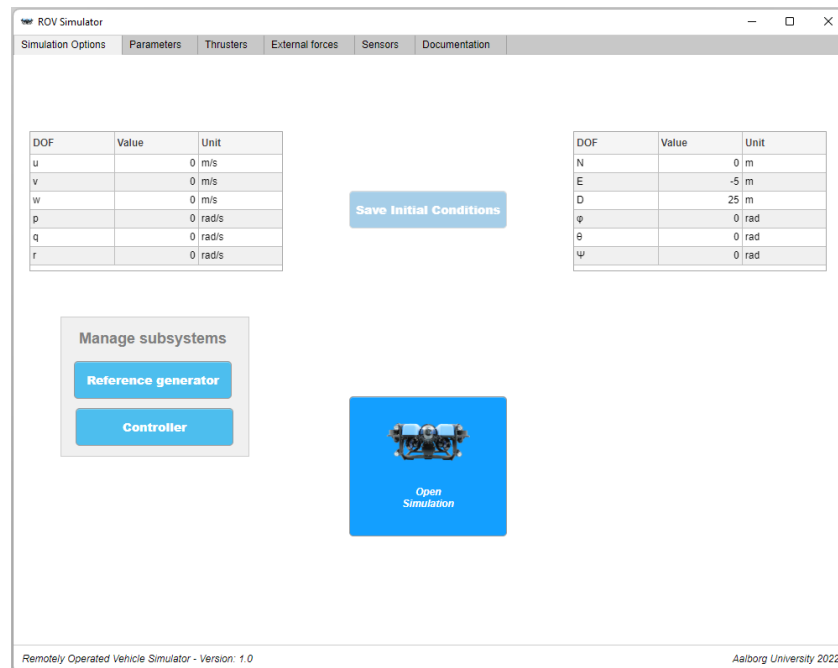


Figure A2. Overview of the simulation options pane within the ROV simulator app.

Before the simulation is launched, the initial settings of the robot can be changed. It is possible to change both initial positions and velocities. Using the checkbox, it is possible to enable or disable visualization of the ROV and its movements. By disabling visualization within the simulation, the simulation speed can be increased. The controller or the references subsystem can be opened using the buttons in the lower left part.

#### Appendix B.4. Parameters

Within this section, it is possible to add or change the robot and environment parameters used for the simulation. An overview of this section is seen in Figure A3.

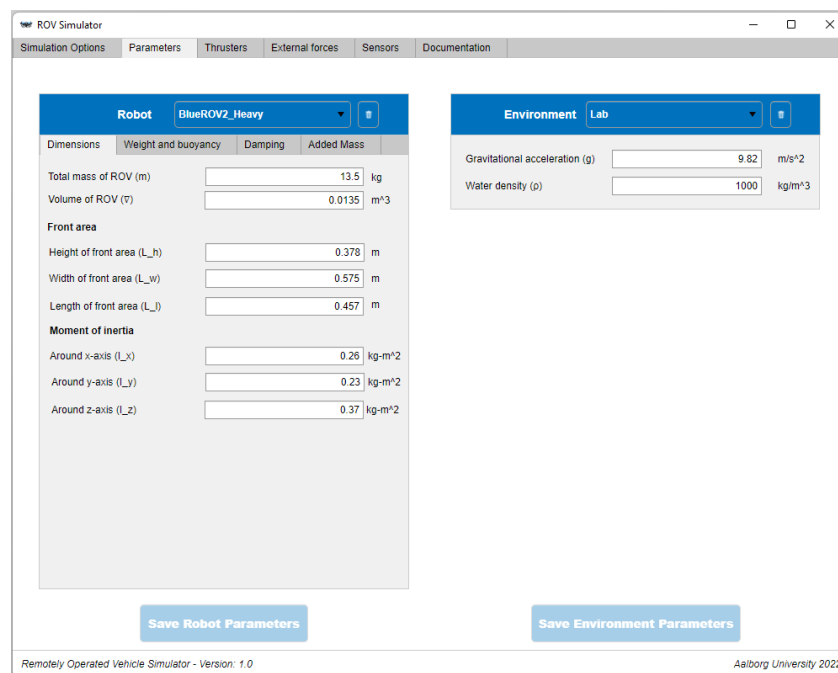


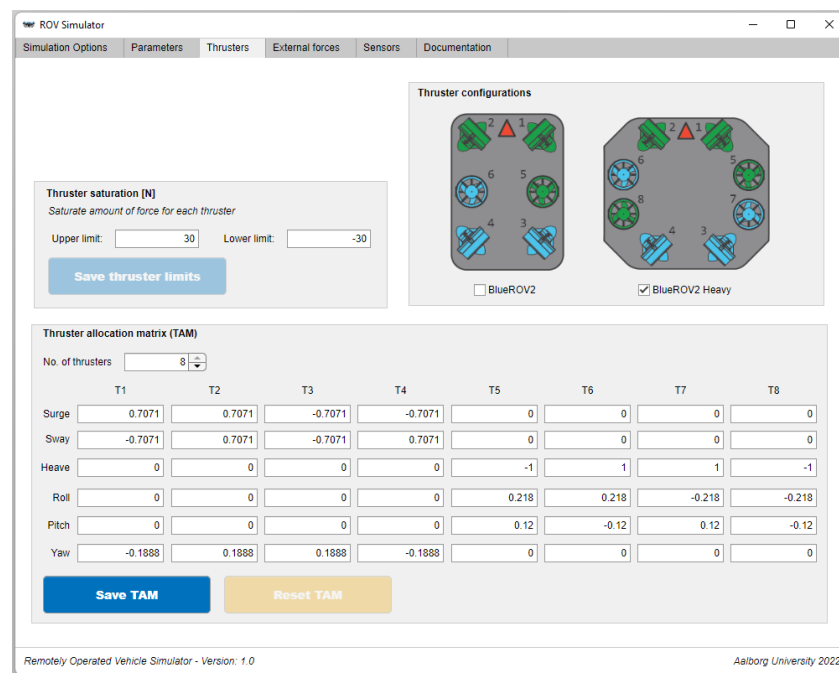
Figure A3. Overview of the robot and environment parameters pane within the ROV simulator app.

The individual parameters of the robots as well as environments can be changed by editing the corresponding parameter-field. User created robots and environments can be deleted by using the trash-can located next to the drop-down list. To choose a robot or environment, click the desired one on either of the drop-down lists. To add a new robot or environment, choose “Add New” option from within the drop-down list. If the projected area is updated within the robot parameters, the added mass can be recalculated using the “Update Added Mass” button using the approach described in Section 4.3. Any changes can be saved using the save buttons.

All parameters chosen within this section are saved locally and transferred directly to the simulation file. Parameters are also updated within each run if changed in the GUI.

*Appendix B.5. Thrusters*

This section is used to define the saturation force supplied by the individual thrusters as well as determining the location, orientation and direction of the thrusters located on the robot. This section refers to Equation (13). The upper and lower limits used for saturating the thrusters in simulations can be changed in their respective text fields. As seen in Figure A4, it is possible to switch between predefined thruster allocations for the BlueROV2 and BlueROV2 heave configuration. Using the “No. of thrusters” spinner, the number of thrusters can be changed between 2 to 8. The location, orientation and direction of each thruster with respect to a movement in any of the six degrees of freedoms can be edited by changing the values within their respective text-fields. Using the “Reset TAM” button, the thruster allocation matrix can be reset to the initial BlueROV2 heavy configuration. Remember to save any changes using the “Save TAM” button.



**Figure A4.** Overview of the thruster parameters pane within the ROV simulator app.

*Appendix B.6. External Forces*

In this section, it is possible to enable or disable external forces on the ROV.

As seen in Figure A5 it is possible to enable or disable the effects from ocean currents Section 5 and the tether from Section 6. An extra option is added in order to add custom external forces in any of the six degrees of freedom, this is edited under “Manual forces”. It is possible to edit the different parameters of any of the three forces within the text-fields grouped together with the forces.



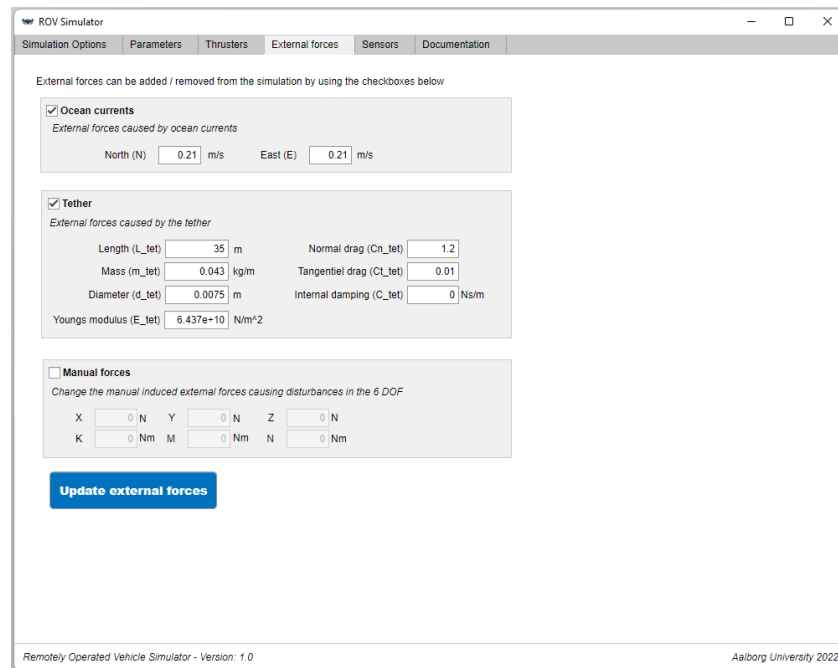


Figure A5. Overview of the external forces pane within the ROV simulator app.

### Appendix B.7. Sensor Disturbances

Multiple sensors can be used to determine the position and velocity of the robot. These sensors can be associated with disturbances such as noise, delay, and drift. This section enables the user to add the effect of such disturbances to any of the 12 states. The sensors included in the software correspond to those found in Table 1.

As seen in Figure A6, all position and velocity states are located in the top left of the section. Using the individual drop-down menus of each state, sensor disturbances can be selected. If “Ideal” is selected, no sensor disturbance will affect the selected state. In the bottom of the section, a table is located containing seven specific sensors as well as one custom sensor. The table displays the following parameters for each of the sensors. It contains state-measurements, sampling time, delay, variance and drift. It is not possible to change the state-measurements of any of the sensors; however, the rest of the settings can be edited by using a double click on any of the fields. The sampling time should be set in Hz, and can not be equal to nor less than zero. The delay of sensor measurements can be set in seconds, the variance is either set in meters, radians or radians per second, with regards to which states the sensor measures. The drift of the sensor is set as an increase in meters or radians per second. Only the reference generator as well as the controller subsystems should be edited, settings of the remaining subsystems can be edited from the app. The scopes in the bottom of the block diagram allow for viewing positions and velocities for both *b*-frame and *n*-frame. As a further remark, the simulations are based upon two simulation files, with and without the visualization option. All subsystems are added as reference subsystems, meaning that it does not matter in which of the simulation files these are edited, since they are loaded into both. The app and simulation files are tested for use with MATLAB and Simulink 2021b and 2022a.

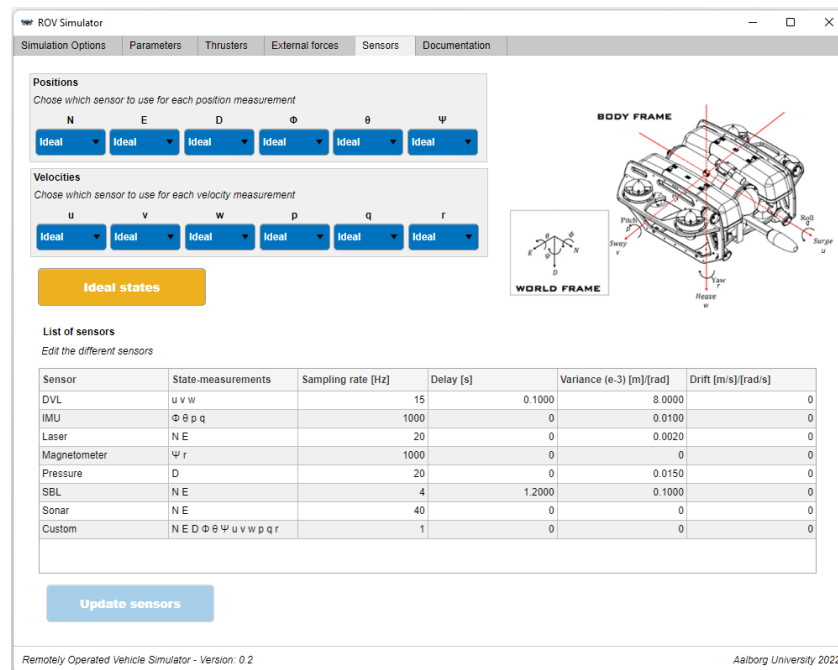


Figure A6. Overview of the Sensor disturbances pane within the ROV simulator app.

Appendix B.8. Simulation Structure

Figure A7 shows the Simulink block diagram of the simulation model. If the visualization option is disabled, their corresponding blocks will be removed.

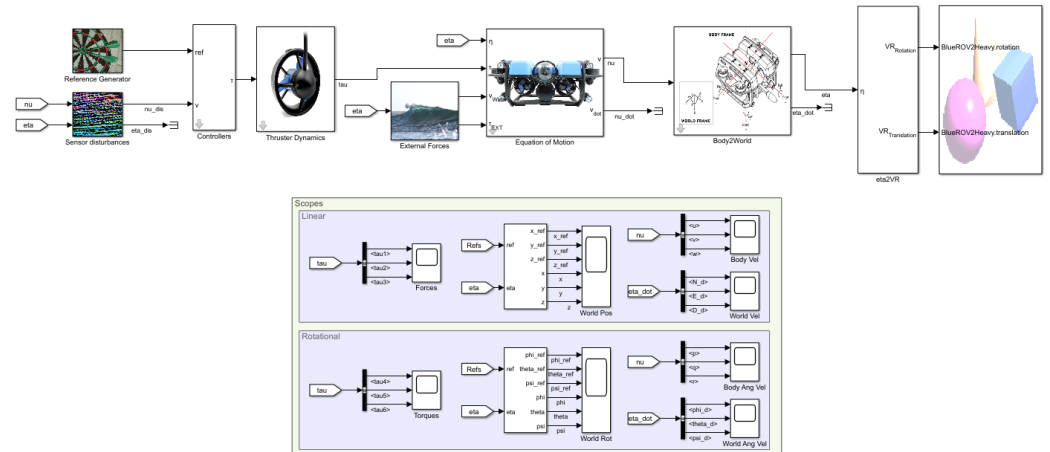


Figure A7. Overview of the simulation structure of the simulation file as seen in Simulink.

References

1. Healey, A.J.; Lienard, D. Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE J. Ocean. Eng.* **1993**, *18*, 327–339. [CrossRef]
2. Willcox, S.; Goldberg, D.; Vaganay, J.; Curcio, J. Multi-vehicle cooperative navigation and autonomy with the bluefin CADRE system. In Proceedings of the IFAC (International Federation of Automatic Control) Conference, Heidelberg, Germany, 15 September 2006; pp. 20–22.
3. von Benzon, M.; Sorensen, F.; Liniger, J.; Pedersen, S.; Klemmensen, S.; Schmidt, K. Integral Sliding Mode control for a marine growth removing ROV with water jet disturbance. In Proceedings of the 2021 European Control Conference (ECC), Delft, The Netherlands, 29 June–2 July 2021. [CrossRef]
4. Pedersen, S.; Liniger, J.; Sørensen, F.F.; Schmidt, K.; Benzon, M.v.; Klemmensen, S.S. Stabilization of a ROV in three-dimensional space using an underwater acoustic positioning system. *IFAC-PapersOnLine* **2019**, *52*, 117–122. [CrossRef]
5. Kuhn, V.N.; Drews Jr, P.L.J.; Gomes, S.C.P.; Cunha, M.A.B.; Botelho, S.S.d.C. Automatic control of a ROV for inspection of underwater structures using a low-cost sensing. *J. Braz. Soc. Mech. Sci. Eng.* **2015**, *37*, 361–374. [CrossRef]

6. Eslami, M.; Chin, C.S.; Nobakhti, A. Robust Modeling, Sliding-Mode Controller, and Simulation of an Underactuated ROV Under Parametric Uncertainties and Disturbances. *J. Mar. Sci. Appl.* **2019**, *18*, 213–227. [[CrossRef](#)]
7. Keviczky, T.; Borrelli, F.; Fregene, K.; Godbole, D.; Balas, G.J. Decentralized Receding Horizon Control and Coordination of Autonomous Vehicle Formations. *IEEE Trans. Control Syst. Technol.* **2008**, *16*, 19–33. [[CrossRef](#)]
8. Selvakumar, J.; Selvakumar, J.; Asokan, T.; Asokan, T. Station keeping control of underwater robots using disturbance force measurements. *J. Mar. Sci. Technol.* **2016**, *21*, 70–85. [[CrossRef](#)]
9. Willners, J.S.; Carlucho, I.; Luczyński, T.; Katagiri, S.; Lemoine, C.; Roe, J.; Stephens, D.W.; Xu, S.J.; Carreno, Y.; Pairet, È.; et al. From market-ready ROVs to low-cost AUVs. In Proceedings of the OCEANS 2021: San Diego–Porto, San Diego, CA, USA, 20–23 September 2021; pp. 1–7.
10. Robotics, B. BlueROV2: The World’s Most Affordable High-Performance ROV. In *BlueROV2 Datasheet*; Blue Robotics: Torrance, CA, USA, June 2016.
11. Wilby, A.; Lo, E. Low-Cost, Open-Source Hovering Autonomous Underwater Vehicle (HAUV) for Marine Robotics Research based on the BlueROV2. In Proceedings of the 2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV), St. Johns, NL, Canada, 30 September–2 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
12. Chua, A.; MacNeill, A.; Wallace, D. Democratizing ocean technology: Low-cost innovations in underwater robotics. In Proceedings of the EGU General Assembly Conference Abstracts, Online, 4–8 May 2020; p. 10190.
13. Manzanilla, A.; Reyes, S.; Garcia, M.; Mercado, D.; Lozano, R. Autonomous navigation for unmanned underwater vehicles: Real-time experiments using computer vision. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1351–1356. [[CrossRef](#)]
14. Manhães, M.M.M.; Scherer, S.A.; Voss, M.; Douat, L.R.; Rauschenbach, T. UUV Simulator: A Gazebo-based package for underwater intervention and multi-robot simulation. In Proceedings of the OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA, 19–23 September 2016; IEEE: Piscataway, NJ, USA, 2016. [[CrossRef](#)]
15. de Cerqueira Gava, P.D.; Nascimento Júnior, C.L.; Belchior de França Silva, J.R.; Adabo, G.J. Simu2VITA: A General Purpose Underwater Vehicle Simulator. *Sensors* **2022**, *22*, 3255. [[CrossRef](#)] [[PubMed](#)]
16. Potokar, E.; Ashford, S.; Kaess, M.; Mangelson, J. HoloOcean: An Underwater Robotics Simulator. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022.
17. Bogue, R. Robots in the offshore oil and gas industries: A review of recent developments. *Ind. Robot* **2020**, *47*, 1–6. [[CrossRef](#)]
18. Rofallski, R.; Tholen, C.; Helmholtz, P.; Parnum, I.; Luhmann, T. Measuring artificial reefs using a multi-camera system for unmanned underwater vehicles. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.-ISPRS Arch.* **2020**, *43*, 999–1008. [[CrossRef](#)]
19. Pedersen, S.; Liniger, J.; Sørensen, F.F.; von Benzon, M. On Marine Growth Removal on Offshore Structures. In Proceedings of the OCEANS 2022-Chennai, Chennai, India, 21–24 February 2022; pp. 1–6. [[CrossRef](#)]
20. Liniger, J.; Jensen, A.L.; Pedersen, S.; Sørensen, H.; Mai, C. On the Autonomous Inspection and Classification of Marine Growth on Subsea Structures. In Proceedings of the 2022 OCEANS Conference & Exposition, Oceans 2022 Chennai, Chennai, India, 21–24 February 2022; IEEE Press: Piscataway, NJ, USA, 2022.
21. Boogaard, F.; de Lima, R.; de Graaf, R. Innovative water quality and ecology monitoring using underwater unmanned vehicles: Field applications, challenges and feedback from water managers. *Water* **2020**, *12*, 1196.
22. Wang, Y.; Thanyamanta, W.; Bulger, C.; Bose, N. Experimental study to make gas bubbles as proxies for oil droplets to test AUV detection of oil plumes. *Appl. Ocean Res.* **2022**, *121*, 103080. [[CrossRef](#)]
23. Betancourt, J.; Coral, W.; Colorado, J. An integrated ROV solution for underwater net-cage inspection in fish farms using computer vision. *SN Appl. Sci.* **2020**, *2*, 1946. [[CrossRef](#)]
24. Fossen, T.I. *Marine Control Systems—Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles*; Marine Cybernetics: Trondheim, Norway, 2002; ISBN 82-92356-00-2.
25. BlueRobotics. *BlueROV2 Assembly*; BlueRobotics: Torrance, CA, USA, 2021.
26. Antonelli, G. *Underwater Robots*, 4th ed.; Springer International Publishing AG: Cham, Switzerland, 2018; Volume 123. [[CrossRef](#)]
27. Veritas, D.N. *Environmental Conditions and Environmental Loads*; Det Norske Veritas: Oslo, Norway, 2021.
28. Fossen, T.I. Nonlinear Modelling and Control of Underwater Vehicles. Ph.D. Thesis, The Norwegian Institute of Technology, Trondheim, Norway, 1991.
29. SNAME: The Society of Naval Architects and Marine Engineers. *Nomenclature for Treating the Motion of a Submerged Body Through a Fluid*; Technical and Research Bulletin No. 1–5; SNAME: Alexandria, VA, USA, 1950.
30. Gertler, M.; Hagen, G.R. *Standard Equations of Motion for Submarine Simulation*; Technical Report; David w Taylor Naval Ship Research and Development Center: Bethesda, MD, USA, 1967.
31. BlueRobotics. *BlueROV2 Heavy Configuration Retrofit Kit*; BlueRobotics: Torrance, CA, USA, 2021.
32. MathWorks Student Competitions Team. *MATLAB and Simulink Robotics Arena: From Data to Model*; MATLAB Central File Exchange; 2022. Retrieved 24 November 2022. Available online: <https://www.mathworks.com/matlabcentral/fileexchange/65919-matlab-and-simulink-robotics-arena-from-data-to-model> (accessed on 1 October 2022).
33. Eidsvik, O.A. Identification of Hydrodynamic Parameters for ROV’s. Master’s Thesis, The Norwegian Institute of Technology, Trondheim, Norway, 2015.
34. Det Norske Veritas (DNV). *Modelling and Analysis of Marine Operations*; DNV: Oslo, Norway, 2011.
35. McLain, T.W.; Rock, S.M. Experimental Measurement of ROV Tether Tension. In Proceedings of the Intervention/ROV ’92, San Diego, CA, USA, 10–12 June 1992.

36. Hall, M.; Goupee, A. Validation of a lumped-mass mooring line model with DeepCwind semisubmersible model test data. *Ocean Eng.* **2015**, *104*, 590–603. [[CrossRef](#)]
37. Driscoll, F.; Lueck, R.; Nahon, M. Development and validation of a lumped-mass dynamics model of a deep-sea ROV system. *Appl. Ocean Res.* **2000**, *22*, 169–182. [[CrossRef](#)]
38. Slotine, J.J.E.; Li, W. *Applied Nonlinear Control*; Prentice Hall: Englewood Cliffs, NJ, USA, 1991; pp. 276–310.
39. HOFOR Wind A/S. *Offshore and Onshore Technical Project Description nr. 13-97289-2*; Technical Report; Energinet.dk: Fredericia, Denmark, 2021.
40. Anderson, M. Nondestructive testing of offshore structures. *NDT Int.* **1987**, *20*, 17–21. [[CrossRef](#)]
41. Souza, E.C.D.; Maruyama, N. Intelligent UUVs: Some issues on ROV dynamic positioning. *IEEE Trans. Aerosp. Electron. Syst.* **2007**, *43*, 214–226. [[CrossRef](#)]