Ref. Ares(2025)11631738 - 31/12/2025

Scalable full-cycle marine litter remediation in the Mediterranean: Robotic and participatory solutions

SeaClear2.0

https://www.seaclear2.eu

D5.2
Report on Smart Grapple Software Framework
WP5 — Control Overhaul

Grant Agreement no. 101093822

Lead beneficiary: TUM

Date: 30/12/2025
Type: R
Dissemination level: PU

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS
RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

Document information

Grant agreement no.	101093822
Acronym:	SeaClear2.0
Full title:	Scalable full-cycle marine litter remediation in the Mediterranean: Robotic and participatory solutions
Start date of the project	01/01/2023
Duration of the project	48 months
Deliverable	D5.2: Report on Smart Grapple Software Framework
Work package	WP5: Control Overhaul
Deliverable leader	TUM
Delivery date	Contractual: 31/12/2025 Actual: 30/12/2025
Status	Draft ☐ Final ✓
Type*	R ✓ DEM ☐ OTHER ☐ DMP ☐
Dissemination level†	PU ✓ C-UE/EU-C ☐ SEN ☐
Author(s)	Stefan Sosnowski (TUM) Nicolas Hoischen (TUM) Tzu-Yuan Huang (TUM) Anastasios Natsakis (UTC) Hamish Grant (TUM) Lucas De Oliveira e Silva (TUM) Zara Zhotabayeva (TUM)
Responsible author	Stefan Sosnowski, email: sosnowski@tum.de TUM
Deliverable description	Report on the perception and control approach that will be developed, based on the smart grapple design in T3.2, for localizing and precisely manipulating debris items under environmental disturbances. Including a holistic optimal control approach of the integrated USV - smart grapple system. This deliverable reflects work done in task T5.2 "Smart grapple sensing and control".

[1] R = Document, report, DEM = demonstrator, OTHER: Software, technical diagram, etc. DMP = Data Management Plan

[2] PU = Public, C-UE/EU-C = EU Confidential under Decision 2015/444, SEN = Sensitive

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

Document history

Name	Date	Version	Description
Nicolas Hoischen	03/11/2025	V0.1	initial draft setup
Stefan Sosnowski	28/11/2025	V0.2	final draft
Stefan Sosnowski	30/12/2025	V1.0	final version

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

Disclaimer of warranties

This document has been prepared by SeaClear2.0 project partners as an account of work carried out within the framework of Grant Agreement no. 101093822. Neither the Project Coordinator, nor any signatory party of the SeaClear2.0 Project Consortium Agreement, nor any person acting on behalf of any of them:

- makes any warranty or representation whatsoever, express or implied, with respect to the use of any information, apparatus, method, process, or similar item disclosed in this document, including merchantability and fitness for a particular purpose, that such use does not infringe on or interfere with privately owned rights, including any party's intellectual property; or
- makes any warranty or representation whatsoever, express or implied, that this document is suitable to any particular user's circumstance; or
- assumes responsibility for any damages or other liability whatsoever (including any consequential damages, even if the Project Coordinator or any representative of a signatory party of the Project Consortium Agreement, has been advised of the possibility of such damages) resulting from your selection or use of this document or any information, apparatus, method, process, or similar item disclosed in this document.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

Table of Contents

Definitions . . . 7

Abbreviations . . . 7

Executive Summary . . . 10

1 Introduction . . . 11
- 1.1 SeaClear2.0 at a glance . . . 11
- 1.2 A review of the litter collection capabilities in SeaClear1.0 . . . 12
- 1.3 Deliverable objective and Document Structure . . . 12

2 Smart Grapple control stack . . . 14
- 2.1 Smart Grapple State Estimation . . . 14
- 2.2 Smart Grapple Control Allocation . . . 15
  - 2.2.1 Thruster Positions on the Hardware . . . 15
  - 2.2.2 Derivation of the Control Allocation Matrix . . . 16
  - 2.2.3 Thruster Allocation Under Power Constraints . . . 17
- 2.3 Joint control . . . 21
  - 2.3.1 Graphic user interface . . . 21
  - 2.3.2 Calibration . . . 22
- 2.4 State machine of the operation cycle . . . 23
  - 2.4.1 Deploying . . . 23
  - 2.4.2 Planning . . . 24
  - 2.4.3 Tracking . . . 24
  - 2.4.4 Visual servoing . . . 24
  - 2.4.5 Grasping . . . 25
  - 2.4.6 Dropping (Recovering) . . . 25
  - 2.4.7 Error . . . 25

3 Holistic Optimal Control Approach . . . 27
- 3.1 Modeling of the Integrated Uncrewed Surface Vehicle (USV)-Underwater Smart Grapple System (USG) . . . 27
  - 3.1.1 Coordinate Frame and State Definition . . . 27
  - 3.1.2 Modeling Dynamics . . . 28
  - 3.1.3 External Forces . . . 30
  - 3.1.4 Model Summary . . . 31
- 3.2 Technical Approach . . . 31
  - 3.2.1 Stochastic Optimal Control Problem . . . 31
  - 3.2.2 Feedback Augmentation . . . 33
  - 3.2.3 Implementation . . . 34
  - 3.2.4 Evaluations: Latest Simulation Results . . . 35

4 Perception System for Litter Object Localization . . . 37
- 4.1 Underwater Object Detection . . . 37
- 4.2 Truncated Signed Distance Function based 3D Reconstruction . . . 38

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

5 Smart Grapple Grasping Approach . . . 41
5.1 Torque Monitor Grasping Algorithm . . . 41
5.2 Grapple Kinematics . . . 42
5.2.1 Analytic Solution . . . 42
5.2.2 Iteration Solution . . . 45
5.3 Optimal Grasping Manipulability Under Joint Position and Joint Torque Constraints 45
5.3.1 Grasping Scenarios . . . 48
5.3.2 Static Force Analysis . . . 49
5.3.3 Grasping Posture in Different Scenarios . . . 50
5.4 Gripping points optimization . . . 52

6 Digital Twin of the Smart Grapple . . . 54

7 Experiments . . . 56
7.1 Laboratory tests . . . 56
7.2 Field Results in Marseille 2025 . . . 56

8 Conclusions . . . 61

References . . . 62

A Parameter lists . . . 63
A.1 Thrust allocation . . . 63
A.2 Kinematic parameters . . . 63
A.3 Holistic control approach simulation parameters . . . 64

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

Definitions

- **Beneficiary:** A legal entity that is signatory of the EC Grant Agreement no. 101093822.
- **Consortium:** The SeaClear2.0 Consortium, comprising the below-mentioned list of beneficiaries.
- **Consortium Agreement:** Agreement concluded amongst SeaClear2.0 Beneficiaries for the implementation of the Grant Agreement.
- **Grant Agreement:** The agreement signed between the beneficiaries and the EC for the undertaking of the SeaClear2.0 project (Grant Agreement no. 101093822).

Beneficiaries of the SeaClear2.0 Consortium are referred to herein according to the following abbreviations:

- **TU Delft:** TECHNISCHE UNIVERSITEIT DELFT
- **DUNEA:** REGIONALNA AGENCIJA DUNEA
- **Fraunhofer:** FRAUNHOFER GESELLSCHAFT ZUR FORDERUNG DER ANGEWANDTEN FORSCHUNG EV
- **HPA:** HAMBURG PORT AUTHORITY
- **ISOTECH**: ISOTECH LTD
- **MDanchor**: M. DANCHOR LTD
- **Subsea Tech:** SUBSEA TECH SAS
- **TECNOSUB**: TÉCNICAS Y OBRAS SUBACUÁTICAS, SLU
- **TUM**: TECHNISCHE UNIVERSITAET MUENCHEN
- **UNIDU:** SVEUCILISTE U DUBROVNIKU
- **UTC**: UNIVERSITATEA TEHNICA CLUJ-NAPOCA
- **VEO**: VEOLIA PROPRETE
- **VLPF**: VENICE LAGOON PLASTIC FREE

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

Abbreviations

CoM Center of Mass. 16, 17

CVaR Conditional Value at Risk. 27, 32

DVL Doppler Velocity Logger. 12, 14, 24, 54

EKF Extended Kalman Filter. 14

EtherCAT Ethernet for Control Automation Technology. 21

FABRIK Forward And Backward Reaching Inverse Kinematics. 42, 45–48

GNSS Global Navigation Satellite System. 15

GUI Graphical User Interface. 20–25

IMU Inertial Measurement Unit. 14

LARS Launch and Recovery System. 12

MARUS MArine Robotics Unity Simulator. 54

OCP Optimal Control Problem. 32

octomap octree map. 24

OMPL Open Motion Planning Library. 24

PID Proportional-Integral-Derivative. 21, 24

QP Quadratic Program. 18–20

RA-SAA Risk-Aware Sample Average Approximation. 27, 31

ROS2 Robot Operating System 2. 21, 25

RRT* Rapidly Exploring Random Tree Star. 24

SDE Stochastic Differential Equation. 31, 34

SDF Signed Distance Fields. 38

SIL Software in the Loop. 54

SOEM Simple Open EtherCAT Master. 21

STO Safe Torque Off. 22

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

TSDF Truncated Signed Distance Function. 38, 40

USBL Ultra-Short Baseline. 15

USG Underwater Smart Grapple System. 5, 14–16, 19–21, 27, 29, 30, 32, 33, 54, 59, 64

USV Uncrewed Surface Vehicle. 5, 11, 23, 25, 27–30, 32, 36, 58, 64

UUV Uncrewed Underwater Vehicle. 11, 12, 24, 36

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

Executive summary

This deliverable (D5.2) presents the software framework developed for the Smart Grapple system within the SeaClear2.0 project, an EU-funded initiative for autonomous marine litter remediation in the Mediterranean. The Smart Grapple represents a significant advancement over SeaClear, enabling collection of debris up to 250 kg (compared to the previous 3 kg limit) including large items such as tires, bicycles, car seats, and fences.

The report details four major technical contributions. First, a **holistic optimal control approach** is presented for the integrated Unmanned Surface Vehicle (USV) and Underwater Smart Grapple (USG) system. This includes mathematical modeling using Euler-Lagrange dynamics, a stochastic optimal control formulation using Risk-Aware Sample Average Approximation (RA-SAA), and feedback augmentation for robust trajectory tracking under environmental disturbances and parameter uncertainties.

Second, the **control allocation system** for the eight-thruster configuration is described, including derivation of the allocation matrix, a quadratic programming formulation for thrust distribution, and a percentage phase-back strategy to handle the 5 kW power constraint while maintaining real-time feasibility.

Third, comprehensive **grasping algorithms** are documented, encompassing torque-monitor-based adaptive grasping with a three-state machine architecture, forward and inverse kinematics solutions (including the FABRIK iterative method), and optimal grasping posture determination under joint position and torque constraints across three operational scenarios.

Fourth, the **perception system** for autonomous litter localization is presented, integrating YOLOv8-based underwater object detection with Truncated Signed Distance Function (TSDF) 3D reconstruction for metric geometry estimation.

The system was validated through laboratory experiments demonstrating successful grasping of loads up to 251 kg, and field trials in Marseille (September 2025) where the system successfully retrieved various objects including a car seat (45 kg dry weight), tire, and attempted recovery of a shipwreck fragment. A digital twin using the MARUS simulator enables software-in-the-loop testing between field deployments.

The deliverable contributes to project objective O8 and supports KPIs 8.1-8.3 targeting 90% collection success rates for seafloor and surface objects across minimum three litter types.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

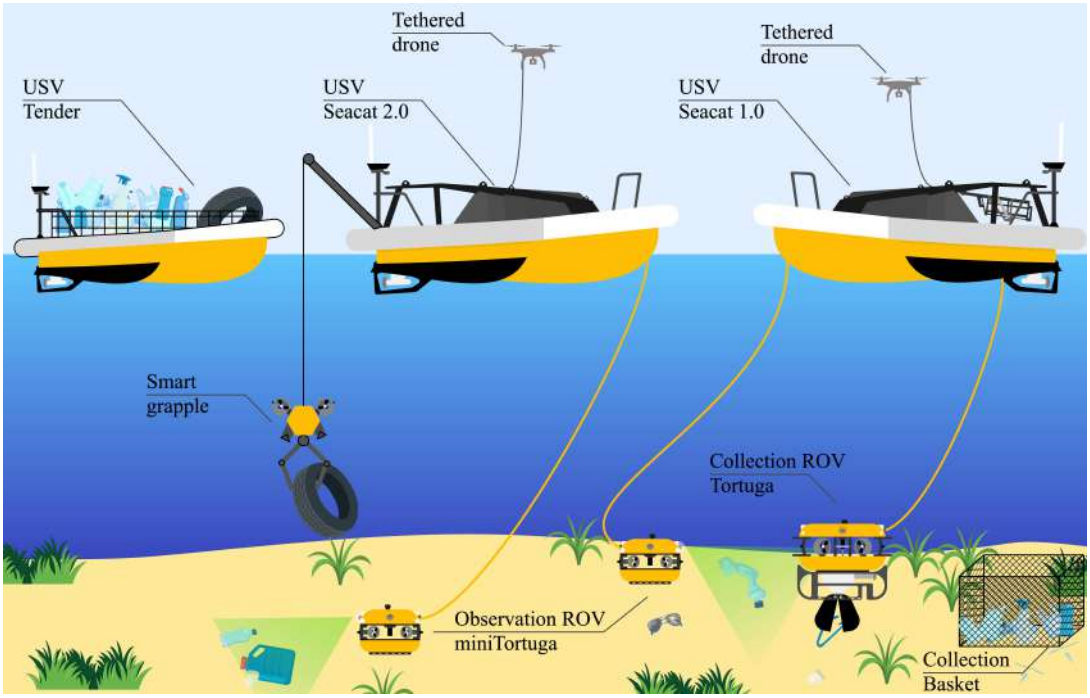


Figure 1: Concept of SeaClear2.0 robotic system (left/middle) and SeaClear1.0 (right)

1 Introduction

1.1 SeaClear2.0 at a glance

SeaClear2.0 is the next step to the H2020-funded SeaClear (https://seaclear-project.eu), where several of the SeaClear2.0 partners have been developing the first autonomous robotic system for seafloor litter collection. An USV called SeaCAT acts as a hub for an unmanned aerial vehicle (UAV) that searches for litter from the air and contributes to situational awareness, an observation Uncrewed Underwater Vehicle (UUV) called Mini-Tortuga that searches for litter underwater, and a collection UUV called Tortuga that collects the litter with a custom gripper plus-suction device and deposits it in a basket lowered from the USV. The very nature of SeaClear (labelled 1.0 in this document) as the first project to design such a solution means it must focus almost exclusively on technological developments in robotics hardware and algorithms, with a single system.

Moreover, the system is limited to only seafloor litter of up to tens of centimeters in size and at depths of up to tens of meters; it cannot collect surface litter or larger objects like fishing equipment, bikes, e-scooters, tires, shipping equipment, etc, which SeaClear1.0 identified as a significant subset of the litter in industrial port or mariculture settings. The aim of SeaClear2.0 is to upscale and upgrade the system initially made for SeaClear1.0. This means collecting bigger and heavier litter with a modified SeaCAT2 and a smart grapple, storing and shipping it to shore with a dedicated USV Tender, as well as collecting surface litter with a USV and a team of highly manoeuvrable surface robots. In the end, the SeaClear1.0 and SeaClear2.0 systems will be working together, exploiting their shared and complementary litter detection and collection capabilities.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union



Figure 2: SeaClear1.0 collection UUV with small-scale litter.

1.2 A review of the litter collection capabilities in SeaClear1.0

In the SeaClear1.0 project the collection was limited to small-scale and lightweight litter as a proof-of-concept, with a weight limit of up to 3 kg and a maximum size of about 30 cm. This limitation stemmed from the choice of the Tortuga robot as the robotic platform, given its maximum payload of 10 kg and overall dimensions, as well as the compatibility with the Tortuga Launch and Recovery System (LARS) system. For the SeaClear1.0 project, this was sufficient to research the mapping and collection strategy, and focus on the predominant litter sizes at the test sites. The litter occurrence analysis in *Deliverable 2.1 "Marine litter occurrence domains report"*, however, shows that there is an additional need for being able to handle both larger and heavier objects that can be found in the Mediterranean Sea and elsewhere. Furthermore, the monoocular downward-facing camera allowed for object detection and tracking, but could neither determine the distance to the object nor its 3D shape. The distance to the object was estimated from altitude measurements with the Doppler Velocity Logger (DVL), but the quality of the measurement is highly dependent on the sound-reflecting characteristics of the seafloor at that particular spot. This triggers a demand for a redesign of the system's collection capacities in SeaClear2.0 to enhance the capabilities and robustness of the system.

1.3 Deliverable objective and Document Structure

In the SeaClear2.0 system, we focus on retrieving larger and more irregularly shaped debris, including tires, bicycles, and pipes. This represents a significant change from SeaClear1.0, as the objects are now 100 times heavier (up to 250 kg). In order to achieve collecting such items, a new smart grapple has been designed and novel capabilities developed. This activity is in direct line with objective 8 (O8): "Enhance, build and test mechanisms for efficient marine waste collection of various types of waste on different types of seafloor and on the surface" and contributes towards fulfilling the following Key Performance Indicators (KPIs):

- KPI 8.1: Collect the objects from the seafloor: success rate 90%
- KPI 8.2: Collect the objects from the surface: success rate 90% KPI 8.3: Successful collection of various litter types: minimum 3

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

This deliverable documents the perception and control software framework developed for the Smart Grapple, addressing the challenges of localizing and precisely manipulating debris items under environmental disturbances. The report is structured as follows:

- Section 2 describes the Smart Grapple control stack, including state estimation via sensor fusion, control allocation for the eight-thruster configuration with power constraint handling, joint control and calibration procedures, and the operational state machine governing the autonomous cycle.
- Section 3 presents the holistic optimal control approach for the integrated USV-USG system, covering Euler-Lagrange dynamic modeling, stochastic trajectory optimization using Risk-Aware Sample Average Approximation, and feedback augmentation for robust control under uncertainties.
- Section 4 details the perception system for litter localization, integrating YOLOv8-based underwater object detection with TSDF-based 3D reconstruction for metric geometry estimation.
- Section 5 describes the Smart Grapple grasping approach, including the torque-monitor adaptive grasping algorithm, forward and inverse kinematics solutions (analytical and FABRIK-based), optimal grasping posture analysis under joint constraints, and gripping points optimization.
- Section 6 presents the digital twin implementation using the MARUS simulator for software-in-the-loop testing between field deployments.
- Section 7 provides experimental validation through laboratory tests demonstrating 250 kg grasping capability and field results from the Marseille 2025 demonstration.
- Section 8 concludes with a summary of achievements and future work directions.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

2 Smart Grapple control stack

This section describes the software architecture and control modules that enable autonomous and manual operation of the Smart Grapple system. The control stack integrates state estimation using an Extended Kalman Filter that fuses IMU and DVL sensor data for position and velocity estimation, thruster control allocation for six-degree-of-freedom maneuvering, and joint control for the eight SensoJoint actuators via EtherCAT communication within the ROS 2 framework. A hierarchical state machine orchestrates the complete operation cycle through deploying, planning, tracking, visual servoing, grasping, and recovery phases, with provisions for error handling and seamless transitions between autonomous and manual control modes through dedicated graphical user interfaces.

2.1 Smart Grapple State Estimation

The goal of state estimation is to compute the approximate position and velocity of the USG. This information is required for trajectory tracking during locomotion of the robot, as described in Section 2.4.3. The proposed approach leverages an Extended Kalman Filter (EKF) on the measured sensor data to achieve state estimation.

Sensor fusion. Two types of sensors are used as data sources for state estimation: an Inertial Measurement Unit (IMU) and a DVL. The IMU provides measurements of linear acceleration and angular velocity, while the DVL provides measurements of linear and angular velocity and altitude of the vehicle relative to the seabed. In order to increase the precision of state estimation, the IMU and DVL measurements are fused together via the EKF [1].

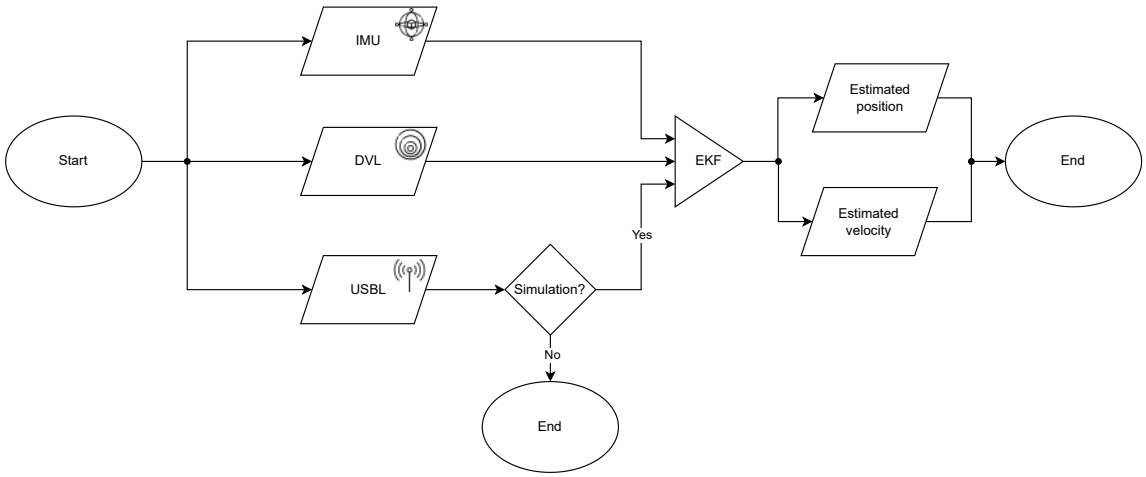


Figure 3: Information flow of the state estimation

Implementation. The *robot_localization* package, developed by Charles River Analytics, is used to implement the EKF [2]. It is an open-source package that provides a basic framework for localization tasks that can be easily adapted to different robotic systems such as the USG. The main benefits of this package are:

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

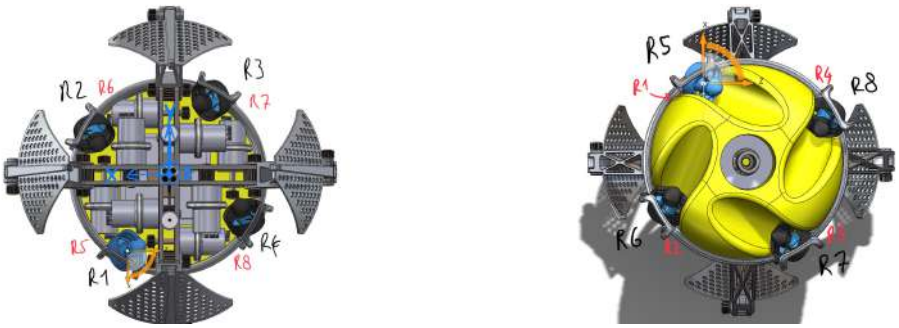
- Supports ROS 2 out-of-the-box,
- Handles an unlimited number of inputs from multiple sensor types,
- Allows 3D localization,
- Provides higher control over the sensor data and how it impacts the estimation.

Simulation. The state estimation module is part of the digital twin. The data sources for sensor fusion in simulation are slightly different as they also include Global Navigation Satellite System (GNSS) measurements. These measurements are used to emulate a Ultra-Short Baseline (USBL) acoustic positioning system, which will be added to the USG hardware in the future. For more details on the digital twin, refer to Section 6.

2.2 Smart Grapple Control Allocation

This section describes the control allocation system that distributes commanded wrenches across the eight thrusters mounted on the USG. The allocation matrix is derived from the geometric configuration of the thrusters, mapping individual thrust commands to the resulting six-degree-of-freedom wrench at the center of mass. To handle the 5 kW power constraint shared among all thrusters, a quadratic programming formulation with a percentage phase-back post-processing strategy is employed, which maintains real-time computational feasibility (average runtime of 18 ms) while achieving tracking performance within 15.6% of the optimal nonlinear solution.

2.2.1 Thruster Positions on the Hardware



(a) Bottom view (looking along $-Z_{\mathrm{ref}}$). Thrusters R1–R4 shown in black (visible) and occluded units in red.

(b) Top view (looking along $+Z_{\mathrm{ref}}$). Thrusters R5–R8 shown in black (visible) and occluded units in red.

Figure 4: Thruster layout of the USG platform in the reference frame $\mathcal{F}_{\mathrm{ref}}$ blue in (a). Subfigure a depicts the bottom assembly (R1–R4), and Subfigure b the top assembly (R5–R8). Color coding: black = visible in the current viewpoint; red = occluded.

A detailed table on the positions and orientations of the thrusters can be found in Appendix A.1.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union



Figure 5: Thrusters layout on the hardware: Marseille (September 2025).

2.2.2 Derivation of the Control Allocation Matrix

Each thruster $i \in \{1, \dots, 8\}$ is rigidly mounted on the USG and defined by its pose with respect to the USG reference frame $\mathcal{F}_{\text{ref}} = (X_{\text{ref}}, Y_{\text{ref}}, Z_{\text{ref}})$.

Position Let $\mathbf{p}_i \in \mathbb{R}^3$ denote the nominal mounting position of thruster i expressed in the reference frame $\mathcal{F}_{\text{ref}}$. Since the allocation matrix is referenced to the USG's Center of Mass (CoM), located at $\mathbf{r}_{\text{COM}} = [0,\ 0,\ 0.1362]^\top$, the position vector of thruster i relative to the COM becomes

$$\mathbf{r}_i = R_z(\alpha_0)\,(\mathbf{p}_i - \mathbf{r}_{\text{COM}}), \qquad \alpha_0 = -9^\circ. \tag{1}$$

because the positions in Table 6 are expressed **without** the shift around Z_{ref} of $\alpha_0 = -9^\circ$, which is seen on Figure 4 and in the hardware Figure 5. Note that the CoM serves as the reference point for the wrench computation.

Orientation

Each thruster orientation is defined by the angles $(\theta_i^{(y)}, \theta_i^{(z)}, \theta_i^{(x)}, \psi_i^{(z)})$. With

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}, \quad R_y(\alpha) = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}, \quad R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

the rotation from $\mathcal{F}_i$ to $\mathcal{F}_{\text{ref}}$ is

$$R_i = R_z\big(\psi_i^{(z)}\big)\,\big(R_x(\theta_i^{(x)})\,R_z(\theta_i^{(z)})\,R_y(\theta_i^{(y)})\big). \tag{3}$$

The inner product of matrices in parentheses represents three consecutive intrinsic rotations, applied successively about the current y, z, x-axes of the thruster. The leading factor $R_z\big(\psi_i^{(z)}\big)$ introduces an additional extrinsic rotation about the fixed Z_{ref}-axis, which represents the shift of 90° between consecutive thrusters (Table 6).

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

Homogeneous Transformation The transformation matrix with translation and rotation:

$$^{\text{ref}}T_i = \begin{bmatrix} R_i & \mathbf{r}_i \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{4}$$

Force and Moment Mapping Each thruster produces a thrust u_i along its local $+Z_i$-axis:

$$\mathbf{f}_i^{(i)} = \begin{bmatrix} 0 \\ 0 \\ t_i \end{bmatrix}, \qquad \mathbf{f}_i = R_i\,\mathbf{f}_i^{(i)} = u_i\,\mathbf{a}_i, \quad \mathbf{a}_i \coloneqq R_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{5}$$

The resulting moment about the CoM is

$$\mathbf{m}_i = \mathbf{r}_i \times \mathbf{f}_i = u_i\,(\mathbf{r}_i \times \mathbf{a}_i). \tag{6}$$

The corresponding wrench is

$$\boldsymbol{\tau}_i = u_i \begin{bmatrix} \mathbf{a}_i \\ \mathbf{r}_i \times \mathbf{a}_i \end{bmatrix}. \tag{7}$$

Allocation Matrix Stacking all 8 thrusters gives

$$\mathbf{B} = \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_8 \\ \mathbf{r}_1 \times \mathbf{a}_1 & \cdots & \mathbf{r}_8 \times \mathbf{a}_8 \end{bmatrix} \in \mathbb{R}^{6\times 8}, \qquad \mathbf{u} = [u_1, \ldots, u_8]^\top, \tag{8}$$

such that

$$\boxed{\boldsymbol{\tau} = \begin{bmatrix} \mathbf{f} \\ \mathbf{m} \end{bmatrix} = \mathbf{B}\,\mathbf{u}.} \tag{9}$$

For this configuration, the reference frames of the thrusters are illustrated in Figure 6, with IDs according to Table 6, where the thrust direction aligns with the local Z-axis.

2.2.3 Thruster Allocation Under Power Constraints

This section builds on the approach outlined in Deliverable 3.2. Here, we focus on the thrust allocation problem for a given reference wrench. This is a module of the control software referred to as the *thrust_manager*, see an overview in Figure 7.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

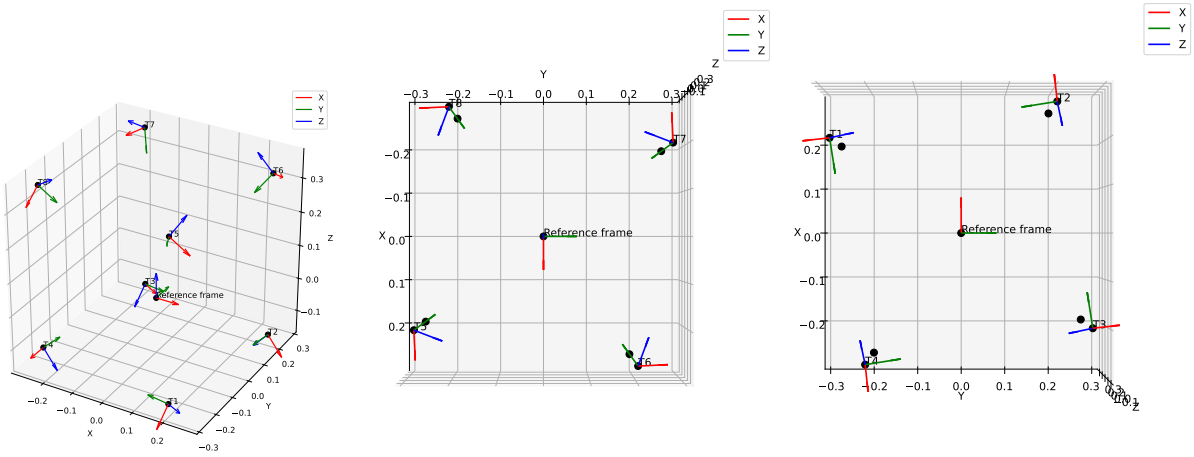


Figure 6: Thrusters frames obtained after applying the transformations from the reference frame. The blue Z-axis also matches the thrust direction.

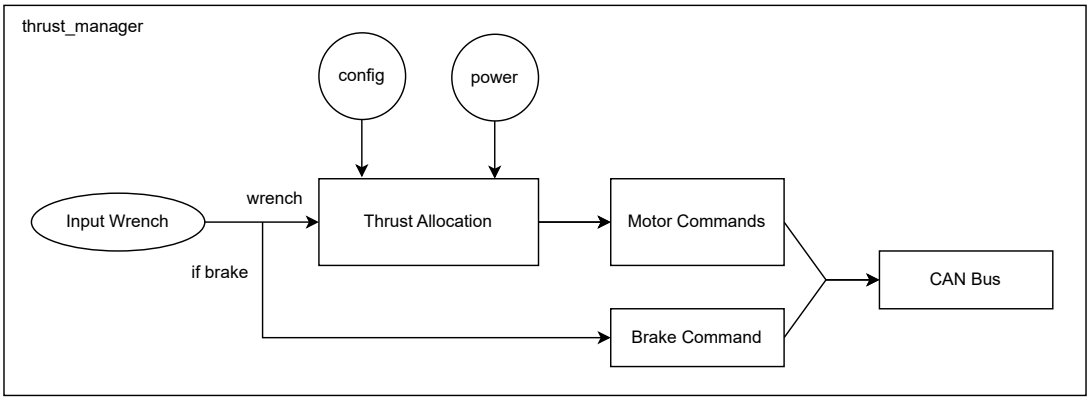


Figure 7: Block diagram of *thrust_manager*

Quadratic Program for Thrust Allocation Given a desired reference wrench $\boldsymbol{\tau}_{\text{ref}}$, the goal of thrust allocation is to compute the individual thruster inputs $u \in \mathbb{R}^8$ such that the resulting wrench tracks $\boldsymbol{\tau}_{\text{ref}}$ as closely as possible while respecting actuator limits. Let B denote the thruster allocation matrix and W a diagonal selection matrix that can deactivate faulty or intentionally disabled thrusters. The nominal allocation problem is formulated as

$$\min_{u} \ \frac{1}{2}(BWu - \boldsymbol{\tau}_{\text{ref}})^{\top} H (BWu - \boldsymbol{\tau}_{\text{ref}}) \tag{10a}$$

$$\text{subject to} \quad u_{\min} \leq u \leq u_{\max}, \tag{10b}$$

where $H \succ 0$ is a weighting matrix, and $u_{\min}, u_{\max}$ denote the lower and upper thruster bounds, respectively. The objective (10a) together with the box constraints (10b) defines a convex Quadratic Program (QP), e.g., CVXPY that can be solved efficiently in real time.

Dynamic Thruster Constraints To maintain precise control, ensure that no individual thruster is driven beyond its operational limits, and reduce mechanical wear, the thruster bounds $u_{\min}, u_{\max}$ are adapted

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

dynamically to limit the rate of change of the thruster inputs. The bounds are updated based on the previous input u_{prev} and a maximum allowed rate of change $\triangle u$.

For each thruster $i \in [8]$, we use $1, 2, \cdots, 8$ the individual bounds are computed as

$$u_{i,\min} = \begin{cases} u_{\min} & \text{if } u_{i,\text{prev}} - \triangle u < u_{\min}, \\ u_{i,\text{prev}} - \triangle u & \text{otherwise,} \end{cases}$$
$$u_{i,\max} = \begin{cases} u_{\max} & \text{if } u_{i,\text{prev}} + \triangle u > u_{\max}, \\ u_{i,\text{prev}} + \triangle u & \text{otherwise.} \end{cases} \tag{11}$$

These bounds are defined individually for each thruster and are recomputed at every allocation step. In practice, the QP (10) is solved using the updated bounds $u_{i,\min}$ and $u_{i,\max}$ instead of the static limits $u_{\min}$ and $u_{\max}$.

Available Power and Thruster Power Model The smart grapple provides only limited electrical power to the thrusters: the maximum available power is $P_{\max} = 5\,\text{kW}$, which must be shared between the eight Bluerobotics T500 thrusters and the SensoJoint drives. By contrast, if all eight thrusters operated at their maximum power at $24\,\text{V}$, the ROV would require $8 \times 1044\,\text{W} = 8.35\,\text{kW}$, exceeding the available budget. Hence, the total thruster power must satisfy $\sum_{i=1}^{8} p_i \le P_{\max}$, where p_i denotes the power of thruster i.

The mapping between thruster force (or command) and power is nonlinear. For the T500 thrusters, the total power can be approximated as

$$\sum_{i=1}^{8} k_i |u_i|^{3/2} \le P_{\max},$$

where u_i is the control input of thruster i and k_i is a fitted constant that depends on whether the thruster operates in forward or reverse mode. Including the power constraint above turns the allocation problem into a nonlinear optimization problem.

$$\begin{aligned} \min_{u} \quad & \frac{1}{2}(BWu - \boldsymbol{\tau}_{\text{ref}})^{\top} H (BWu - \boldsymbol{\tau}_{\text{ref}}) \\ \text{s.t.} \quad & \sum_{i=1}^{8} k_i |u_i|^{3/2} \le P_{\max}, \\ & u_{\min} \le u \le u_{\max}. \end{aligned} \tag{NLP}$$

To retain the efficiency of a QP formulation, we handle the power constraint via a post-processing step, as described below.

Percentage Phase Back (-pp) Since the power constraint in (NLP) is nonlinear and incompatible with the QP structure, we enforce it in post-processing using a *percentage phase back* strategy [3]. Let $u^{\star}$ be the QP solution with total power $P_{u^{\star}} = \sum_{i=1}^{8} p_i$, and let $P_{USG} \le P_{\max}$ be the power budget for the thrusters at the time when the optimization is run. If $P_{u^{\star}} > P_{USG}$, all thruster powers are scaled by

$$\alpha = \frac{P_{USG}}{P_{u^{\star}}}, \qquad p_i \leftarrow \alpha\, p_i.$$

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

The scaled power of thruster i is then mapped back to a thrust command u_i via the fitted T500 power model. For forward and reverse operation this yields

$$u_i = \begin{cases} \dfrac{-b_{\text{fwd}} + \sqrt{b_{\text{fwd}}^2 + 4a_{\text{fwd}}\alpha\, p_i}}{2a_{\text{fwd}}} & \text{(forward)}, \\ \dfrac{-b_{\text{rev}} - \sqrt{b_{\text{rev}}^2 + 4a_{\text{rev}}\alpha\, p_i}}{2a_{\text{rev}}} & \text{(reverse)}, \end{cases} \tag{12}$$

with $a_{\text{fwd}}, b_{\text{fwd}}, a_{\text{rev}}, b_{\text{rev}}$ denoting the direction-dependent T500 parameters. This approach allows us to first solve the fast QP (10) without the nonlinear power constraint and then enforce the power limit by a simple scaling and back-transformation, slightly perturbing optimality but preserving real-time feasibility.

To assess the impact of the power constraint and the choice of allocation method, we generate a dataset of 1000 reference wrenches $\boldsymbol{\tau}_{\text{ref}}$, representing desired forces on the USG. For illustration, all moments are set to zero, as the dominant motion is translational. The samples are obtained from a noisy spiral path around a sphere. All tests are executed on an Intel(R) Core(TM) i5-6200U CPU @ 2.30 GHz (4 cores), and the thruster power budget is fixed to $P_{\text{thrust}} = 5000\,\text{W}$. We compare the QP-pp approach against the NLP-based baseline (NLP) in terms of tracking error (10a) of the desired reference wrench $\boldsymbol{\tau}_{\text{ref}}$, power consumption, and runtime.

Approach	Error (10a) (mean $\pm$ std.) [%]	Power consumption [%]
QP-pp	$+15.6 \pm 11.4$	$+0.2$

Table 1: Relative increase in error and average power consumption of QP-pp compared to the NLP-pow baseline (not shown).

Approach	Worst case [s]	Avg. runtime [s]	Std. dev. [s]
QP-pp	**0.0369**	**0.0181**	**0.0036**
(NLP)	*3.4753*	*0.9000*	*0.4156*

Table 2: Runtimes for the QP-pp and NLP approaches.

The results as shown in Table 1 show that QP-pp incurs only a moderate increase in tracking error ($+15.6\%$ on average with $+0.2\%$ higher power consumption) compared to the NLP baseline, but achieves roughly two orders of magnitude faster runtimes (both on average and in the worst case). Consequently, QP-pp is preferred in practice, as it enables significantly higher update rates while remaining close to the best attainable performance.

Practical Implementation We adopt a modular approach that allows us to overwrite the reference signal at different stages—position, velocity, wrench, or even the duty-cycle command of individual thrusters. This flexibility proved particularly valuable for analyzing and debugging individual modules during the field tests in Marseille (September 2025). The thrusters—and consequently the smart grapple—can be commanded via a Graphical User Interface (GUI). A position target can be specified (as

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

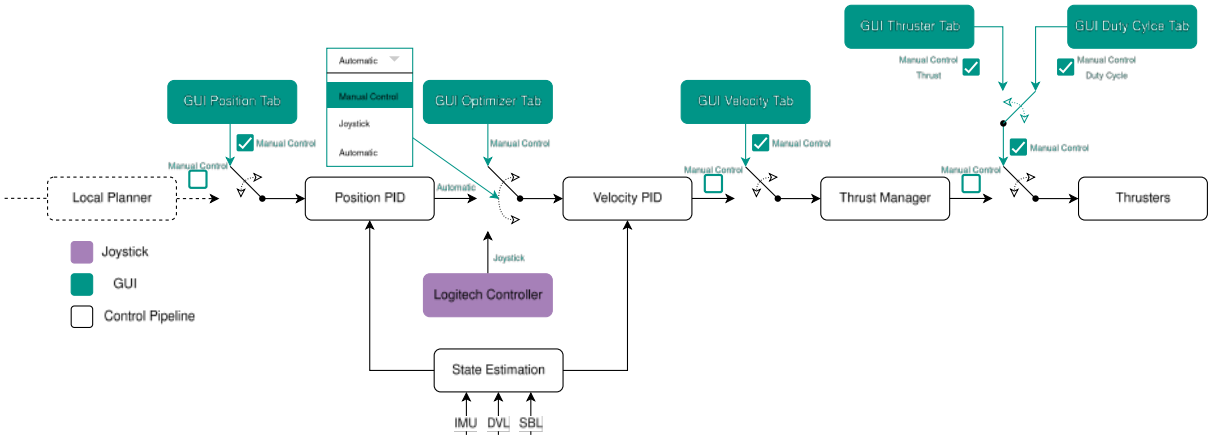


Figure 8: Architecture of the USG control pipeline software.

shown on the left in Figure 8), which is then passed through a cascaded PID controller to generate a target wrench. This wrench is then provided to the optimizer, i.e., the *thrust_manager* (see Section 2.2.3). Alternatively, a velocity command can be set directly—either through the GUI or using the Logitech gamepad controller—thereby bypassing the position Proportional-Integral-Derivative (PID) output. If no PID controller is involved, there exists the possibility of providing the reference (to track) wrench directly in the GUI.

2.3 Joint control

The smart grapple joints are actuated by the SensoJoint 5016 BLDC (brushless direct current) motors with a maximal torque of 120 Nm. The motors utilize the Ethernet for Control Automation Technology (EtherCAT) interface for communication with the computer used for the smart grapple control. In this work, the communication and control of the actuators have been integrated in the overall Robot Operating System 2 (ROS2) software architecture using Simple Open EtherCAT Master (SOEM) Library [4]. All functionalities of the motor can be accessed via a ROS2 topic or service from any other module of the system. For the convenience of the smart grapple operator, all motor commands can be executed from a GUI depicted in Figure 10.

2.3.1 Graphic user interface

The upper button panel contains the ON, OFF, fault reset, and calibration buttons. Pressing ON and OFF will turn the smart grapple on and off, respectively. Turning the smart grapple on means bringing the SensoJoints into the state of enabled operation. From this state, the motors are ready to receive user commands. However, in case of a fault indicated by the built-in fault warning, SensoJoints might enter the fault state, and they will stay in that state unless the fault is cleared by pressing the corresponding

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

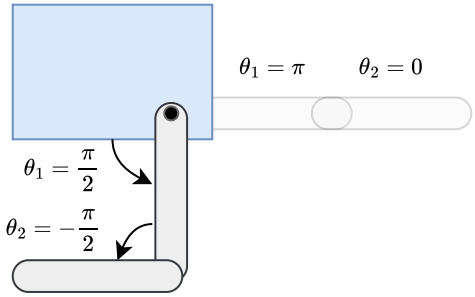


Figure 9: Grapple angle coordinate frame.

button or manually solving the issue onshore. Fault can also be reset automatically by the state machine if it is one of the warnings that can be ignored. The calibration button activates the calibration procedure explained in Section 2.3.2.

The lower button panel contains Safe Torque Off (STO) ON and STO OFF buttons. These buttons are connected to the relay inside the electronic box and are responsible for physically connecting and disconnecting the 24V STO line. SensoJoint will never be able to reach the operational state without this line connected to the power, as an additional safety measure. STO OFF acts as an emergency stop. By default, after the smart grapple has been powered up, the relay is disconnected; therefore, before the operation, the operator is required to press the STO ON button.

On the left panel, all the information regarding each joint is displayed, such as the position of the end-effector, velocity, sensed torque, and the fault message. On the central panel, the user has the possibility to control the smart grapple using so-called percentage mode, where 0% corresponds to the fully closed state and 100% to the fully opened state. The joint configuration corresponding to each percentage level is implemented as described in Sec. 5.3.1, Case 3. The "Joystick control" checkbox above indicates if the joystick is chosen as the source of input; checking it will disable the slider to avoid conflicting control inputs. The table below the slider displays the joint angles in the smart grapple angle coordinate frame. The smart grapple angle coordinate frame is illustrated in Figure 9. The right panel contains buttons for the relative positioning of the individual joints. By pressing each button, the user opens or closes the joint by 0.05 radians. Using this GUI, the onshore operator has full monitoring and control over the smart grapple fingers for the manual grasp.

2.3.2 Calibration

The calibration routine has been implemented to approximate the angle difference between the encoder zero ("factory zero") and zero in the smart grapple coordinate frame (Figure 9). Usually, it must be conducted once, after the smart grapple joint has been reassembled, for example, during routine maintenance. It might also be executed in case of the belt slippage in case of a high joint load. The procedure starts with driving the shoulder joints towards their mechanical limit until the predefined torque threshold is reached. Then, the shoulder joints drive back at a small angle to approach the mechanical limit at a lower speed for precision. After that, they are driven on a fixed angle increment between the mechanical limit and the joint angle corresponding to $\theta_1 = \pi$ in the joint coordinate frame. The same

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

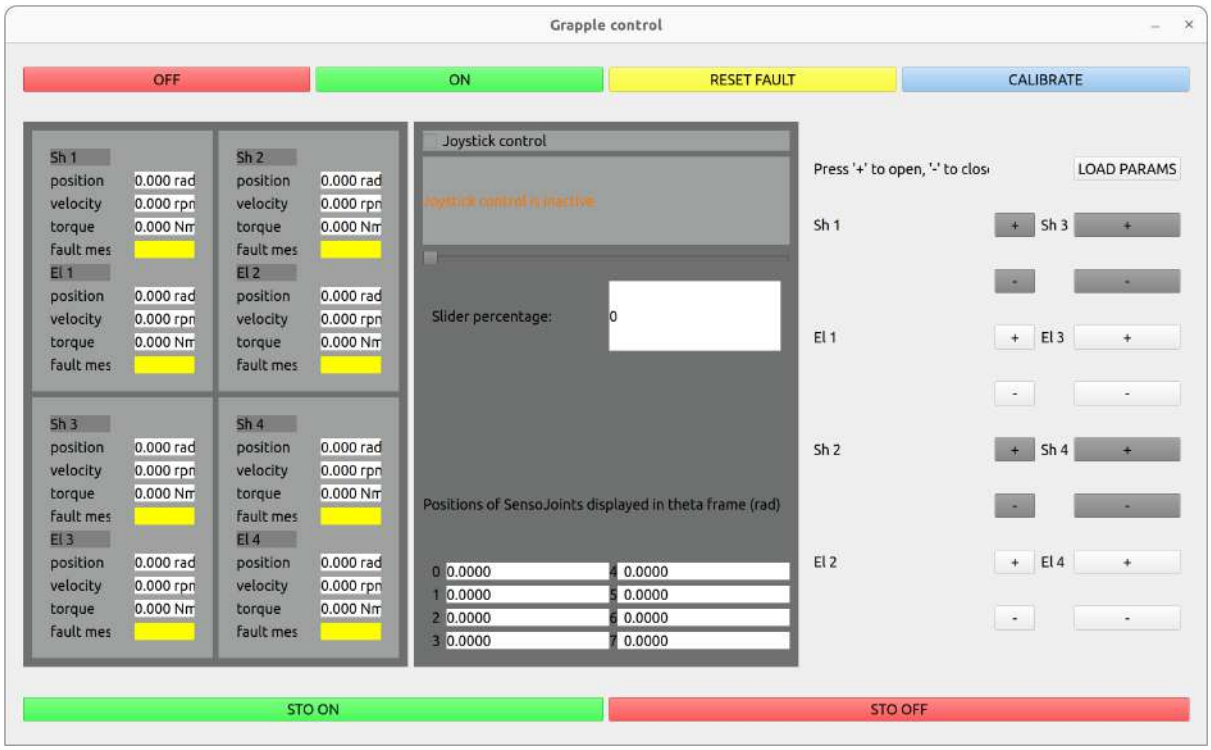


Figure 10: GUI for the smart grapple joint control.

procedure happens then with the elbow joints, but their final position is $\theta_2 = 0$. After the smart grapple has reached its final starfish-like configuration, the current motor angles are written in the parameter file as $\Delta\theta_1, \Delta\theta_2$ and later used for angle transformations between encoder angles and smart grapple angles.

2.4 State machine of the operation cycle

The finite state machine is designed to automate the processes of planning, trajectory tracking, precise positioning, grasping, and trash retrieval. The whole cycle of operation is divided into several states, based on the current stage of operation. Transition between states happens in a deterministic manner when particular conditions are fulfilled. The state machine is designed to be responsible for sending activation commands, monitoring the progress, and aborting the operation cycle in case of an error, such as an internal hardware failure. The overall state flow is presented in Figure 11.

2.4.1 Deploying

In this state, the initialization procedure happens. The initial storage position of the smart grapple is grasping the collecting net between the SeaCAT USV hulls to prevent the unwanted pendulum effect during transportation. Therefore, the deploying state begins with releasing the basket and sending the signal to the SeaCAT USV to deploy the grapple under the water. Then, the smart grapple has to be

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

brought into the operational state, which includes a possible joint calibration routine and internal fault check. To be able to transfer to the next state, the smart grapple has to receive the target coordinates as well as the seabed map in the form of an octree map (octomap) [5] from the miniTortuga UUV.

2.4.2 Planning

As soon as the necessary data are received, no system faults are present, and all components are in the operational state, the smart grapple moves to the planning state. The informed Rapidly Exploring Random Tree Star (RRT*) [6] is currently used in the software to plan a collision-free path, based on the octomap, to the first trash object in the received list. Although RRT* is used in the current implementation of the control pipeline, it can be replaced by the holistic optimal control approach described in Section 3. Once the planning is complete, the state machine activates the tracking state.

Practical implementation The informed RRT* algorithm is implemented using the open-source Open Motion Planning Library (OMPL) [7]

2.4.3 Tracking

The tracking procedure consists of a local planner, state estimation, and the cascaded PID controller. The latter then sends the thruster commands according to the architecture illustrated in Figure 8. As it is seen from the figure, it is possible to disconnect the direct feedback loop and control the smart grapple using a joystick for debugging purposes. The local planner iterates through the trajectory points provided by the global planner and notifies the state machine as soon as the last one has been reached. Then the tracking state is considered as finished. It is possible that, due to the environmental conditions, such as strong tides, the smart grapple will not position itself directly above the target. In this case, a precise positioning procedure is required to reposition the grapple for the grasp. The next state depends on two factors: whether the user wants to proceed with the manual grasp and whether the visual servoing is feasible. During the development process, the manual grasp is selected, and the smart grapple is navigated solely by the operator onshore using the joystick and the GUI to control the fingers. The availability of the visual servoing depends on the visibility conditions of the environment. In case of good visibility, the smart grapple makes a state transition to the visual servoing state, described below. In case of bad visibility, the smart grapple has to position itself based on the last known position of the trash.

2.4.4 Visual servoing

The visual servoing state has the purpose of camera-driven, precise position control of the smart grapple. This state is triggered upon reaching the final path point, provided that water clarity is sufficient for object detection. In this phase, the reference position for the controller is expressed in the camera-based coordinate frame. (Figure 11). Specifically, the object detection model (Section 4.1) identifies the target and provides its bounding box coordinates, while the optimal grasp point estimation algorithm (Section 5.4) determines the desired grasp pose. The control objective is to center the bounding box within the onboard camera's horizontal frame and reach the desired altitude (pose) vertically using the DVL feedback. Once the desired pose is achieved, the state machine transitions to the grasping state.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

2.4.5 Grasping

Automated grasp The automated grasping state consists of its own state machine illustrated in Figure 20. The general state machine is responsible for initiating the grasping procedure by calling a corresponding ROS2 service and passing the desired contact points for the grasp from the algorithm described in Section 5.4 (if available). During the automated grasp, it is possible that the actuators activate the built-in maximum torque warning and enter a fault state. In this case, the state machine is responsible for sending the fault-clearing signal unless the fault cannot be cleared automatically (e.g., a finger has to be manually opened to release the torque), and the smart grapple must be switched to manual control.

Manual grasp The manual grasp is controlled using the GUI illustrated in Figure10. For the manual grasp to be considered as completed, the user has to initiate the state transition from grasping to dropping by calling a corresponding ROS2 service.

In case of a successful grasp, the state transitions to the dropping (recovering) state.

2.4.6 Dropping (Recovering)

In the dropping state, the state machine sends the signal to the USV Tender (illustrated in Figure 1), initiating the retrieval of the object grasped. The smart grapple sends the signal to roll the tether, positioning itself above the water, and waits for the USV Tender to position itself below. After receiving the signal, approving the object drop, it opens and measures the joint torques. It might happen that due to the seaweed and sediment, the object or parts of it get stuck in the shovels or on the joints. If the excess torque is sensed, the smart grapple will try to repeat the cycle of opening and closing k times , before reporting an error due to the impossibility to continue its operation. The number of tries k is a system parameter that can be set by a user. After the successful dropping, the smart grapple can proceed with the next object in the list, or the operator can declare the operation finished, and the smart grapple will send a signal to the SeaCAT USV to be brought back to shore.

2.4.7 Error

The error state is activated in case of an error in any state of operation. The error state means that the state machine notifies the onshore operator about its inability to continue the cycle, and the SeaCAT USV brings the smart grapple back to the shore for maintenance. The possible errors include

1. SensoJoint faults that cannot be resolved automatically
2. Thruster communication failure
3. Object being stuck in the shovels or in the joints
4. User-defined error, when the error state is activated by the operator

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

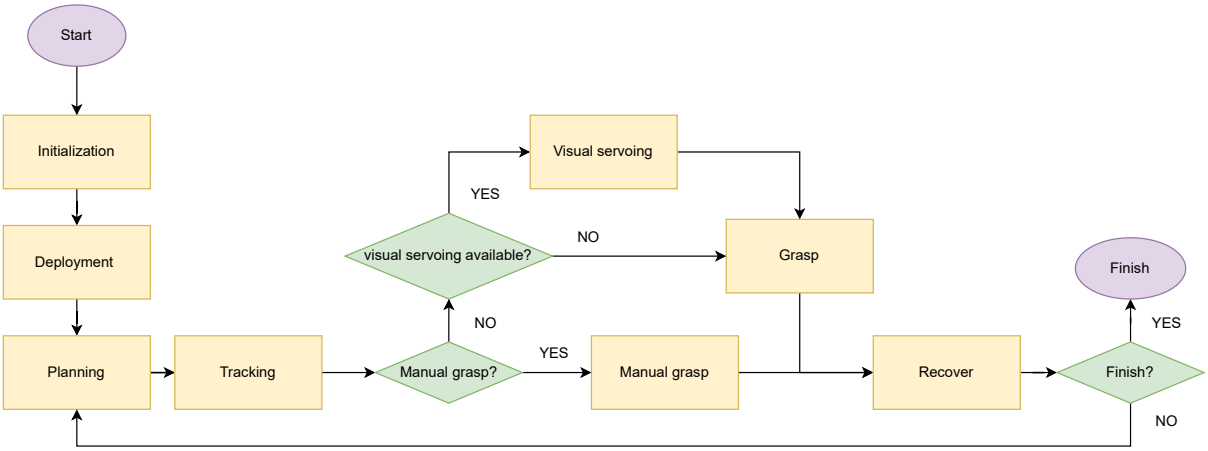


Figure 11: Operation state flow.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

3 Holistic Optimal Control Approach

This section presents a unified optimal control framework for the integrated USV-USG system that accounts for the coupled pendulum dynamics between the surface vessel and the underwater grapple. The approach formulates the trajectory planning problem as a stochastic optimal control problem using Risk-Aware Sample Average Approximation (RA-SAA), which explicitly handles uncertainties in payload mass, environmental disturbances, obstacle positions, and model parameters. The formulation minimizes energy consumption while satisfying Conditional Value at Risk (CVaR)-based obstacle avoidance constraints, terminal positioning requirements, power limits, and tether tension constraints. A feedback augmentation scheme is introduced to center sampled trajectories around a nominal solution, enabling robust high-frequency online control while the optimization plan operates at lower frequency. This scheme can be used to replace the traditional trajectory planning method, e.g., RRT*.

3.1 Modeling of the Integrated USV-USG

The goal in modeling is to simulate the USV-USG behavior with a minimum number of states and as little computational time as possible, without losing any important characteristics. We make the following assumptions for simplification:

- the tether is light and thin, such that its gravity, inertia and friction can be neglected
- the tether is attached to the USG at its center of mass and the orientation of the USG is totally defined by the tether. Meaning the USGs z-axis z_{bUSG} is pointing parallel to the tether.
- The USG rotation around its z-axis is small and does not affect the movement in any relevant manner

3.1.1 Coordinate Frame and State Definition

We define the following reference frames:

- $\{e\}$ is the earth fixed frame
  (used if not specified otherwise)
- $\{bUSV\}$ is the body frame of USV.
- $\{bUSG\}$ is the body frame of USG

Earth and USV frame are defined with the z-axis pointing downwards and use of roll-pitch-yaw angles for orientation, as is typical for airplanes. We also define a joint space based on a floating base for the USV combined with the USG relative position in spherical coordinates. For the definition of the angles see Figures 12 and 13.

The state of the USV-USG system can be written as follows:

$$\mathbf{q} = [x_{USV}, y_{USV}, z_{USV}, \gamma_{USV}, l, \phi, \theta] \tag{13}$$

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

x, y, z_{USV}	position of USV w.r.t. earth frame
γ_{USV}	yaw angle of USV w.r.t. earth frame
l	tether length
ϕ	tilt: angle around fixed USV y axis (swinging in x-z plane)
θ	pan: angle around fixed USV x axis (swinging in y-z plane)

The z-position of the USV is not actuated but included in the joint vector since the waves are acting on this coordinate and represent a relevant source of disturbance.

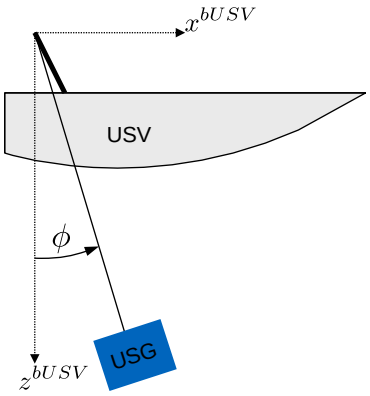


Figure 12: spherical angle definition in x-z plane

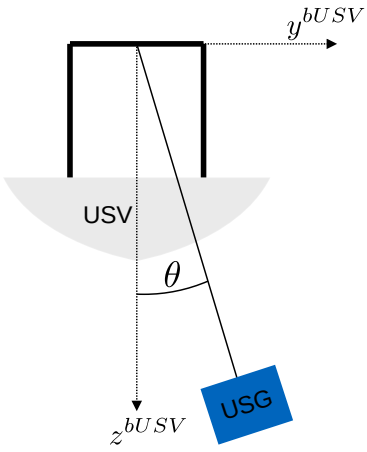


Figure 13: spherical angle definition in y-z plane

3.1.2 Modeling Dynamics

The modeling is performed using the Euler-Lagrange equations, and therefore, the kinetic and potential energies for each part are required. Starting with the USV, the kinetic energy is given by its linear and angular motion:

$$K_{USV} = \frac{1}{2}\dot{\mathbf{x}}_{USV}^{T}\mathbf{R}_{USV}^{e}\mathbf{M}_{USV}{\mathbf{R}_{USV}^{e}}^{T}\dot{\mathbf{x}}_{USV} + \frac{1}{2}{\boldsymbol{\omega}_{USV}^{e}}^{T}\mathbf{R}_{USV}^{e}\mathbf{I}_{USV}{\mathbf{R}_{USV}^{e}}^{T}\boldsymbol{\omega}_{USV}^{e} \tag{14}$$

where the rotation boils down to only the rotation around the z-axis. The potential energy P_{USV} is given by gravity and buoyancy:

$$P_{USV} = -m_{USV}gz_{USV} + b_{USV}(z_{USV} - z_{water})^2 \tag{15}$$

We model the buoyancy similarly to a spring, with the spring constant representing the buoyancy factor

$$b_{USV} = A_{USV,dis}\rho_{water}g \tag{16}$$

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

m_{USV}	weight of the USV
z_{water}	height of the water
$A_{USV,dis}$	displacement area of USV at the water surface
ρ_{water}	density of the water

The next system component is the tether, including the winch to move it. We model its kinetic energy by a simple inertia I_l acting on the first derivative of the tether length $\dot{l}$:

$$K_l = \frac{1}{2} I_l \dot{l}^2 \tag{17}$$

Last but not least we have the energies of the suspended USG. The potential energy is given by the USGs mass in water $\overline{m}_{USG}$

$$P_{USG} = -\overline{m}_{USG}\, g\, z_{USG} \tag{18}$$

and the kinetic energy is given by its motion:

$$\begin{aligned} K_{USG} = &\frac{1}{2} {\dot{\mathbf{x}}^e_{USG}}^T \mathbf{R}^e_{USG} \mathbf{M}^{USG}_{USG} {\mathbf{R}^e_{USG}}^T \dot{\mathbf{x}}^e_{USG} \\ &+ \frac{1}{2} {\omega^e_{USG}}^T \mathbf{R}^e_{USG} \mathbf{I}^{USG}_{USG} {\mathbf{R}^e_{USG}}^T \omega^e_{USG} \end{aligned} \tag{19}$$

Since we later work with the joint state, we have to transform it by using the Jacobian:

$$\dot{\mathbf{x}}^{USG}_{USG} = \mathbf{J}^{USG}_{USG} \dot{\mathbf{q}} \tag{20}$$

$$\begin{aligned} K_{USG} = &\frac{1}{2} (\mathbf{J}^e_{USG,lin} \dot{\mathbf{q}})^T \mathbf{R}^e_{USG} \mathbf{M}^{USG}_{USG} {\mathbf{R}^e_{USG}}^T \mathbf{J}^e_{USG,lin} \dot{\mathbf{q}} \\ &+ \frac{1}{2} (\mathbf{J}^e_{USG,ang} \dot{\mathbf{q}})^T \mathbf{R}^e_{USG} \mathbf{I}^{USG}_{USG} {\mathbf{R}^e_{USG}}^T \mathbf{J}^e_{USG,ang} \dot{\mathbf{q}} \end{aligned} \tag{21}$$

Similar transformations can be done for all kinetic and potential energies, and we can define the Lagrangian in joint space by:

$$\begin{aligned} L(\mathbf{q}, \dot{\mathbf{q}}) = &K_{USV}(\mathbf{q}, \dot{\mathbf{q}}) + K_l(\dot{l}) + K_{USG}(\mathbf{q}, \dot{\mathbf{q}}) \\ &- (P_{USV}(\mathbf{q}) + P_{USG}(\mathbf{q})) \end{aligned} \tag{22}$$

$$L(\mathbf{q}, \dot{\mathbf{q}}) = K(\mathbf{q}, \dot{\mathbf{q}}) - P(\mathbf{q}) \tag{23}$$

By applying the Euler-Lagrange equation

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L}{\partial \mathbf{q}} = \mathbf{f} \tag{24}$$

and reordering the terms, we get the dynamics in standard robotics form:

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathbf{f} \tag{25}$$

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

3.1.3 External Forces

In addition to the internal forces captured by the Euler-Lagrange equation, some external forces act on the system. These are caused by friction (or drag) and actuation under the low velocities condition of USV. The friction forces are divided into three origins, from USV, winch and USG:

$$\boldsymbol{\eta}_D = \boldsymbol{\eta}_{USV} + \boldsymbol{\eta}_l + \boldsymbol{\eta}_{USG} \tag{26}$$

For USV and winch, a linear friction is assumed. For the winch, it is modeled as

$$\eta_l(\dot{l}) = c_l \dot{l} \tag{27}$$

In case of the USV the friction coefficients differ in the USV's x, y and z direction. Thus, the velocities in USV body frame have to be used, and this requires the use of the body frame Jacobian J^{USV}_{USV} Our method didn't assume linearized Jacobian because the operation range may not be small. to transform from joint-space to body frame velocities and from body frame forces back to joint space forces.

$$\dot{\mathbf{x}}^{USV}_{USV} = \mathbf{J}^{USV}_{USV}\dot{\mathbf{q}} \tag{28a}$$
$$\mathbf{f}^{USV}_{\eta USV} = \text{diag}(\mathbf{c}_{USV})\dot{\mathbf{x}}^{USV}_{USV} \tag{28b}$$
$$\eta_{USV} = \mathbf{J}^{USV^T}_{USV}\mathbf{f}^{USV}_{\eta USV} \tag{28c}$$

Since the USG is totally submerged by water, we consider a linear friction model as insufficient. Following [8] we model the USG's friction with a second-order expression. Again, the transformation to the body frame is necessary:

$$\dot{\mathbf{x}}^{USG}_{USG} = \mathbf{J}^{USG}_{USG}\dot{\mathbf{q}} \tag{29a}$$
$$\mathbf{f}^{USG}_{\eta USG} = \text{diag}(\mathbf{c}_{USG})\text{diag}(|\dot{\mathbf{x}}^{USG}_{USG}|)\dot{\mathbf{x}}^{USG}_{USG} \tag{29b}$$
$$\eta_{USG} = \mathbf{J}^{USG^T}_{USG}\mathbf{f}^{USG}_{\eta USG} \tag{29c}$$

The other source of external force is the actuation. The USV has two stern thrusters pointing forward and two bow thrusters pointing to the sides. With this constellation the USV can move in the x and y directions and rotate around z-axis. The USV's z direction can not be actuated. The winch is actuated by an electric motor and a gearbox. In terms of energy efficiency, the winch is the most efficient actuator in the system. The USG has a total of eight thrusters aligned such that a full six degrees of freedom actuation is possible.

To keep the model simple, we consolidate the actuations and consider them to be performed in joint space. This means for every joint defined in (13), except the USV's z position, we model an input force. We further assume a first-order delay between the control signal $\mathbf{u}$ and the input force $\mathbf{f_u}$. With $\mathbf{T}$ as the diagonal matrix with the time constants, we model the input force as

$$\dot{\mathbf{f}}_\mathbf{u}(t) = \mathbf{T}^{-1}(-\mathbf{f}_\mathbf{u}(t) + \mathbf{u}(t)) \tag{30}$$

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

3.1.4 Model Summary

All previously modeled parts can now be combined to one set of model equations:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \boldsymbol{\eta}_D = \mathbf{f_u} \tag{31a}$$

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1}\left(-\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) - \boldsymbol{\eta}_D + \mathbf{f_u}\right) \tag{31b}$$

$$\boldsymbol{\eta}_D = \boldsymbol{\eta}_{USV} + \boldsymbol{\eta}_l + \boldsymbol{\eta}_{USG} \tag{31c}$$

$$\dot{\mathbf{f}}_\mathbf{u}(t) = \mathbf{T}^{-1}(-\mathbf{f_u}(t) + \mathbf{u}(t)) \tag{31d}$$

We introduce a combined state $\mathbf{z}$ and bring the model equation into control-affine form:

$$\mathbf{z} = [\mathbf{q}^T, \dot{\mathbf{q}}^T, \mathbf{f}_\mathbf{u}^T]^T \tag{32a}$$

$$\dot{\mathbf{z}} = [\dot{\mathbf{q}}^T, \ddot{\mathbf{q}}^T, \dot{\mathbf{f}}_\mathbf{u}^T]^T \tag{32b}$$

$$\dot{\mathbf{z}} = \mathbf{f}_{sys}(\mathbf{z}) + \mathbf{G}_{sys}(\mathbf{z})\mathbf{u} \tag{32c}$$

3.2 Technical Approach

With the given model, the next step is to think about the control objective. The goal is to generate trajectories to a desired terminal position. The generated trajectory shall account for

- obstacles with uncertain position
- uncertain parameters (i.e., uncertain payload mass and friction from gripped object)
- process noise

We assume all uncertainties mentioned above have a known probability distribution.

3.2.1 Stochastic Optimal Control Problem

We base our solution on the Trajectory Optimization with the RA-SAA introduced by [9]. In the same manner, we introduce sampled system models $i \in \{1, \dots N\}$ with parameters $\boldsymbol{\xi}^i$ drawn from the parameter distribution. The difference from the original is an alternative cost function and different constraints that fit our proposed model. Before we take a closer look at the cost and constraint functions, we formulate the dynamic system equation as a Stochastic Differential Equation (SDE) with uncertain parameters $\boldsymbol{\xi}$ and process noise $\mathbf{D}d\boldsymbol{\omega}_t$:

$$d\mathbf{z_u} = \mathbf{f}_{sys}(\mathbf{z},\boldsymbol{\xi})dt + \mathbf{G}_{sys}(\mathbf{z})\mathbf{u}dt + \mathbf{D}d\boldsymbol{\omega}_t \tag{33}$$

Following the Sampling Average Approach, we define multiple samples $\boldsymbol{\xi}^i$ of the uncertain parameters and thus get multiple samples of the SDE and trajectory:

$$d\mathbf{z}_\mathbf{u}^i = \mathbf{f}_{sys}(\mathbf{z}^i,\boldsymbol{\xi}^i)dt + \mathbf{G}_{sys}(\mathbf{z}^i)\mathbf{u}dt + \mathbf{D}d\boldsymbol{\omega}_t^i \tag{34}$$

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

Cost Function Based on the sampled trajectories, we define our optimization goal, the cost function, as the expected energy used for the trajectory as

$$\inf_{\mathbf{u}\in\mathcal{U}} \frac{1}{N}\sum_{i=1}^{N}\left[\int_0^{t_f} \dot{\mathbf{q}}^i \mathbf{E}_\eta \mathbf{f}_\mathbf{u}^i\right] \tag{35}$$

with N the number of samples, $\mathbf{E}_\eta$ a diagonal weighting matrix representing the energy efficiency, t_f the time horizon, and $\mathcal{U}$ the control space.

Constraints For the optimization, we have the following constraints to consider:

- obstacle avoidance constraint
- terminal constraint (e.g., to reach a desired state)
- power and input constraint
- tether tension and tether length constraint

For obstacle avoidance, we adopt the CVaR-based constraint formulation proposed in [9]. This approach is advantageous because it strikes a balance between conservativeness and feasibility while capturing the varying severity of unsafe events—for instance, distinguishing between a gentle seabed contact and a high-speed collision. In this formulation, the CVaR constraint enforces chance constraints by ensuring that the probability of constraint violation remains below a specified risk level α.

Given an ellipsoidal obstacle $\mathcal{O}_j$ centered at $\mathbf{o}_j$ with shape matrix $\mathbf{Q}_j$, and a parameter sample $\boldsymbol{\xi}^i$ drawn from the parameter distribution, the obstacle can be represented using a signed distance function:

$$d_{\mathcal{O}_j(\boldsymbol{\xi}^i)}(\mathbf{z}) = 1 - (\mathbf{z} - \mathbf{o}_j(\boldsymbol{\xi}^i))^\top \mathbf{Q}_j (\mathbf{z} - \mathbf{o}_j(\boldsymbol{\xi}^i))$$

Then, for M obstacles, the constraint function G yields

$$G(\mathbf{z}_\mathbf{u}^i, \boldsymbol{\xi}^i) = \max_{j=1,\dots M} d_{\mathcal{O}_j(\boldsymbol{\xi}^i)}(\mathbf{z}_\mathbf{u}^i) \tag{36}$$

Then the CVaR formulation of this constraint [9] reads

$$\inf_{s\in\mathbb{R}} \; s + \frac{1}{\alpha N}\sum_{i=1}^{N} \max\left(\sup_{\tau\in[0,t_f]} G_j(\mathbf{z}_u^i(\tau), \boldsymbol{\xi}^i) - s, 0\right) \leq 0 \tag{37}$$

where α the risk level and $\mathbf{z}_\mathbf{u}$ as the position of coupled USG-USV system.

The terminal constraint is responsible for actually reaching the target position at the end of time horizon t_f. To faciliate the resolution of the Optimal Control Problem (OCP) , we consider the terminal position $\mathbf{z}_d$ as reached, if the expected value lies inside a small environment around the desired position shaped by the diagonal matrix $\mathbf{R}_T$. The goal of closing target position is formulated as follows

$$\frac{1}{N}\sum_{i=1}^{N} H(\mathbf{z}_u^i(t_f)) \leq \delta_M, \tag{38a}$$

$$H(\mathbf{z}_\mathbf{u}^i(t_f)) = (\mathbf{z}_\mathbf{u}^i(t_f) - \mathbf{z}_d)^T \mathbf{R}_T (\mathbf{z}_\mathbf{u}^i(t_f) - \mathbf{z}_d) \tag{38b}$$

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

The power constraint takes the power limit of the USG into consideration. Since the USG actuators are only used to actuate the spherical angles ϕ and θ the constraint is formulated as

$$\dot{\phi}^i f_\phi^i + \dot{\theta}^i f_\theta^i \leq P_{USG,max} \tag{39}$$

Due to the control in joint space, the input constraints are only roughly approximated. With the stacked body frame forces $\mathbf{f}_{all} = [\mathbf{f}_{USV}^{USV^T}, f_l, \mathbf{f}_{USG}^{USG^T}]^T$ and corresponding stacked Jacobian $\widetilde{\mathbf{J}}_{all}(\mathbf{q})$ we define the constraint as

$$\left|\widetilde{\mathbf{J}}_{all}(\mathbf{q}^i)^T\right| \mathbf{f}_{all,min} \leq \mathbf{f}_\mathbf{q} \leq \left|\widetilde{\mathbf{J}}_{all}(\mathbf{q}^i)^T\right| \mathbf{f}_{all,max} \tag{40}$$

with $\left|\widetilde{\mathbf{J}}_{all}^T\right|$ as the element-wise absolute value to give an upper bound. Last but not least the tether constraints enforce the tether to be always under tension and positive in its length:

$$f_l^i \leq 0 \tag{41}$$
$$l > 0 \tag{42}$$

Optimization Problem By combining all constraints with the cost function, we obtain a tractable optimization problem that computes the optimal control trajectory over the horizon t_f, accounting for uncertain dynamics, model mismatch, and uncertain obstacles, as

$$\inf_{\mathbf{u}\in\mathcal{U}} \frac{1}{N}\sum_{i=1}^{N}\left[\int_0^{t_f} \dot{\mathbf{q}}^i \mathbf{E}_\eta \mathbf{f}_\mathbf{u}^i\right] \tag{43a}$$

s.t.

$$\inf_{s\in\mathbb{R}} \; s + \frac{1}{\alpha N}\sum_{i=1}^{N}\max_j\left(\sup_{\tau\in[0,t_f]} G_j(\mathbf{z}_u^i(\tau), \boldsymbol{\xi}^i) - s, 0\right) \leq 0 \tag{43b}$$

$$-\delta_M \leq \frac{1}{N}\sum_{i=1}^{N} H(\mathbf{z}_u^i(t_f)) \leq \delta_M, \tag{43c}$$

$$\dot{\phi}^i f_\phi^i + \dot{\theta}^i f_\theta^i \leq P_{USG,max} \tag{43d}$$

$$\left|\widetilde{\mathbf{J}}_{all}(\mathbf{q}^i)^T\right| \mathbf{f}_{all,min} \leq \mathbf{f}_\mathbf{q} \leq \left|\widetilde{\mathbf{J}}_{all}(\mathbf{q}^i)^T\right| \mathbf{f}_{all,max} \tag{43e}$$

$$f_l^i \leq 0 \tag{43f}$$

$$\mathbf{z}_u^i \text{ satisfies dynamics from (34)} \tag{43g}$$

However, the above optimal control problem can be difficult or even infeasible to solve, as large parameter uncertainties may cause divergent trajectories $\mathbf{z}_\mathbf{u}^i$. To mitigate this, we introduce a feedback control term and optimize over closed-loop controllers.

3.2.2 Feedback Augmentation

We first introduce a nominal model and its trajectory $\mathbf{z}_\mathbf{u}^{nom}$ with nominal parameters $\boldsymbol{\xi}^{nom}$ and design a feedback loop, following [8]. This draws all trajectories towards the nominal trajectory. We design

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

this feedback controller as a PD controller based on the joint position $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$. Formulated in terms of $\mathbf{z}$ we get

$$\mathbf{f}_{fb}^{i}(t) = [\mathbf{K}_P\ \mathbf{K}_D\ \mathbf{0}](\mathbf{z}^{nom}(t) - \mathbf{z}^{i}(t)) \tag{44}$$

By adding the nominal control input $\mathbf{u}^{nom}$ we have the control law as

$$\mathbf{u}_{fb}(\mathbf{u}(t), \mathbf{z}^{nom}(t), \mathbf{z}^{i}(t)) = \mathbf{u}(t) + \mathbf{K}(\mathbf{z}^{nom}(t) - \mathbf{z}^{i}(t)) \tag{45}$$

The nominal models SDE is identical to (33) with the expected value for the parameters $\boldsymbol{\xi} = \boldsymbol{\xi}^{nom}$. For the sampled system models $i \in \{1, \dots N\}$ the updated SDE is formulated as

$$d\mathbf{z}^{i} = \mathbf{f}_{sys}(\mathbf{z}^{i}, \xi^{i})dt + +\mathbf{G}_{sys}(\mathbf{z})\mathbf{u}_{fb}(\mathbf{u}(t), \mathbf{z}^{nom}(t), \mathbf{z}^{i}(t))dt + \mathbf{D}d\boldsymbol{\omega}_{t}^{i} \tag{46}$$

In essence, this extension offers two key benefits:

1. it enables high-frequency online control via the local feedback while the main optimization plan at a lower frequency.
2. it helps **centering** the trajectories of the sampled models along the **nominal** one, yielding more compact, feasible trajectories by optimizing over a class of closed-loop controllers.

More details can be found in [8]. Figure 14 shows the control architecture composed of the offline optimization (dashed line) and the online feedback loop (right blue area). The offline part is divided into simulating the nominal model and simulating the N models with different parameter realizations, $\boldsymbol{\xi}$. For the sampled models the feedback loop is displayed in the blue area.

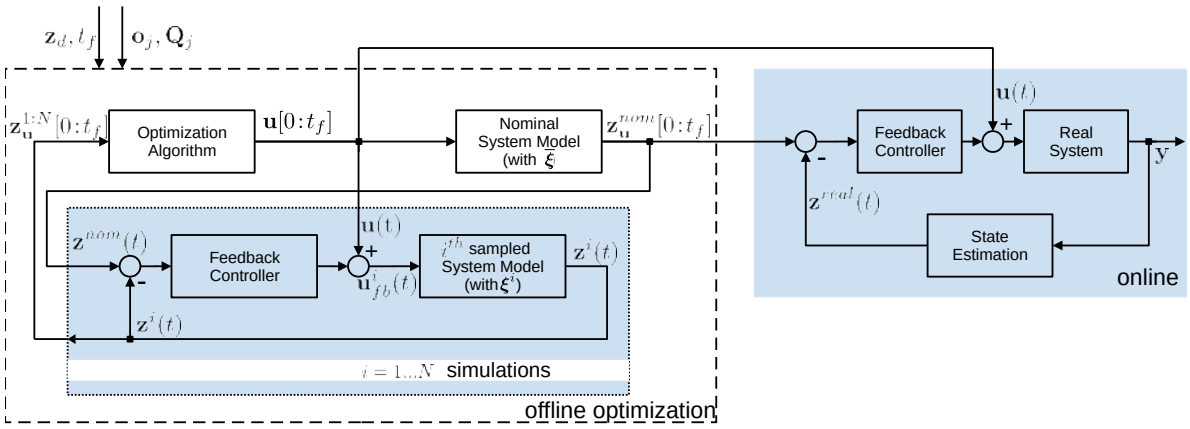


Figure 14: Control Architecture composed of offline planning and online feedback control (adapted from [8]). The feedback loop drives the surrogate sampled system to behvae like the nominal one.

3.2.3 Implementation

We implement a time-discretized version of the optimal control problem (43) with feedback augmentation as depicted in Figure 14. The discretization uses the Euler-Maruyama discretization with time-step Δt for the SDE resulting in

$$\mathbf{z}_{k+1}^{i} = \mathbf{z}_{k}^{i} + \mathbf{f}_{\text{sys}}(\mathbf{z}_{k}^{i}, \boldsymbol{\xi}^{i})\Delta t + \mathbf{G}_{\text{sys}}(\mathbf{z}_{k})\mathbf{u}_{\text{fb}}(\mathbf{u}_{k}, \mathbf{z}_{k}, \mathbf{z}_{k}^{i})\Delta t + \mathbf{D}\boldsymbol{\omega}_{k}^{i}. \tag{47}$$

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

To balance numerical stability and computational tractability, we use a smaller time step Δt for model integration and a larger one for control updates. Specifically, each control input is held constant for $T_\Delta \in \mathbb{N}$ integration steps according to a zero-order-hold scheme:

$$\mathbf{u}_k = \mathbf{u}_j \quad \text{for } k = jT_\Delta, jT_\Delta + 1, \ldots, (j+1)T_\Delta - 1, \tag{48}$$

where $j = 0, 1, \ldots, N_{\text{contr}} - 1$ is the control interval index. This results in N_{contr} control input values to be optimized and $S_T = T_\Delta N_{\text{contr}}$ total simulation steps.

The implementation is carried out in Python using the JAX library [10] for numerical computation. JAX enables efficient execution through just-in-time compilation and automatic vectorization, significantly accelerating the computations. To solve the resulting optimization problem, we employ the off-the-shelf solver IPOPT [11].

3.2.4 Evaluations: Latest Simulation Results

In this section, we aim to validate our model and method through the simulation. Table 8 in the appendix lists some relevant parameters used for simulation, run on an AMD® Ryzen 9 7950x3d 16-core processor.

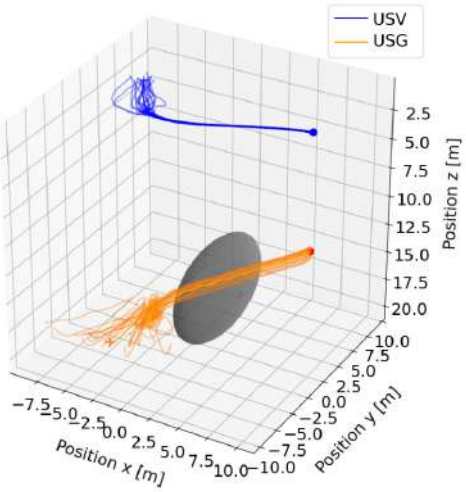


Figure 15: Planned trajectory from starting point "+" to terminal point "•"

Figure 15 shows an example for the planned trajectory avoiding the obstacle. Near the starting point marked by "+" the uncertain and sampled initial position is shown by the different line origins. Also visible in the graphic are the deviating trajectories the model takes for different realizations of the parameters.

Note that, for the proposed implementation and chosen parameters, the optimization problem involves 742 decision variables and a nonlinear dynamic model with 21 states and $120 \times 50 = 6000$ time steps. This configuration poses a significant challenge for the solver, with computation times ranging from approximately 4 h for a small number of sampled models ($N \approx 5$) to around 17 h for larger values ($N \approx 20$). Nevertheless, the proposed method successfully produces feasible trajectories that reach

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

the desired terminal positions while avoiding obstacles. The simulated system behavior accurately captures the underlying pendulum dynamics and demonstrates the benefits of jointly controlling the USV and UUV. Future work may focus on improving computational efficiency, possibly through a different solver strategy or an approximation of the model.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

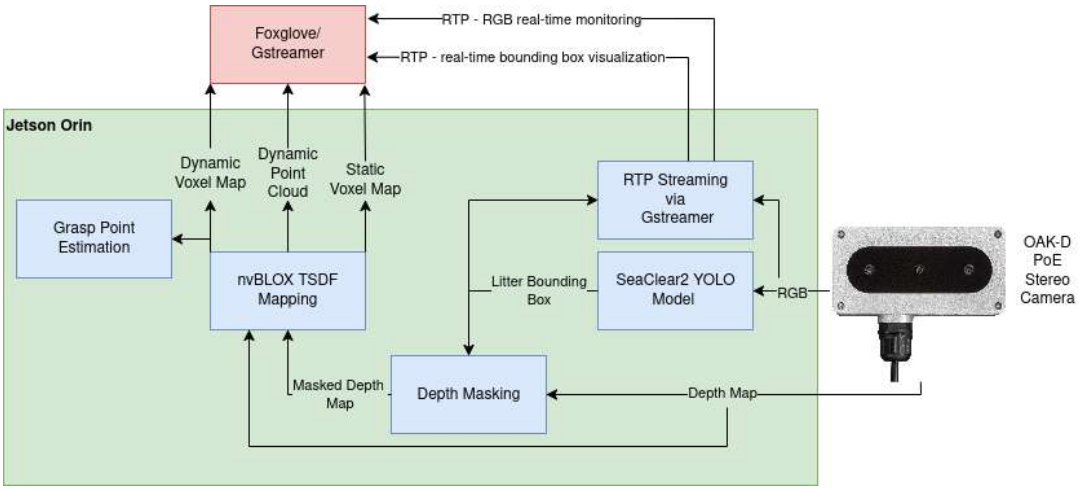


Figure 16: An overview of the perception system. The blue boxes represent the processing nodes. The red box the monitoring node(s). The green background indicates the processes that benefit from edge GPU acceleration.

4 Perception System for Litter Object Localization

The grapple needs to be able to localize litter objects autonomously underwater, infer the relative position of an object to the grapple, and estimate the litter object metric geometry in preparation for grasping. To this end, we present in the following an overview of the perception system design as shown in Figure 16. First, we subscribe to depth and RGB data produced by a passive stereo camera. Secondly, we perform inference with a YOLOv8 object detection deep learning model, trained to detect litter objects underwater. Third, we use the bounding box returned by the object detection inference to create a mask for the depth map produced by the stereo camera, masking the pixel values in the depth mask which don't correspond to the detected object. Fourth, we use the depth map and the masked depth map to construct a voxel map of the scene with static and dynamic elements. Finally, the voxel map, RGB data and object detection model output can be monitored in real-time via Gstreamer and/or Foxglove. The majority of the process, bar the stereo depth estimation computed on-board the stereo camera, is performed on an embedded GPU system, enabling the deployed algorithms to benefit from a high degree of parallelization w.r.t. the computations, ultimately increasing the speed of the pipeline.

4.1 Underwater Object Detection

The SeaClear YOLO based object detection deep learning model for underwater litter objects as described in the Report D4.3, was used as a foundation for the current perception pipeline. The object detection model returns a bounding box around a region in an RGB image at the location of a detected litter object. The bounding box is then 1) laid on top of the actual RGB data and streamed to the surface machine for real-time monitoring, 2) passed to a depth masking node which separates an incoming depth map aligned with the RGB data into "litter object" and "non-litter object" parts. The masked depth mask is then passed to the 3D reconstruction algorithm discussed in Section 4.2.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

4.2 Truncated Signed Distance Function based 3D Reconstruction

Static Voxel Mapping

Compared to point cloud representations, voxel maps approximate each point in 3D space as a voxel, the 3-dimensional equivalent of a pixel. We use the Truncated Signed Distance Function (TSDF) in our approach, which assigns a value corresponding to the (signed) distance of that voxel to the nearest surface. The surface of an object can then be extracted as the location of voxels with distance value 0. This approach provides an estimate for the volumetric properties of objects, informing our grasp planning. Additionally occupancy information of the voxels is stored, such as in occupancy grids, where the probability of a voxel being occupied is stored. Each voxel in space is classified as either *occupied* or *free*, meaning that a particular voxel is either outside of a 3D object if it is *free* and situated within the object if it is *occupied*. This dual representation benefits our functionality two-fold. On the one hand the TSDF based voxel map provides a metric estimate of an objects geometry, allowing us to extract volumetric information informing our grasp planning. The occupancy probabilities can be used for obstacle avoidance. Described below is the computed 3D reconstruction of objects w.r.t. to the camera frame. In order to relate the spatial and volumetric information of detected objects from the camera frame to the robots frame, we apply a transformation consisting of a rotation and a translation from the camera's base frame to the robot's base frame, defined in the transformation tree of the software stack. This allows the spatial information of objects to be used for planning and visual servoing of the grapple system.

Truncated Signed Distance Fields are based on Signed Distance Fields (SDF) which work in the following way. Given a depth map $\mathcal{D}$, the center point of a voxel w.r.t. the voxel grid coordinate frame $\boldsymbol{p_L}$ and the camera pose w.r.t. the voxel map $\boldsymbol{T_{CL}}$, we can sample the depth information from the depth map by projecting $\boldsymbol{p_L}$ onto the depth map and assigning the depth value at that point to the corresponding voxel [12]

$$d = \mathcal{D}\left[\pi_{camera}\left(\boldsymbol{T}_{CL}\boldsymbol{p_L}\right)\right] \tag{49}$$

where $\mathcal{D}\left[\boldsymbol{u}\right]$ represents sampling from the depth map at point $\boldsymbol{u}$. The camera projection function $\pi_{camera}\left(\boldsymbol{p_C}\right)$ for a arbitrary point in 3D space $\boldsymbol{p_C}$ is defined as

$$\pi_{camera}(\boldsymbol{p}_C) = \frac{1}{p_{C,z}} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \end{bmatrix} \begin{bmatrix} p_{C,x} \\ p_{C,y} \\ 1 \end{bmatrix} \tag{50}$$

where f_x, f_y, c_x and c_y are the pinhole camera intrinsic parameters which we receive from camera calibration. Now that we have the depth value sampled at the location of the voxel center projected onto the depth map, we subtract the distance of the voxel center to the camera in order to receive the signed distance field value

$$sdf_i(\boldsymbol{p}_L) = \mathcal{D}\left[\pi_{camera}\left(\boldsymbol{T}_{CL}\boldsymbol{p_L}\right)\right] - cam_z(\boldsymbol{p}_L). \tag{51}$$

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

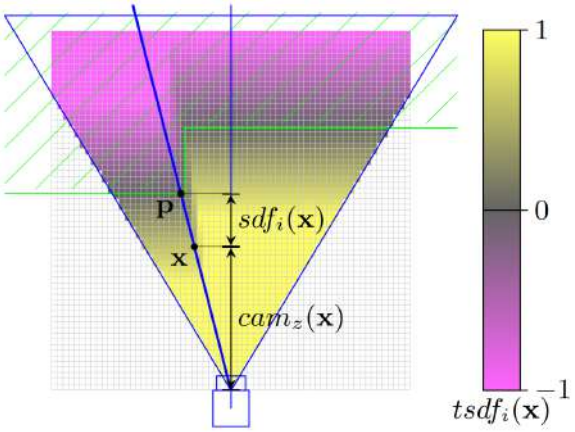


Figure 17: The 3D object in space is denoted in green. The truncated distance values are denoted in color from pink to yellow. In this case an exemplary truncation distance ϵ is set to 1 [13].

The result $sdf_i(\boldsymbol{p}_L)$ is the signed distance field value for the voxel. Repeating this procedure for each pixel in-view, i.e. each voxel which can be projected onto the depth map at a given timestep, we can classify each voxel in-view as occupied or unoccupied, extracting the surface of an object as the zero-crossing between occupied and unoccupied values. TSDF is sufficient for surface reconstruction, where all voxel outside of a truncation distance ϵ are not assigned a value.

For TSDF, the sdf_i are only considered for voxels within small truncation distance ϵ away from the surface of objects. The truncated version of the sdf_i for the i-th observation of a scene is defined as $tsdf_i$ [13]:

$$tsdf_i(\boldsymbol{p}_L) = \max\left(-1, \min\left(1, \frac{sdf_i(\boldsymbol{p}_L)}{\epsilon}\right)\right) \tag{52}$$

An illustration of how signed distance fields work can be seen in Figure 17, here computing the distance of voxel $\boldsymbol{x}$ to the surface point $\boldsymbol{p}$ is illustrated. The colored regions in Figure 17 indicate the values within the truncation threshold.

Voxel grids can be efficiently implemented for GPU acceleration. The grid is divided into larger chunks of voxels, e.g. an 8x8x8 block, and each chunk is stored within GPU memory; each block is then continuously updated in parallel by the GPU, freeing up voxels that are no longer occupied, assigning depth values to newly observed voxels and updating distances as the camera moves through the scene.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

Dynamic Voxel Mapping



Figure 18: The bounding box returned by the object detection model post-processed and laid on top of the RGB data from the downward facing stereo camera.



Figure 19: The static and dynamic voxel layers computed from the input scene in Figure 18. The red voxels correspond to the litter object detected by the model.

So far we have only discussed mapping of a static scene by the TSDF algorithm. Mapping the whole scene is computationally expensive even in the TSDF case. Since we receive a bounding from from the object detection model, we use this to reduce the region of interest for applying TSDF. The bounding box received from the object detection applied to the RGB data as discussed in Section 4.1, is used to mask the depth map from the stereo camera into litter object parts and non-litter object parts. The voxel map is then separated into a static and a dynamic layer, where the static layer is computed as before, but the dynamic layer now only contains voxels computed from the masked depth values, i.e. contains voxels that correspond to a detected object. We can then apply grasp point estimation to the dynamic layer corresponding to the detected object.

We relax the assumption that only static objects are reconstructed by the algorithm from Section 4.2, since this allows the bounding box to move within the scene if e.g. the object is moved by ocean dynamics. For this, we need to apply a decay step to the occupancy values of the voxels. At a fixed frequency, all voxel occupancy probabilities are decayed towards 0.5 over time. This means that the state of a voxel (occupied or free) becomes less certain after it has fallen out of view until it becomes unknown i.e. the voxel holds 0.5 occupancy probability.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

5 Smart Grapple Grasping Approach

This section details the grasping methodology developed for the Smart Grapple to reliably capture underwater litter of varying sizes and weights. The approach integrates a torque-monitor-based adaptive grasping algorithm that employs a three-state machine architecture (closing, loading, and finalization phases) to ensure stable grasp execution through continuous torque feedback. Building upon the kinematic foundations, the section further presents an optimal grasping posture analysis that determines finger configurations maximizing gripping force under joint position and torque constraints, addressing three operational scenarios with varying levels of perception information. Finally, a gripping points optimization service is described that leverages 3D point cloud data to identify optimal contact surfaces for the four-finger configuration.

5.1 Torque Monitor Grasping Algorithm

The torque-monitor-based grasping algorithm, which is the module in "grasping" block in Figure 11, is designed to enable adaptive and reliable grasp execution by continuously monitoring the joint torques of the smart grapple during the closing process. The function follows a state-machine architecture that ensures the grasping motion transitions smoothly across distinct control phases, each governed by torque feedback and kinematic state information. This design enhances grasp stability while preventing excessive force that could damage the object or the gripper mechanism.

As shown in Figure 20, the grasping process begins with the initialization of the system state and movement of the smart grapple to a predefined start configuration. Once positioned, the state machine is activated to coordinate the grasp sequence. The algorithm proceeds through three principal states: State 1 (Closing Phase), State 2 (Loading Phase), and State 3 (Finalization Phase). Each state defines a specific control behavior based on torque feedback and configuration updates.

1. State 1 – Closing Phase:
   The smart grapple starts from an open position and begins closing toward the object. The current joint configuration is recorded, and a corresponding closing command is generated and stored in the command buffer. This command drives the fingers toward the target until the torque values begin to rise, indicating initial contact with the object. Once the torque measured on any joint exceeds the predefined threshold, the algorithm transitions to State 2. If no contact is detected during the closing process, the smart grapple continues to close until reaching its fully closed configuration, after which the state machine switches to State 3.

2. State 2 - Loading Phase:
   The algorithm continuously monitors the torque signals to determine whether the object has been securely grasped. If the measured torque exceeds the predefined threshold, indicating that the object is firmly held with sufficient contact force, the current configuration is maintained to stably hold the object. The state machine remains in this phase until an open command is received. Conversely, if the detected torque falls below the threshold—suggesting that the contact is insufficient—the system reverts to State 1 to resume the closing process and reattempt the grasp.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

3. State 3 - Finalization Phase:
   In the final stage, the fully closed configuration is assigned to the command buffer to ensure completion of the grasp. The algorithm then flags the state machine as finished, marking the termination of the grasp cycle. The final joint configuration is maintained to preserve a stable grasp until further instructions are issued.

In each state, the joint command is computed based on torque feedback and kinematic information, and then stored in the command buffer. The command buffer is transmitted to the motor driver at each sampling iteration, ensuring real-time execution of the grasping control loop.

5.2 Grapple Kinematics

This subsection presents the kinematic modeling of the Smart Grapple finger mechanism, which is essential for motion planning and grasp execution. Each finger is modeled as a planar two-link manipulator operating in the xy-plane, with the kinematic analysis providing both forward kinematics to determine fingertip positions from joint angles and inverse kinematics to compute required joint configurations for target positions. Two solution approaches are detailed: an analytical closed-form solution derived from geometric relationships, and an iterative solution based on the Forward And Backward Reaching Inverse Kinematics (FABRIK) algorithm, which offers computational efficiency and natural handling of joint constraints in dynamic underwater environments.

5.2.1 Analytic Solution

The kinematic configuration of the smart grapple system is illustrated in Figure 21. As shown, the system can be divided into two main components:

- **Basement (blue section):** the fixed part of the smart grapple that houses essential modules such as the thrusters, electronic units, and structural supports.
- **Finger mechanism (gray links):** the movable portion of the smart grapple responsible for grasping, composed of a series of rigid links driven by rotational joints.

Each finger of the smart grapple is modeled as a planar two-link manipulator that operates in the xy-plane with its base attached to the basement as shown in Figure 21. The two revolute joints allow the finger to reach and conform to objects of varying sizes during grasping. The base joint of the right finger is located at coordinates (x_0, y_0), and the two finger links have lengths l_1 and l_2, respectively. The joint angles θ_1 and θ_2 are defined as shown in the figure: θ_1 is measured from the negative x-axis of the basement frame to the first link, and θ_2 represents the relative rotation between the first and second links. The fingertip position, denoted as (T_x, T_y), corresponds to the end-effector of the two-link chain. Thus, the kinematic model of each finger is analogous to that of a 2-DOF serial robotic arm.

The forward kinematics of a single finger can be formulated as:

$$T_x = x_0 - l_1 \cos(\theta_1) - l_2 \cos(\theta_1 + \theta_2) \tag{53}$$

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

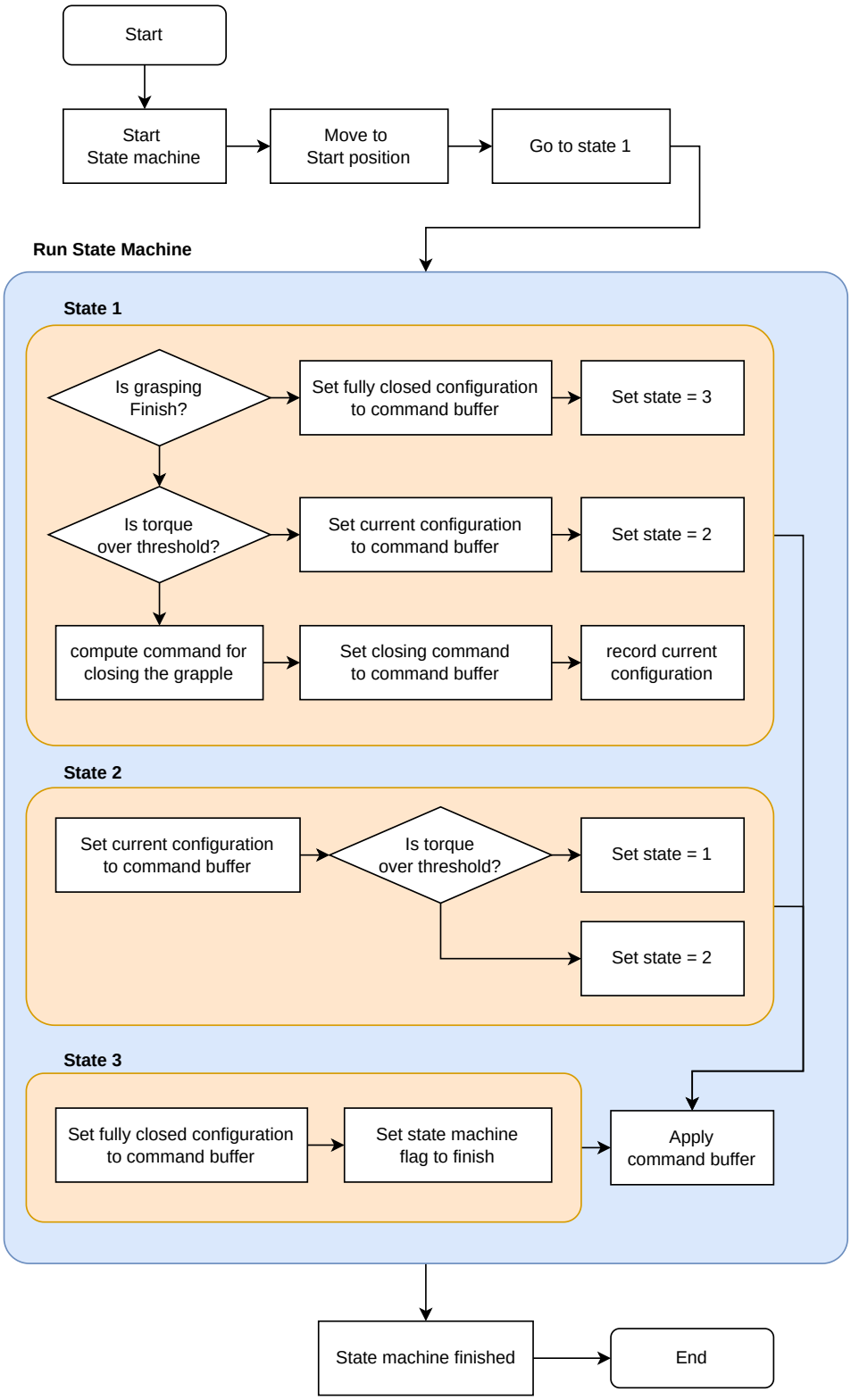


Figure 20: The pipeline of the torque monitor grasping algorithm.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

$$T_y = y_0 - l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) \tag{54}$$

where (x_0, y_0) represents the base position of the finger, and (T_x, T_y) denotes the fingertip (end-effector) position. These equations describe the geometric relationship between the joint angles and the Cartesian position of the fingertip. Given specific joint configurations, the position of the fingertip relative to the base can be determined directly through these equations.

Conversely, the inverse kinematics problem aims to determine the joint angles corresponding to a desired fingertip position. For the two-link finger mechanism, the analytical inverse kinematics can be derived as:

$$\theta_1 = \arctan(\frac{T_y - y_0}{T_x - x_0}) - \arcsin(\frac{l_2 \sin(\theta_2)}{\sqrt{(T_x - x_0)^2 + (T_y - y_0)^2}}) - \pi \tag{55}$$

$$\theta_2 = \arccos(\frac{(T_x - x_0)^2 + (T_y - y_0)^2 - l_1^2 - l_2^2}{2 l_1 l_2}) \tag{56}$$

These equations enable the computation of the required joint angles given a target fingertip position. In practical applications, this inverse kinematics formulation is used to generate motion commands for grasping tasks. Once the target position of an object (e.g., marine litter) is identified, the inverse kinematics solution provides the corresponding motor commands to position each finger appropriately and achieve a stable grasp.

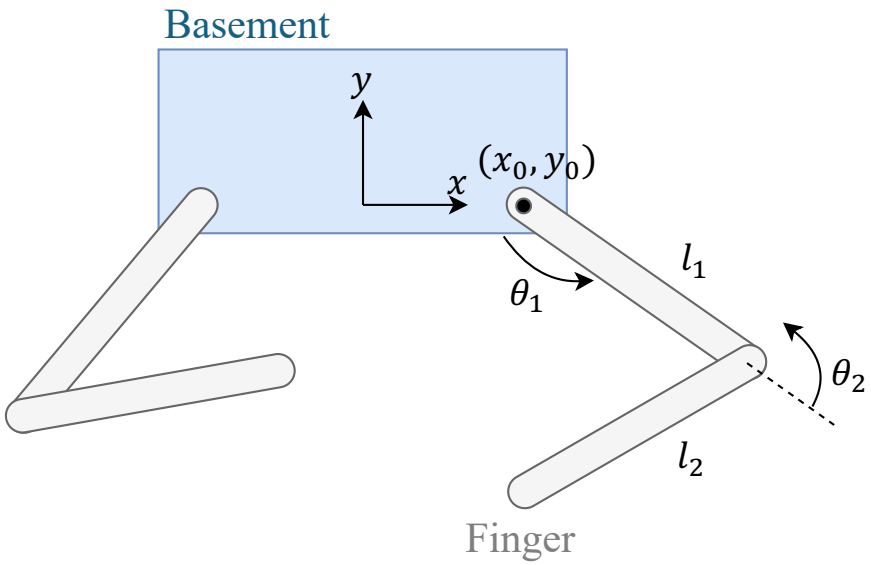


Figure 21: The kinematics of the smart grapple system.

The kinematic parameters of the smart grapple system are summarized in Table 7 in the appendix. These parameters define the physical dimensions and reference configuration of each finger, which serve as the foundation for the motion planning and grasp control algorithms described in subsequent sections.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

5.2.2 Iteration Solution

To address the inverse kinematics problem of a multi-link manipulator, the FABRIK (Forward And Backward Reaching Inverse Kinematics) algorithm [14] provides an efficient and intuitive iterative solution. Unlike traditional Jacobian-based or trigonometric approaches, FABRIK determines joint positions geometrically by locating each joint along a line connecting its neighboring joints. This geometric interpretation eliminates the need for matrix inversion or complex angle computations, resulting in a low computational cost and fast convergence, typically within only a few iterations.

The algorithm operates through a two-phase iterative process: a forward reaching step and a backward reaching step. In the forward phase, the computation begins from the end effector and sequentially updates each preceding joint position while maintaining the link length constraints, gradually bringing the chain closer to the target. In the backward phase, the process starts from the base joint and propagates the corrections forward again, ensuring that both the end effector and the base maintain their respective positions. Each full forward–backward iteration minimizes the positional error between the end effector and the desired target in Cartesian space.

Assume $\mathbf{p}_1, \ldots, \mathbf{p}_n$ are the joint positions of a manipulator. Also, assume that $\mathbf{p}_1$ is the root joint and $\mathbf{p}_n$ is the end effector, for the simple case where only a single end effector exists. The target is symbolised as t and the initial base position by b. FABRIK is illustrated in pseudo-code in Algorithm 1 and a graphical representation of a full iteration with a single target and 4 joints is presented and explained in Figure 22.

FABRIK is particularly suitable for robotic applications that require fast and stable inverse kinematics computation. It naturally accommodates joint and positional constraints and can be extended to handle multiple kinematic chains with multiple end effectors. Owing to its numerical stability, low computational demand, and realistic motion generation, the FABRIK algorithm is an effective choice for solving the inverse kinematics problem of the smart grapple finger system.

Since the underwater environment is inherently complex and dynamic, the target object may experience motion due to external disturbances. Therefore, the robustness of the FABRIK algorithm is crucial. To evaluate its performance within our system, we simulate several scenarios in which the target positions are shifted to mimic the influence of environmental factors, requiring the manipulator fingers to continuously track the actual target location. As illustrated in Figure 23, the robot's arms successfully follow the moving object in real time. When the object moves beyond the manipulator's reachable workspace under the current constraints, the system automatically issues a warning, indicating that the robot must reposition itself to re-establish reachability and resume grasping.

5.3 Optimal Grasping Manipulability Under Joint Position and Joint Torque Constraints

To ensure that the smart grapple can securely lift underwater litter, it must generate sufficient force to counteract the object's weight while maintaining a reliable frictional interface. Consider a litter object of mass M, with friction coefficient μ, gravitational acceleration g, and end-effector force components (F_x, F_y). The static equilibrium of the object requires

$$Mg = \mu F_x + F_y, \tag{57}$$

where μ corresponds to the surface properties of different debris types, summarized in Table 3.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

Algorithm 1 A full iteration of the FABRIK algorithm [14]

Input: The joint positions $\mathbf{p}_i$ for $i = 1, \ldots, n$, the target position $\mathbf{t}$, and the distances between each joint $d_i = \|\mathbf{p}_{i+1} - \mathbf{p}_i\|$ for $i = 1, \ldots, n-1$.
Output: The new joint positions $\mathbf{p}_i$ for $i = 1, \ldots, n$.

% The distance between root and target
dist $= \|\mathbf{p}_1 - \mathbf{t}\|$
% Check whether the target is within reach
if dist $> d_1 + d_2 + \ldots + d_{n-1}$ **then**
    % The target is unreachable
    for $i = 1, \ldots, n-1$ **do**
        % Find the distance r_i between the target $\mathbf{t}$ and the joint $\mathbf{p}_i$
        $r_i = \|\mathbf{t} - \mathbf{p}_i\|$, $\lambda_i = d_i / r_i$
        % Find the new joint positions $\mathbf{p}_i$
        $\mathbf{p}_{i+1} = (1 - \lambda_i)\mathbf{p}_i + \lambda_i \mathbf{t}$
    end for
else
    % The target is reachable; thus, set as $\mathbf{b}$ the initial position of the joint $\mathbf{p}_1$
    $\mathbf{b} = \mathbf{p}_1$
    % Check whether the distance between the end effector $\mathbf{p}_n$ and the target $\mathbf{t}$ is greater than a tolerance
    difA $= \|\mathbf{p}_n - \mathbf{t}\|$
    while difA $>$ tol **do**
        % **Stage 1: Forward reaching**
        % Set the end effector $\mathbf{p}_n$ as the target $\mathbf{t}$
        $\mathbf{p}_n = \mathbf{t}$
        for $i = n-1, \ldots, 1$ **do**
            % Find the distance r_i between $\mathbf{p}_{i+1}$ and $\mathbf{p}_i$
            $r_i = \|\mathbf{p}_{i+1} - \mathbf{p}_i\|$, $\lambda_i = d_i / r_i$
            % Find the new joint positions $\mathbf{p}_i$
            $\mathbf{p}_i = (1 - \lambda_i)\mathbf{p}_{i+1} + \lambda_i \mathbf{p}_i$
        end for
        % **Stage 2: Backward reaching**
        % Set the root $\mathbf{p}_1$ to its initial position
        $\mathbf{p}_1 = \mathbf{b}$
        for $i = 1, \ldots, n-1$ **do**
            % Find the distance r_i between $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$
            $r_i = \|\mathbf{p}_{i+1} - \mathbf{p}_i\|$, $\lambda_i = d_i / r_i$
            % Find the new joint positions $\mathbf{p}_{i+1}$
            $\mathbf{p}_{i+1} = (1 - \lambda_i)\mathbf{p}_i + \lambda_i \mathbf{p}_{i+1}$
        end for
        difA $= \|\mathbf{p}_n - \mathbf{t}\|$
    end while
end if

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

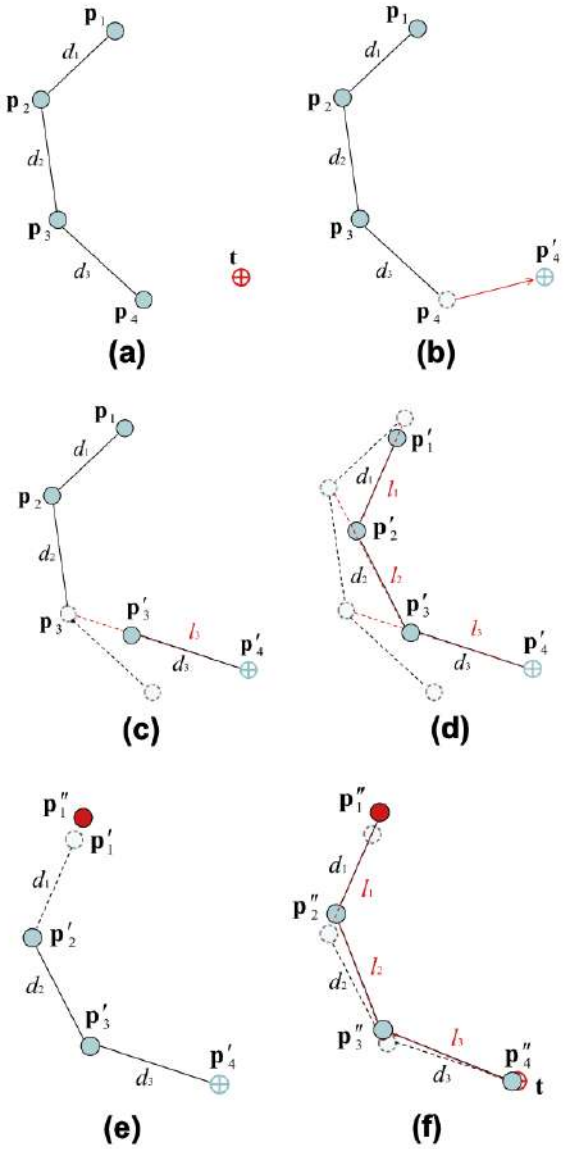


Figure 22: [14] Illustration of one full iteration of the FABRIK algorithm for a manipulator with four joints and a single target. (a) Initial configuration of the manipulator and target; (b) the end-effector $\mathbf{p}_4$ is moved to the target position; (c) the new joint position $\mathbf{p}'_3$ is computed along the line l_3 connecting $\mathbf{p}'_4$ and $\mathbf{p}_3$, maintaining the distance d_3 from $\mathbf{p}'_4$; (d) the same procedure is applied sequentially to the remaining joints; (e) in the backward stage, the root joint $\mathbf{p}'_1$ is moved back to its original position; (f) the process is repeated from the base toward the end-effector. The iteration continues until the end-effector reaches or sufficiently approaches the target.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

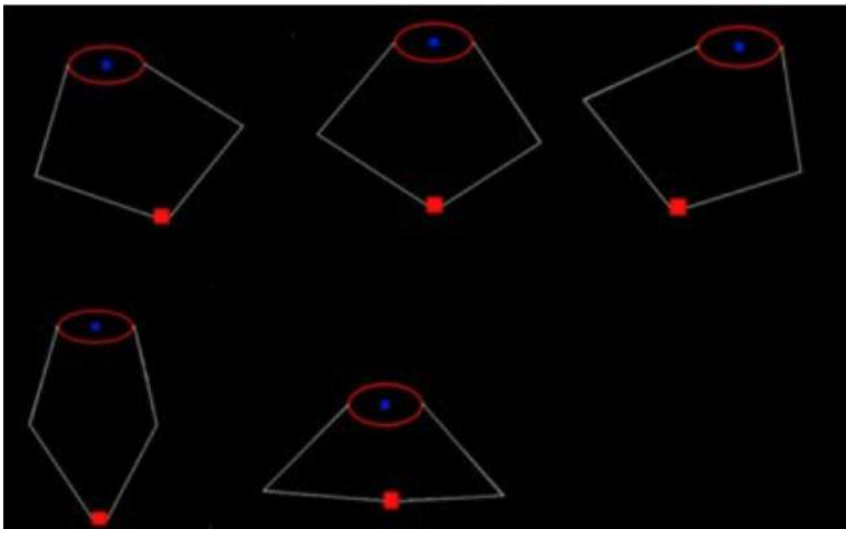


Figure 23: FABRIK reacting to target movement. The red square symbolizes the robot's body. Blue square symbolizes the target base point. The red ellipse visualizes the target's shape. Gray lines visualize the robot arms.

Table 3: Friction coefficients of common litter objects.

Object Type	Friction Coefficient
Barrel	0.60
E-scooter	0.60
Steel pipe	0.60
Tractor tire	0.86
Car tire	0.86

5.3.1 Grasping Scenarios

In practice, the available perception information varies depending on environmental conditions. Therefore, three grasping scenarios are considered, as illustrated in Figure 24. These scenarios differ in flexibility, force performance, and dependence on vertical control.

- **Case 1: Full Cartesian Control.** The desired end-effector position is directly commanded through the joint space. This offers maximum grasping flexibility but does not necessarily maximize force output.
- **Case 2: Force-Optimized Grasping.** The vertical grasping position is optimized with respect to object width to maximize the attainable horizontal force. This enables the system to exploit the full manipulability of the finger mechanism.
- **Case 3: Fixed Optimal Height for Multi-Width Objects.** A single, globally optimal vertical height is selected, enabling robust grasping performance across various object widths while simplifying

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

underwater motion planning.

In practice, since we need to compromise the flexibility and force performance, we consider the case 3 in the sea trial.



Figure 24: Comparison of grasping flexibility and force performance across the three scenarios.

5.3.2 Static Force Analysis

To describe the static relationship between joint torques and end-effector forces, we apply the principle of virtual work. Let W denote virtual work and P denote power. For the two-joint finger mechanism, the work and power on the joint side are

$$W = \tau_1 \, \delta\theta_1 + \tau_2 \, \delta\theta_2, \tag{58}$$

$$P = \tau_1 \dot{\theta}_1 + \tau_2 \dot{\theta}_2 = \dot{\boldsymbol{\theta}}^T \boldsymbol{\tau}. \tag{59}$$

On the end-effector side, the corresponding expressions are

$$W = \delta x \cdot F, \tag{60}$$

$$P = v \cdot F = (J(\boldsymbol{\theta})\dot{\boldsymbol{\theta}})^T F = \dot{\boldsymbol{\theta}}^T J(\boldsymbol{\theta})^T F, \tag{61}$$

where $J(\boldsymbol{\theta})$ is the Jacobian matrix of the finger, given by

$$J(\boldsymbol{\theta}) = \begin{bmatrix} l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) & l_2 \sin(\theta_1 + \theta_2) \\ -l_1 \cos\theta_1 - l_2 \cos(\theta_1 + \theta_2) & -l_2 \cos(\theta_1 + \theta_2) \end{bmatrix}. \tag{62}$$

Equating the power expressions in (59) and (61), we obtain the fundamental static relationship:

$$\boldsymbol{\tau} = J(\boldsymbol{\theta})^T F. \tag{63}$$

Provided that $J(\boldsymbol{\theta})$ is invertible, the end-effector force can be computed from the joint torques as

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = (J(\boldsymbol{\theta})^T)^{-1} \boldsymbol{\tau}. \tag{64}$$

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

Explicitly, this yields

$$F_x = \frac{\cos\theta_1}{l_2 \sin\theta_2}\boldsymbol{\tau}, \tag{65}$$

$$F_y = \frac{\sin\theta_1}{l_2 \sin\theta_2}\boldsymbol{\tau}. \tag{66}$$

A singular configuration arises when $\det(J(\boldsymbol{\theta})) = 0$, resulting in loss of controllability and unbounded joint torques. From (62), the determinant is

$$\det(J(\boldsymbol{\theta})) = l_1 l_2 \sin(\theta_2). \tag{67}$$

Thus, singularity occurs when $\theta_2 = 0$ or $\theta_2 = \pi$. To avoid these conditions, we construct a 5 cm band around the singular boundary, defining the *safe workspace* shown in Figure 25.

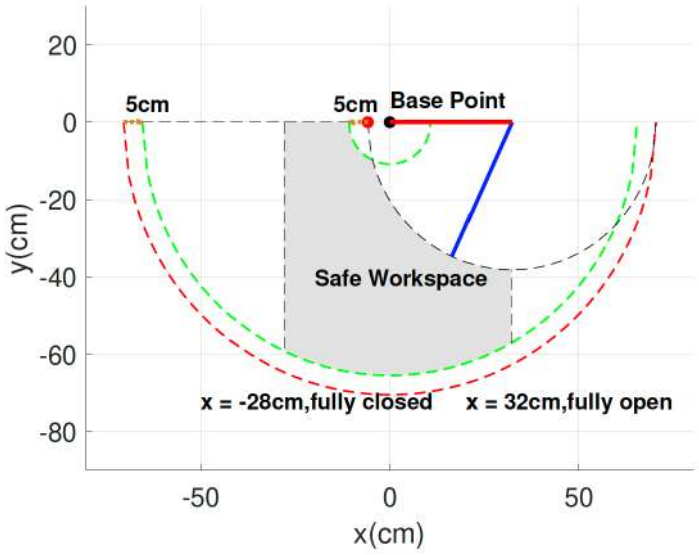


Figure 25: The safe workspace for a single finger after excluding a 5 cm neighborhood of singular configurations.

5.3.3 Grasping Posture in Different Scenarios

To determine an appropriate grasping posture across different operational scenarios, we develop a unified framework for computing the joint commands associated with each case. In all scenarios, we assume that the *horizontal* target position (i.e., the object width) is obtained from the perception system. Depending on the level of available information, we consider three distinct cases.

Case 1: Full Target Position Known. In this case, both the horizontal and vertical target positions of the end effector are given. Thus, the complete Cartesian target (x^*, y^*) is available. The corresponding joint configuration can be computed using either the analytical inverse kinematics method introduced in Section 5.2.1 or the iterative FABRIK method described in Section 5.2.2. Once the joint angles are

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

obtained, the grasping forces can be evaluated using (65) and (66). The resulting force distribution within the safe workspace is illustrated in Figure 26.

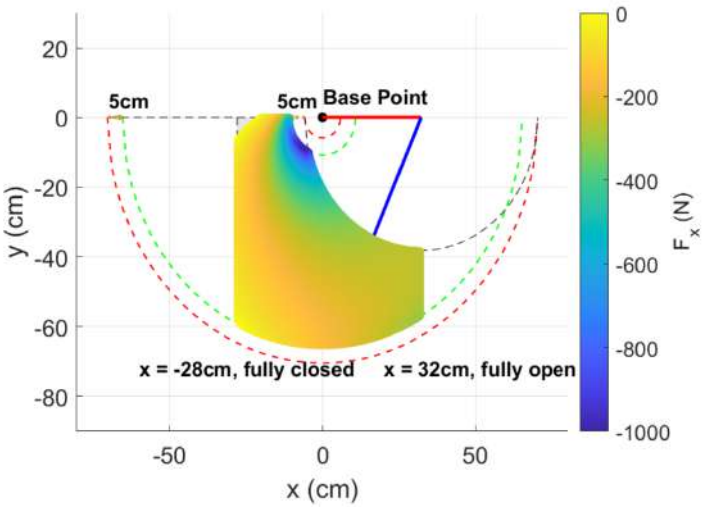


Figure 26: The force distribution in the safe workspace.

Case 2: Horizontal Position Known, Vertical Position to be Optimized. When only the horizontal target position is known, i.e., $x^* \in \mathcal{X}$ where $\mathcal{X}$ denotes the admissible horizontal workspace, the vertical position must be selected so as to maximize the achievable horizontal gripping force. Formally, the optimal vertical position is obtained by

$$y^* = \arg\max_{y \in \mathcal{Y}} F_x(x^*, y), \tag{68}$$

where $\mathcal{Y}$ is the feasible vertical workspace. The value y^* is computed using an optimization toolbox, after which the associated joint configuration can be determined via inverse kinematics. By evaluating the optimizer over all feasible object widths, we obtain a mapping from x^* to the corresponding optimal vertical position, which is illustrated in Figure 27, and the joint configuration that maximizes the horizontal force within the safe workspace.

Case 3: Fixed Grasping Position with Good Average Performance. Although Case 2 yields the force-maximizing posture for each object width, the resulting optimal vertical position y^* varies with the target. Consequently, the smart grapple would need to continuously adjust its vertical position, which is challenging in underwater environments due to hydrodynamic disturbances and limited maneuverability. To avoid frequent repositioning, we instead seek a *single* vertical position that provides consistently good average performance across different object widths.

To achieve this, let (x_j, y_i)

$$\bar{F}_x^i = \frac{1}{|\mathcal{X}|} \sum_j F_x^j(y_i), \qquad \sigma_i^2 = \frac{1}{|\mathcal{X}|} \sum_j \left(F_x^j(y_i) - \bar{F}_x^i\right)^2.$$

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

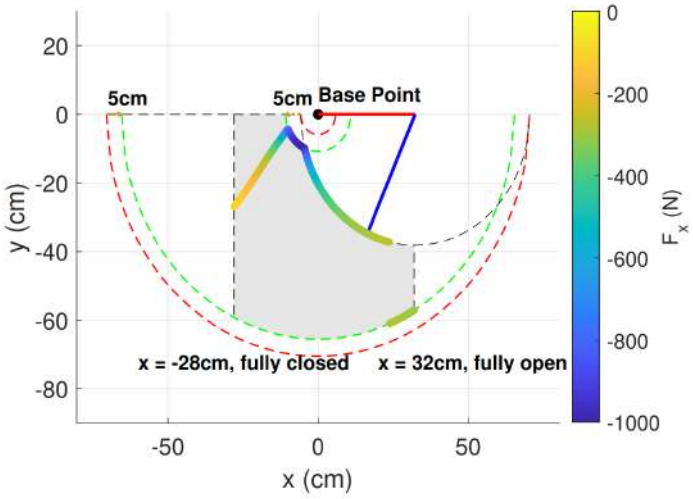


Figure 27: The optimized vertical position trajectory.

The optimal vertical posture is obtained by balancing high average performance and low variability, expressed as

$$y^* = \underset{y_i \in \mathcal{Y}}{\arg\max}\ \Phi(\,\bar{F}_x^i,\,\sigma_i\,), \tag{69}$$

where $\Phi(\cdot)$ denotes a performance metric that rewards larger average force while penalizing high variance. This yields a robust grasping posture that performs well across a range of object widths without requiring vertical repositioning of the smart grapple.

5.4 Gripping points optimization

As presented in Subsection 5.3, there might be several possible grasping configurations for a given object, some of them being optimal under selected criteria. However, these criteria cannot be complete, unless the shape and size of the object is also taken into consideration. The object detection component presented in Section 4.2 is providing a point cloud of the litter that needs to be picked. As a next step to the gripping pipeline, we have developed a ROS2 service, that consumes this point cloud and is then reconstructing a surface mesh based on it. Surfaces that are not suitable for gripping (e.g. horizontal surfaces) are then filtered out. An optimization routine is checking all possible pairs of opposite gripping surfaces (for two fingers), and for each pair it checks all kinematically possible surfaces for the other two fingers. For each quadruplet of surfaces, it calculates the forces that the grapple can exert on them, and eventually picks the quadruplet that can be grasped with maximal forces. This ensures that the grapple will pick the object in the most favourable configuration, therefore increasing the chances of a successful litter picking. The procedure for the gripping points optimization is presented graphically in Figure 28. This gripping points optimization was done in the SeaGrip project, supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS/CCCDI -UEFISCDI, project number PN-IV-P8-8.1-PRE-HE-ORG-2023-0012, within PNCDI IV. The project results were later integrated in the SeaClear2.0 grasping pipeline.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

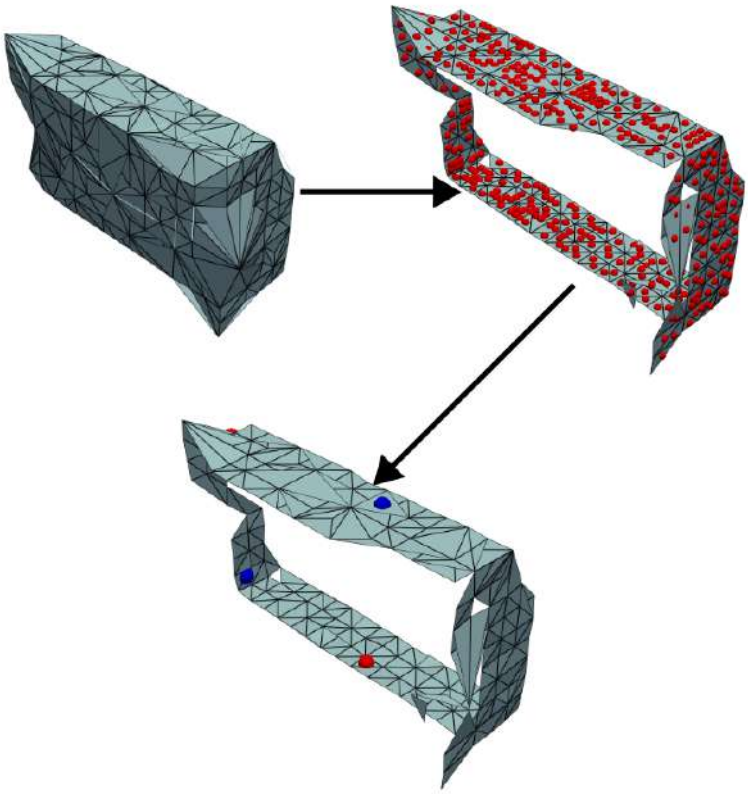


Figure 28: Steps of the gripping points service. We first reconstruct a mesh based on the point cloud provided by the object detection algorithm. The surfaces appropriate for grasping are then isolated, and the service is calculating the optimal surfaces for gripping based on force analysis of the grapple.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

6 Digital Twin of the Smart Grapple

Due to the scarce availability of sea trials, the USG is mostly tested in the lab. Additionally, a simulation-based, digital twin approach is used to validate the system between field tests.

Simulation is conducted using the MArine Robotics Unity Simulator (MARUS) tool. The source code of the USG is stored in a separate Docker container from the MARUS interface. This Software in the Loop (SIL) method helps to bridge the sim-to-real gap by allowing the same source code to run in the real system with minimal changes. It also accelerates development since no extra code is needed for simulation.

Scope of the Simulation. The digital twin is currently able to reproduce navigation of the USG. That means the perception, planning, and tracking sub-modules are already integrated in the simulation pipeline. Figure 29 visualizes the software stack for simulation and Figure 30 shows the digital twin in the virtual environment.

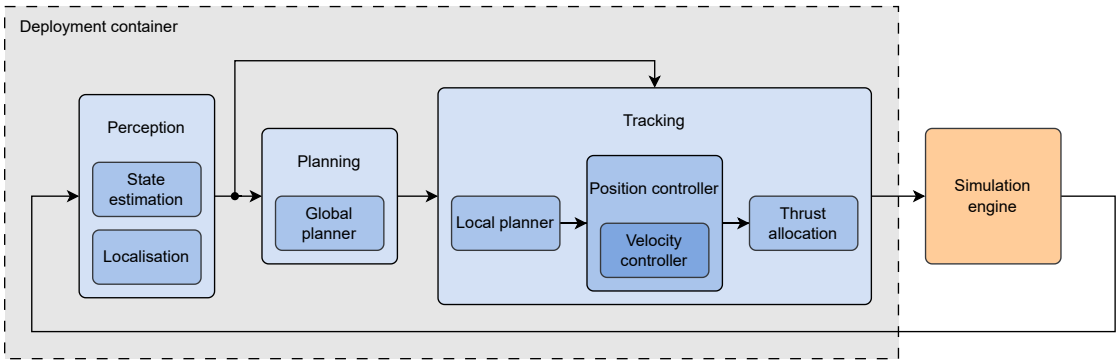


Figure 29: Software stack of the simulation pipeline

MARUS Engine. The MARUS simulation engine was developed by the *Laboratory for Underwater Systems and Technologies* at the University of Zagreb [15]. It is an open-source simulator for marine robotics based on the Unity3D software. The main benefits of this simulator are:

- Native ROS 2 support,
- Realistic underwater visual data (needed for machine learning computer vision algorithms),
- Availability of standard sensors in marine robotics, such as DVL, depth sensor, sonar, and acoustic communication,
- Mesh-based approximation of hydrodynamics,
- Ongoing development.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union

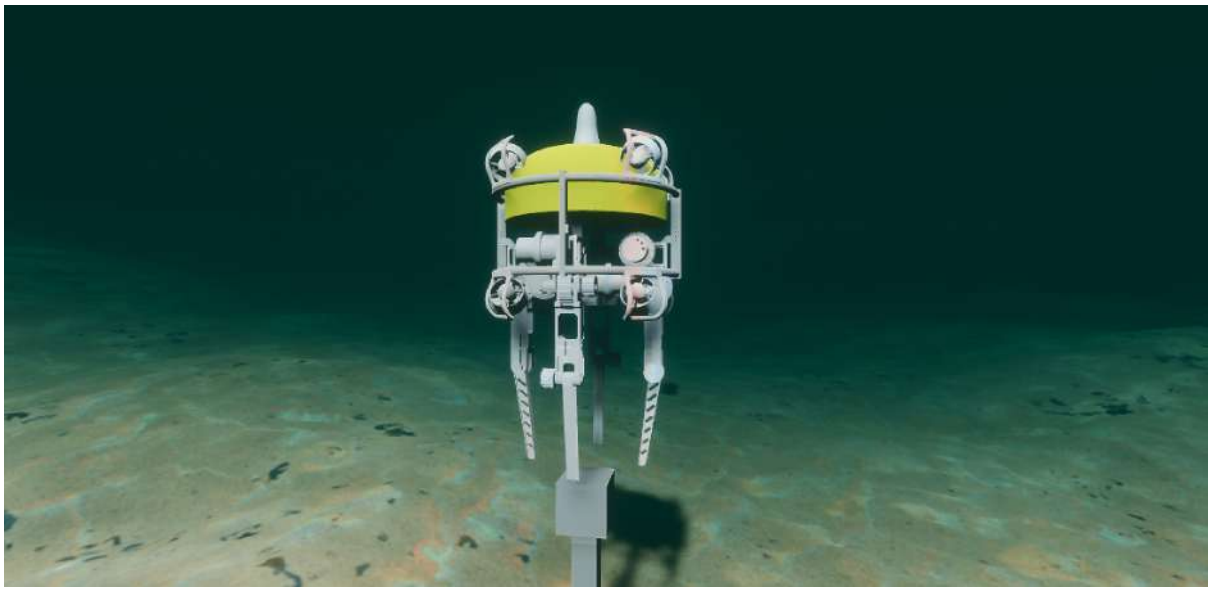


Figure 30: Digital twin in the virtual environment

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

7 Experiments

This section presents the experimental validation of the Smart Grapple system through both controlled laboratory tests and field demonstrations. Laboratory experiments systematically evaluated the grapple's load-bearing capacity by progressively grasping objects up to 251 kg, recording joint positions and torques at each weight increment to verify that the system meets the 250 kg design requirement. Field results from the Marseille demonstration in September 2025 provide real-world validation, documenting successful retrieval of various marine litter objects including a car seat, tire, fences, and an attempted grasp of a shipwreck fragment, with detailed analysis of sensor outputs during representative grasping maneuvers.

7.1 Laboratory tests

To validate that the grapple can securely grasp objects up to 250 kg, a series of experiments was conducted in which weights were progressively added to a test container until the total load exceeded 250 kg. For each trial, the grapple was lifted up with the container vertically, replicating the operational conditions encountered during underwater litter retrieval. The stability of the grasp was evaluated by verifying whether the object remained securely held throughout the lifting motion. All tested weights were successfully grasped, and the results are summarized in Table 4. These experiments demonstrate that the system is capable of reliably handling objects ranging up to 250 kg, as shown in Figure 31.

Table 4: Joint position and torque values for different object weights.

Weight [kg]	Joint Position [rad]	Joint Torque [Nm]
20	(2.72, -1.71, 2.47, -2.28, 2.77, -1.79, 2.47, -2.14)	(5.82, 2.70, 2.88, 18.53, 2.15, 0.73, 7.35, 5.27)
40	(2.77, -1.79, 2.32, -2.16, 2.77, -1.79, 2.57, -2.04)	(3.14, 5.81, 5.97, 2.68, 3.96, 5.39, 3.76, 3.73)
60	(2.77, -1.79, 2.32, -2.16, 2.77, -1.79, 2.57, -2.04)	(3.07, 5.81, 6.00, 2.65, 4.00, 5.39, 3.76, 3.73)
80	(2.77, -1.79, 2.32, -2.16, 2.77, -1.79, 2.57, -2.04)	(3.06, 5.81, 6.02, 3.63, 4.02, 5.87, 3.31, 4.60)
100	(2.72, -1.81, 2.47, -2.28, 2.77, -1.89, 2.47, -2.24)	(14.18, 27.89, 3.27, 14.61, 2.88, 23.92, 2.62, 15.93)
120	(2.72, -1.81, 2.47, -2.28, 2.77, -1.89, 2.47, -2.24)	(13.69, 27.71, 3.27, 14.66, 2.49, 23.23, 2.41, 16.02)
140	(2.72, -1.81, 2.47, -2.28, 2.77, -1.89, 2.47, -2.24)	(12.91, 27.57, 3.01, 14.68, 2.31, 22.99, 2.24, 16.05)
160	(2.72, -1.81, 2.47, -2.28, 2.77, -1.89, 2.47, -2.24)	(12.94, 27.58, 2.96, 14.95, 3.17, 23.22, 2.88, 16.13)
180	(2.72, -1.81, 2.47, -2.28, 2.77, -1.89, 2.47, -2.24)	(12.44, 27.49, 3.38, 15.17, 2.96, 23.03, 3.02, 16.43)
194	(2.81, -2.29, 2.78, -1.89, 2.41, -2.57, 2.77, -1.95)	(3.74, 22.13, 4.31, 13.85, 2.97, 13.54, 6.94, 16.32)
200	(2.72, -1.81, 2.47, -2.28, 2.77, -1.89, 2.47, -2.24)	(12.71, 27.52, 4.40, 15.48, 3.31, 23.06, 2.98, 16.44)
220	(2.70, -2.23, 2.87, -1.88, 2.56, -2.53, 2.83, -1.91)	(6.07, 17.19, 4.49, 9.34, 6.73, 10.77, 7.28, 16.56)
251	(2.55, -1.54, 2.76, -1.61, 2.47, -1.50, 2.76, -1.62)	(11.43, 6.90, 12.74, 11.34, 12.59, 8.90, 16.12, 12.71)

7.2 Field Results in Marseille 2025

Grasping of a car seat and a tire on the media day In the following, we analyze selected sensor outputs from the Marseille demo in September 2025. We focus on a representative grasping maneuver of a car seat to verify that the grasping algorithms and hardware components behave as expected. The result for grapsing object is recorded in Table 5. The grasped object is shown in Figure 32. At the time of grasping,

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union



Figure 31: The smart grapple lifting a box of 250 kg under controlled conditions.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

Object	Weight (USV Force Sensor Reading)	Dry Weight (Only Object)
Chair (1)	—	24 kg
Tire	—	6 kg
Car Seat	185 kg (all together)	45 kg
Fences (×2)	156 kg (everything / force sensor)	14 kg
Ship wreck	185 kg	—

Table 5: Objects grasped in Marseille with corresponding weights. Dry weight is measured separately outside of the water, if available. The USV sensor reading includes the weight of the smart grapple and additional components.

which occurs shortly after $t \approx 800\,\mathrm{s}$ in the recorded ROS bag, the shoulder joints move from their open to their closed configuration (Figure 33a), and the elbow joints follow the same pattern (Figure 33b). As soon as the smart grapple closes, the joint torques at all degrees of freedom increase noticeably, again shortly after $800\,\mathrm{s}$, as illustrated in Figure 34.



(a) Grapple view: downward-facing camera.



(b) View from the control center.

Figure 32: Grasping of a car seat during the media demo day in Marseille, September 2025.

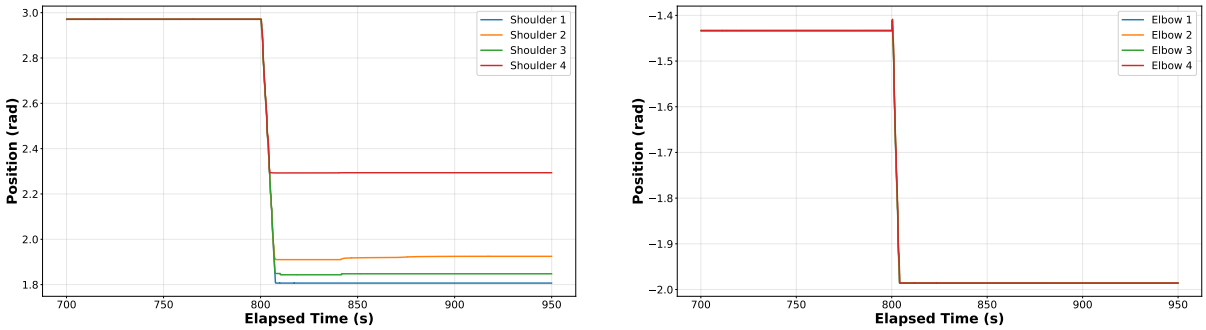
Besides grasping smaller objects, such as the car seat (and, in separate trials, a tire), an attempt to grasp a larger piece of shipwreck was also conducted successfully, as seen in Figure 35, although the contact area was too large to recover it completely from the water.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union



(a) Shoulder joint positions [rad].

(b) Elbow joint positions [rad].

Figure 33: Closing motion of the USG to grasp the car seat. The joint positions can be related to the reference frame introduced in Figure 21. The USG begins to grasp around $t \approx 800\,\mathrm{s}$.

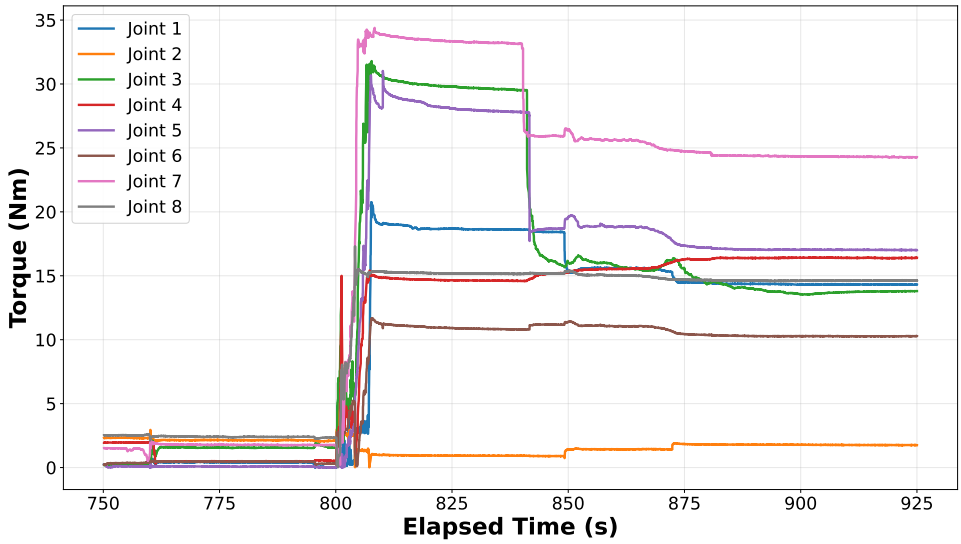


Figure 34: Joint torque profiles during the grasping maneuver, showing a clear increase shortly after $t \approx 800\,\mathrm{s}$ once there is contact with the car seat.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS
Co-funded by the European Union



Figure 35: Attempted grasp of a shipwreck fragment during the media demo day.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

8 Conclusions

This deliverable successfully documents a comprehensive software framework for the SeaClear2.0 Smart Grapple system, demonstrating significant advances of the system for autonomous underwater manipulation for marine litter collection. The integrated approach combines a holistic optimal control, robust state estimation, adaptive grasping algorithms, and real-time perception. Key accomplishments include validated grasping capability for objects up to 250 kg, successful field demonstrations in Marseille 2025, and a modular software architecture enabling both autonomous and manual operation modes. The system's ability to handle environmental disturbances through stochastic optimal control and feedback augmentation addresses critical challenges in underwater robotics. Future work should focus on improving computational efficiency of the trajectory optimization (currently requiring 4-17 hours), enhancing visual servoing for poor visibility conditions, and further integration testing of the complete autonomous operation cycle. The digital twin infrastructure provides a solid foundation for continued development between sea trials.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

References

[1] G. Welch and G. Bishop, "An introduction to the kalman filter," USA, Tech. Rep., 1995.

[2] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, July 2014.

[3] N. A. Jenssen and B. Realfsen, "Power optimal thruster allocation," in *Proc. dynamic positioning conference, Houston*, 2006.

[4] Simple open ethercat master. [Online]. Available: https://github.com/OpenEtherCATsociety/SOEM.git

[5] A. Hornung, K. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, 04 2013.

[6] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, "Informed sampling for asymptotically optimal path planning," *IEEE Transactions on Robotics*, vol. 34, no. 4, p. 966–984, Aug. 2018. [Online]. Available: http://dx.doi.org/10.1109/TRO.2018.2830331

[7] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December 2012, https://ompl.kavrakilab.org.

[8] Y. Origane, N. Hoischen, T.-Y. Huang, D. Kurabayashi, S. Sosnowski, and S. Hirche, "Risk-aware trajectory optimization and control for an underwater suspended robotic system," in *16th IFAC Conference on Control Applications in Marine Systems, Robotics and Vehicles (CAMS)*, 2025.

[9] T. Lew, R. Bonalli, and M. Pavone, "Risk-averse trajectory optimization via sample average approximation," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1500–1507, 2023.

[10] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne *et al.*, "Jax: composable transformations of python+ numpy programs," 2018.

[11] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," vol. 106, no. 1, pp. 25–57. [Online]. Available: https://doi.org/10.1007/s10107-004-0559-y

[12] A. Millane, H. Oleynikova, E. Wirbel, R. Steiner, V. Ramasamy, D. Tingdahl, and R. Siegwart, "nvblox: Gpu-accelerated incremental signed distance field mapping," 2023.

[13] D. Werner, A. Al-Hamadi, and P. Werner, *Truncated Signed Distance Function: Experiments on Voxel Size*. Springer International Publishing, 2014, pp. 357–364.

[14] A. Aristidou and J. Lasenby, "FABRIK: A fast, iterative solver for the inverse kinematics problem," *Graph. Models*, vol. 73, no. 5, pp. 243–260, Sep. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.gmod.2011.05.003

[15] I. Lončar, J. Obradović, N. Kraševac, L. Mandić, I. Kvasić, F. Ferreira, V. Slošić, Đ. Nađ, and N. Mišković, "Marus - a marine robotics simulator," in *OCEANS 2022, Hampton Roads*, 2022, pp. 1–7.

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

A Parameter lists

A.1 Thrust allocation

Table 6: Nominal thruster positions and orientation angles (Blue Robotics T500). Note the $90°$ increments in ψ_z around $+Z_{\text{ref}}$ (see Figure 4) between consecutive thrusters.

Thruster	VESC ID	Nominal position $\mathbf{p}_i$ [m]			Angles [deg]			
		p_x	p_y	p_z	θ_y	θ_z	θ_x	ψ_z
1	16	0.23586	-0.23906	-0.13283	-45	45	-180	-54
2	140	0.23586	0.23906	-0.13283	-45	45	-180	36
3	132	-0.23586	0.23906	-0.13283	-45	45	-180	126
4	40	-0.23586	-0.23906	-0.13283	-45	45	-180	216
5	32	0.23586	-0.23906	0.34903	-45	45	-90	-54
6	124	0.23586	0.23906	0.34903	-45	45	-90	36
7	116	-0.23586	0.23906	0.34903	-45	45	-90	126
8	24	-0.23586	-0.23906	0.34903	-45	45	-90	216

A.2 Kinematic parameters

Table 7: The kinematic parameters of the smart grapple system.

Parameter	Value	Description
l_1	323.09 mm	the first link of the finger
l_2	381.40 mm	the second link of the finger
x_0	280 mm	x base position of the finger
y_0	0 mm	y base position of the finger

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union

A.3 Holistic control approach simulation parameters

Table 8: List of relevant parameters to the simulation

Parameter	Value (Mean)	Uncertainty (Std. Dev.)	Description
m_{USG}	210 kg	73 kg	Mass of the USG (120 kg) + payload sampled from Weibull distribution with $\lambda = 100$ and shape $k = 1.5$
$m_{\text{USG,water}}$	90 kg	31 kg	Apparent mass of USG underwater
m_{USV}	1300 kg	—	Mass of SeaCAT (no LARS, full fuel tanks)
$\boldsymbol{\eta}_{\text{USG}}$	$[120, 120, 120, 0, 0, 0]^T$	—	USG friction coefficients (kg·s/m^2)
$\boldsymbol{\eta}_{\text{USV}}$	$[120, 120, 120, 0, 0, 0]^T$	—	USV friction coefficients (kg·s/m^2)
η_l	300 kg/s	—	Friction coefficient for the winch
α	0.1	—	Risk parameter
ΔT	20 ms	—	Simulation discretization time step
T_Δ	50	—	Multiplicity factor between simulation timestep and control input steps
S_T	120	—	Number of control input timesteps
$N_{\text{max,iter}}$	600	—	Maximum number of iterations

SeaClear2.0 is co-funded by the European Union under the Horizon Europe Programme (Grant Agreement 101093822). Views and opinions expressed herein are those of the author(s) and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them

© European Union, 2021. All rights reserved

PART OF THE EU MISSIONS RESTORE OUR OCEAN & WATERS

Co-funded by the European Union