ENHANCING UNDERWATER OPERATIONS THROUGH ADVANCED AUTONOMOUS MANIPULATION

Roger Pi Roig

Universitat de Girona

Doctoral Thesis

Enhancing Underwater Operations through Advanced Autonomous Manipulation

Roger Pi Roig

2024

Universitat de Girona

Doctoral Thesis

Enhancing Underwater Operations through Advanced Autonomous Manipulation

Roger Pi Roig

2024

Doctoral Program in Technology

Supervised by:

Pere Ridao Rodríguez
Narcís Palomeras Rovira

Thesis submitted to the University of Girona
in fulfillment of the requirements for the degree of

Doctor of Philosophy

Certificate of Thesis Direction

Prof. Pere Ridao Rodríguez, head of the Centre d'Investigació en Robòtica Submarina (CIRS) and member of the *Departament d'Arquitectura i Tecnologia de Computadors* of *Universitat de Girona*, and Dr. Narcís Palomeras Rovira, member of the *Departament d'Arquitectura i Tecnologia de Computadors* of *Universitat de Girona*,

Declare:

That the work entitled *Enhancing Underwater Operations through Advanced Autonomous Manipulation* presented by Roger Pi Roig to obtain the degree in Doctor of Philosophy has been compleded under our supervision.

Therefore, in order to certify the aforesaid statement, we sign this document.

Girona, April 2024

Prof. Pere Ridao

Dr. Narcís Palomeras

A tots aquells que l'han inspirat
i que mai la llegiran

Agraïments

Els últims anys han sigut un període intens d'aprenentatge i creixement, en els quals he passat incomptables hores davant de l'ordinador i barallant-me amb els robots. Per sort, aquesta experiència ha anat acompanyada d'un gran suport de persones increïbles que m'han ajudat a superar els reptes i celebrar els èxits.

Aquesta tesi no hauria sigut possible sense en Pere i en Narcís, els meus directors de tesis, que m'han guiat durant tots aquests anys i sempre han sigut pacients i constructius. Amb ells, he compartit moments de pluja d'idees, tant assenyades com esbojarrades, i moments d'anàlisi i reflexió, que demostren la dedicació i passió que posen en la recerca.

Aquesta tesi tampoc hauria sigut el mateix sense tota la gent amb qui he coincidit al CIRS durant el transcurs d'aquest doctorat i amb els quals hem compartit bons moments i hem superat els reptes que hem anat trobant.

Una menció especial a en Miguel, en Pau, en Patryk i en Guillem, els quals sempre s'han implicat per ajudar en tot el possible. Moltes gràcies a en Lluís Magí, que posseeix l'habilitat excepcional de solucionar qualsevol problema de hardware amb una facilitat sorprenent, i també moltíssimes gràcies a la resta de gent del grup: Marta, Valerio, Eduardo, Joan, Hayat, Nuno, Marc, Rafa, Mireia i Anna, i a tota la gent d'Iqua també pel seu suport: Lluís, Tali, David, Joseta, Àlex, Carles, Eduard i Jep.

Per acabar, el meu agraïment més especial és per a la meva família, la meva parella, la Mar, i tota la gent a la qual estimo i que tant m'han acompanyat durant aquest viatge. Sempre m'heu animat, ajudat i m'heu fet costat sense importar la situació, i és gràcies a vosaltres que he arribat fins aquí.

Acknowledgments

The past few years have been a period of intense learning and growth, fueled by countless hours in front of screens and alongside whirring robots. Thankfully, this journey was immeasurably enriched by the support of incredible people, who helped me navigate the challenges and celebrate the successes.

This thesis wouldn't have been possible without Pere and Narcis, my supervisors, who have walked with me throughout this journey and have always been patient and constructive. With them, I have shared many moments of brainstorming, some of them worthy of a movie, as well as moments of analysis and reflection, which demonstrate the dedication and passion they put into the research.

It also would not have been the same without all the people with whom I have coincided at CIRS during the course of this Ph.D. thesis, and with whom we have shared good moments and overcome the challenges we have encountered.

A special mention goes to Miguel, Pau, Patryk, and Guillem, who consistently went above and beyond to offer their help and support. Many thanks to Lluís Magí, who has the unique ability to solve any hardware problem with surprising ease, and also very special thanks to the rest of the people of the group: Marta, Valerio, Eduardo, Joan, Hayat, Nuno, Marc, Rafa, Mireia, and Anna, and also to all the people of Iqua for their support: Lluís, Tali, David, Joseta, Àlex, Carles, Eduard and Jep.

Finally, my most special thanks go to my family, my partner, Mar, and all the people I love, who have accompanied me during this journey. You have always encouraged, helped, and supported me, and I don't doubt that without you I would have never arrived here.

List of Publications

Publications in the compendium

The presented thesis is a compendium of the following research articles:

- **Roger Pi**, Patryk Cieślak, Pere Ridao, and Pedro J Sanz. “Twinbot: Autonomous underwater cooperative transportation”. In: *IEEE Access* 9 (2021), pages 37668–37684. DOI: 10.1109/ACCESS.2021.3063669
  Quality index: JCR2021 Engineering, Electrical & Electronic, Impact Factor: 3.476, Q2 (105/276).

- **Roger Pi**, Patryk Cieślak, Joan Esteba, Narcís Palomeras, and Pere Ridao. “Compliant Manipulation With Quasi-Rigid Docking for Underwater Structure Inspection”. In: *IEEE Access* 11 (2023), pages 128957–128969. DOI: 10.1109/ACCESS.2023.3332486
  Quality index: JCR2022 Engineering, Electrical & Electronic, Impact Factor: 3.9, Q2 (100/275).

- **Roger Pi**, Pau Vial, Narcís Palomeras, and Pere Ridao. “Robust calibration of multi-sensor multi-manipulator mobile robots using nonlinear on-manifold optimization”. Submitted to IEEE Transactions on Robotics (2024)
  Quality index: JCR2022 Robotics, Impact Factor: 7.8, Q1 (5/42).

Publications derived from this thesis

The work developed in this thesis also led to the following publications:

- Miguel Castillón, **Roger Pi**, Narcís Palomeras, and Pere Ridao. “Extrinsic visual–inertial calibration for motion distortion correction of underwater 3D scans”. In: *IEEE Access* 9 (2021), pages 93384–93398. DOI: 10.1109/ACCESS.2021.3092180

- **Roger Pi**, Dina Youakim, Patrick Cieslak, and Pere Ridao. “Multi-representation multi-heuristic A* motion planning for a dual-arm underwater vehicle manipulation system”. In: *IFAC-PapersOnLine* 52.21 (2019), pages 205–210. DOI: 10.1016/j.ifacol.2019.12.308

- **Roger Pi**, Joan Esteba, Patryk Cieslak, Narcís Palomeras, Pedro J Sanz, Raúl Marín, and Pere Ridao. "OPTIHROV: Optically Linked Hybrid Autonomous/Remotely Operated Vehicle, Beyond Teleoperation in a New Generation of Underwater Intervention Vehicles". In: *OCEANS 2023-Limerick*. IEEE. 2023, pages 1–7. DOI: 10.1109/OCEANSLimerick52467.2023.10244690

Acronyms

AUV	autonomous underwater vehicle
BT	Behavior Tree
CP	cathodic protection
DoF	degree of freedom
DP	dynamic positioning
DVL	Doppler velocity log
EE	end-effector
FT	force-torque
I-AUV	intervention autonomous underwater vehicle
IMR	inspection, maintenance, and repair
INS	inertial navigation system
LLM	Large Language Model
ROV	remotely operated vehicle
SLAM	simultaneous localization and mapping
SVS	sound velocity sensor
TMS	tether nanagement system
TP	Task Priority

Institutions

CIRS	Centre d'Investigació en Robòtica Submarina
UdG	Universitat de Girona
ViCOROB	Institut de Recerca en Visió per Computador i Robòtica

Contents

List of Figures

Abstract

The interest in the use of autonomous underwater vehicles (AUVs) has increased in recent decades. While former research focused on underwater exploration for sea bottom mapping (bathymetries, sonar, and photo mosaics), it evolved soon into 3D optical reconstruction and offshore infrastructure inspection. Progress in these areas has sparked the interest of the community in employing AUVs for intervention tasks, thereby replacing remotely operated vehicles (ROVs) and manned submersibles with intervention autonomous underwater vehicles (I-AUVs). This substitution offers the potential to automate tasks, improving efficiency and repeatability while reducing costs, time, and logistics. However, autonomous intervention underwater is challenging. It requires the joint control of a heterogeneous multibody system composed of the AUV and the manipulators, which have significant differences in terms of control and accuracy. Most intervention tasks, such as object grasping or valve turning, require centimeter accuracy in the position of the end effector. This accuracy is severely affected by a chain of errors, beginning with the navigation error and continuing with the calibration errors of the involved systems, including inaccuracies in the positions of the cameras, lasers, and manipulators, joint calibration errors, and other uncertainties within the system. Another challenge is the manipulation of bulky objects which are difficult, if not impossible, to satisfy with a single vehicle. Most probably, future autonomous intervention systems will be multi-robot. This poses new problems to solve, like the joint control of a team of I-AUVs coordinated through low bandwidth communication channels. Finally, it is necessary to root the autonomous underwater intervention research to the actual needs of field applications. This thesis is a contribution along these lines. It aims to advance the autonomous underwater intervention state of the art to increase the autonomy of I-AUVs for inspection, maintenance, and repair (IMR) tasks in offshore infrastructures. First, a new framework is proposed to calibrate the intrinsic/extrinsic parameters of the I-AUVs components, using robust modeling of the minimization equations leveraging Lie theory. Then, the Task Priority redundancy control algorithm is enhanced to control two I-AUVs, communicating through a low-rate communications channel to transport a bulky object. Finally, an effort is made to study the actual capabilities of I-AUVs to face field applications in the area of offshore renewable energies. A Task Priority algorithm supporting admittance control is used to control an I-AUV performing non-destructive inspection for cathodic protection on a floating semi-submersible windmill structure. Throughout the thesis, all the works present both simulation and experimental results, validating the efficiency and potential of the proposed solutions.

Resum

L'interès per l'ús de vehicles autònoms submarins (AUVs) ha augmentat en les últimes dècades. Mentre que la recerca prèvia se centrava en l'exploració submarina per la cartografia del fons marí (batimetries, sonar i mosaics fotogràfics), aviat va evolucionar cap a la reconstrucció òptica 3D i la inspecció d'infraestructures en alta mar. Els avenços en aquests àmbits han despertat l'interès de la comunitat científica per utilitzar AUVs per les tasques d'intervenció, substituint així els vehicles operats remotament (ROVs) i els submergibles tripulats per vehicles autònoms submarins d'intervenció (I-AUVs). Aquesta innovació ofereix la possibilitat d'automatitzar les tasques, millorant l'eficàcia i repetibilitat alhora que es redueix el cost, temps i logística d'aquestes operacions.

Tanmateix, la intervenció autònoma submarina és un gran repte. Requereix el control conjunt d'un sistema heterogeni compost pel vehicle i els braços robòtics (o manipuladors) que presenten diferències significatives en termes de control i precisió. La majoria de les tasques d'intervenció, com la recuperació d'objectes o la manipulació de vàlvules, requereixen una precisió de centímetres, fins i tot mil·límetres, en la posició de la pinça. Aquesta precisió es veu greument afectada per una cadena d'incerteses, que comença amb els errors de navegació del vehicle i continua amb els errors de calibració dels sistemes implicats, incloent-hi les imprecisions en les posicions de les càmeres, làsers i manipuladors, els errors de calibració de les articulacions, i altres incerteses del sistema.

Un altre repte és la manipulació d'objectes voluminosos, que és difícil, si no impossible, de satisfer amb un sol vehicle i requereix l'ús de sistemes multi robot. Això planteja nous problemes a resoldre, com el control conjunt d'un equip d'I-AUVs coordinats a través de canals de comunicació amb poca amplada de banda. Finalment, és necessari arrelar la recerca en intervenció autònoma submarina a les necessitats actuals de les aplicacions de camp. Aquesta tesi ofereix una contribució en aquesta línia. L'objectiu és avançar en l'estat de l'art de la intervenció autònoma submarina per augmentar l'autonomia dels I-AUVs en les tasques d'inspecció, manteniment i reparació d'infraestructures marítimes. Primer, es proposa un nou sistema per calibrar els paràmetres intrínsecs i extrínsecs dels components del robot, utilitzant un model robust de les equacions de minimització emmarcat en la Teoria de Lie. A continuació, es proposa un algoritme descentralitzat de control i gestió de redundància d'un sistema compost per dos robots, que transporten cooperativament un objecte voluminós, mentre es comuniquen per una línia de baixa amplada de banda. Finalment, els esforços es dirigeixen a estudiar les capacitats reals dels I-AUVs per afrontar aplicacions en l'àmbit de la generació d'energies renovables en alta mar. Es proposa una extensió en l'algorisme de control del robot per suportar el control actiu de força i poder dur a terme inspeccions de la protecció catòdica en l’estructura semi submergible d’un aerogenerador d’alta mar. Al llarg de la tesi, tots els treballs presenten resultats tant en simulació com experimentals, validant la eficiència i potencial de les solucions proposades.

Resumen

El interés por el uso de vehículos autónomos submarinos (AUVs) ha aumentado en las últimas décadas. Mientras que la investigación previa se centraba en la exploración submarina por la cartografía del fondo marino (batimetries, sonar y mosaicos fotográficos), pronto evolucionó hacia la reconstrucción óptica 3D y la inspección de infraestructuras en alta mar. Los adelantos en estos ámbitos han despertado el interés de la comunidad científica para utilizar AUVs por las tareas de intervención, sustituyendo así los vehículos operados remotamente (ROVs) y los sumergibles tripulados por vehículos autónomos submarinos de intervención (I-AUVs). Esta innovación ofrece la posibilidad de automatizar las tareas, mejorando la eficacia y repetibilitat a la vez que se reduce el coste, tiempo y logística de estas operaciones.

Aun así, la intervención autónoma submarina es un gran reto. Requiere el control conjunto de un sistema heterogéneo compuesto por el vehículo y los brazos robóticos (o manipuladores) que presentan diferencias significativas en términos de control y precisión. La mayoría de las tareas de intervención, como la recuperación de objetos o la manipulación de válvulas, requieren una precisión de centímetros, incluso milímetros, en la posición de la pinza. Esta precisión se ve gravemente afectada por una cadena de incertidumbres, que empieza con los errores de navegación del vehículo y continúa con los errores de calibració de los sistemas implicados, incluyendo las imprecisiones en las posiciones de las cámaras, láseres y manipuladores, los errores de calibració de las articulaciones, y otras incertidumbres del sistema.

Otro reto es la manipulación de objetos voluminosos, que es difícil, si no imposible, de satisfacer con un solo vehículo y requiere el uso de sistemas multirrobot. Esto plantea nuevos problemas a resolver, como el control conjunto de un equipo de AUVs coordinados a través de canales de comunicación con poco ancho de banda. Finalmente, es necesario arraigar la investigación en intervención autónoma submarina a las necesidades actuales de las aplicaciones de campo. Esta tesis ofrece una contribución en esta línea. El objetivo es avanzar en el estado del arte de la intervención autónoma submarina para aumentar la autonomía de los AUVs en las manchas de inspección, mantenimiento y reparación de infraestructuras marítimas. Primero, se propone un nuevo sistema para calibrar los parámetros intrínsecos y extrínsecos de los componentes del robot, utilizando un modelo robusto de las ecuaciones de minimización enmarcado en la Teoría de Lie. A continuación, se propone un algoritmo descentralizado de control y gestión de redundancia de un sistema compuesto por dos robots, que transporten cooperativamente un objeto voluminoso, mientras se comunican por una línea de bajo ancho de banda. Finalmente, los esfuerzos se dirigen a estudiar las capacidades reales de los AUVs para afrontar aplicaciones en el ámbito de la generación de energías renovables en alta mar. Se propone una extensión en el algoritmo de control del robot para soportar el control activo de fuerza y poder llevar a cabo inspecciones de la protección catódica en lo estructura semisumergible de un aerogenerador de alta mar. A lo largo de la tesis, todos los trabajos presentan resultados tanto en simulación como experimentales, validando la eficiencia y potencial de las soluciones propuestas.

1

Introduction

This chapter summarizes the motivation behind the development of this Ph.D. thesis. First, the background of autonomous underwater intervention and previous work in the field are presented in Section 1.1. Then, Section 1.2 presents the challenges that motivate the research direction taken in this Ph.D. thesis. Next, Section 1.3 outlines the specific objectives and goals set to address the identified challenges, and Section 1.4 describes the context in which this work has been developed. Finally, Section 1.5 concludes with a summary of the organization of this document.

1.1 State of the Art

Today, autonomous underwater vehicles (AUVs) are predominantly used for survey missions, serving as practical tools for collecting oceanographic data, creating detailed bathymetric maps, and generating photomosaics. However, as technology advances and the demands of underwater activities evolve, the current capabilities of AUVs fall short of enabling direct intervention operations. Such applications include marine rescue, marine science, archaeology, and inspection, maintenance, and repair (IMR) tasks common in offshore industries. These intervention operations at sea remain extremely costly and time-consuming, requiring the use of heavy-weight remotely operated vehicles (ROVs) supported by large dynamic positioning (DP) vessels and complex tether nanagement systems (TMSs). In the last three decades, research on autonomous underwater robots and robotic intervention has been slowly gaining speed, with the aim of tackling some of the IMR tasks that could be performed in the future by intervention autonomous underwater vehicles (I-AUVs). However, the practical implementation of these aspirations remains a challenge, and the majority of contributions in this field are theoretical or confined to simulation environments.

Research began with the pioneering works of OTTER [7], ODIN [8], UNION [9], and AMADEUS [10], which contributed to the development of core technologies, but it was not until the first decade of the 21st century that field demonstrations arrived. Due to the difficulty of demonstrations, most of the trials were carried out in mock-up or constrained sea environments, focusing on two types of applications: **Object search and recovery**, and **inspection, maintenance, and repair (IMR)**.

The first results on floating manipulation were achieved in 2009 within the SAUVIM project [11, 12], demonstrating the ability to find an object whose position was roughly known a priori. The object was endowed with artificial landmarks, and the robot autonomously located it and hooked it up from the seabed with a recovery device while hovering. A 6-ton vehicle was used, hence the mass of the arm did not cause significant disturbances. The same task was later addressed in the RAUVI project [13] using a lighter vehicle (<200 Kg). A multipurpose object search and recovery strategy was proposed and organized in two steps: 1) first, the I-AUV performed an optical survey of the area of interest, building a photomosaic, and 2) the user selected a target object in the photomosaic, and the robot was sent to recover it autonomously. These trials were performed in a water tank. The project RAUVI evolved to the European project TRIDENT [14], and the experimental trials were carried out in a harbor environment, using a 7 DoF manipulator endowed with a specially designed 3 finger hand, demonstrating the first multipurpose object search and recovery strategy at sea.

In the context of IMR, recent research focuses on representative tasks such as "valve turning" and "connector plug / unplug" operations. The first fully autonomous operation at sea was demonstrated by the ALIVE project [15], where a 1.5-ton I-AUV used a mechanical scanning imaging sonar to locate and approach a subsea panel. Then, the robot docked to the structure using hydraulic grasps and performed a fixed-base valve turning operation. Similar experiments were later reproduced in the TRITON [16] project, using a significantly lighter I-AUV for valve turning and hot stab connection operations. A more challenging approach is to perform intervention while the vehicle is floating, requiring seamless coordination between the vehicle and its manipulator while reacting to dynamic environmental changes. The first autonomous floating valve turning operation was demonstrated during the PANDORA European project, first using learning-by-demonstration techniques [17] and then using Task Priority (TP) control [18]. The later work was extended in [19] to perform a valve-turning operation in the presence of obstacles. The work presented a reactive obstacle avoidance formulation within the TP framework, assuming a prior knowledge of the obstacles' positions.

Additionally, in [20], the authors proposed the use of a 3D laser scanner to build an occupancy map online used for motion planning, enabling autonomous manipulation in the presence of *a priori* unknown obstacles.

In the context of the DexRov project, the work in [21] also demonstrated a valve turning operation using the TP framework. They included an admittance loop to exploit force/torque measurements of a wrench sensor and make the system compliant with external forces or unexpected collisions with the environment. Experimental results were presented using a dual-arm ROV, which clamped to a structure close to the target panel to perform the valve turning operation. Admittance control was also demonstrated in the context of pipe inspection in [22], using the force/torque measurements to ensure a robust contact between the inspection tool and the inspected surface during the floating-base operation.

1.2 Autonomous underwater intervention challenges

Despite the advancements presented in the previous section, the field still struggles with slow incremental progress rather than substantial leaps forward. This section explores key challenges that limit I-AUVs from reaching their full potential for real-world tasks:

1. **The need for experimental field demonstrations:** The majority of advancements in the field still largely reside within the realm of theoretical propositions or confined simulation environments, lacking tangible outcomes. These limitations in autonomous underwater intervention capabilities impede the widespread adoption of I-AUVs for complex and large-scale operations. Consequently, the predominant reliance on heavy ROVs operated from large vessels persists in practical intervention operations despite their high costs, operational logistics, and scalability limitations. The eventual adoption of I-AUVs to perform such tasks requires first a move of the research community towards more experimental field demonstrations. The current focus on theoretical frameworks and simulations needs to evolve into practical implementations to bridge the gap between conceptual developments and real-world applications.

2. **High precision demands:** Underwater intervention inherently demands high precision. The intricate dynamics and unpredictable underwater environment, compounded by the limitations of the exteroceptive sensors, render control, localization, and mapping, still open research topics. However, the efficacy of these intricate components is fundamentally subjected to having a precisely calibrated system. The sensors' extrinsic and intrinsic parameters, along with their associated uncertainties, must be well characterized to perform precise measurements. In the same way, the location of the manipulators with respect to the exteroceptive sensors must also be well characterized. Otherwise, the robot may completely miss or even collide with the target object. Additionally, it is important to know precisely the intrinsic geometry properties of the manipulators, as small joint offset errors can propagate to centimeter inaccuracies in the end-effectors.

3. **Scalability and size:** The current reliance on large, heavy Work Class ROVs for intervention operations poses significant challenges in terms of scalability and cost. The transition towards autonomous intervention necessitates exploring alternatives, and lightweight I-AUVs offers a promising solution. These smaller robots can be deployed from virtually any vessel, significantly reducing operational costs and logistical complexity. Additionally, their smaller size translates to lower acquisition and maintenance

costs, which facilitates the use of multiple robots. However, lighter platforms inherently have limited power and payload capacity.

4. **The Potential of Multi-Robot Systems:** Deploying multiple I-AUVs collaboratively presents a compelling strategy to overcome the limitations of single robots. Among other open possibilities, multi-robot systems can coordinate to share the load and optimize manipulator stress, enabling the manipulation of larger, heavier objects beyond the capabilities of individual vehicles, regardless of their size. Additionally, increased robustness and coverage can be achieved through group redundancy, where the failure of one robot does not necessarily compromise the entire mission. This approach holds promise for enhancing efficiency and reducing mission time in search and recovery scenarios, but the underwater domain forces these technologies to face additional challenges, such as the limited bandwidth available for communication.

1.3 Objectives

In light of these challenges and the relatively small-scale contributions in real, experimental demonstrations, the main goal of this thesis can be stated as:

> ***To research, develop, and demonstrate advanced solutions to enhance autonomous intervention capabilities of field I-AUVs.***

This general goal is further divided into the following objectives:

1. **Develop a solution to accurately calibrate the I-AUV**: Manipulation tasks require high precision, which cannot be achieved if the sensors' extrinsic and intrinsic parameters are poorly characterized. Therefore, it is important to develop a robust and generalizable calibration framework that can be used after altering the payload of an I-AUV.

2. **To advance on inspection, maintenance, and repair of semi-submergible structures using an I-AUV:** Currently, the energy industry is evolving toward the generation of renewable energy, which leads to the settlement of new offshore floating turbines. These structures will require IMR operations to reduce the need for extensive support vessels and human intervention. Therefore, floating semi-submergible structures become a perfect proving ground for demonstrating the capabilities of I-AUVs in real-world scenarios.

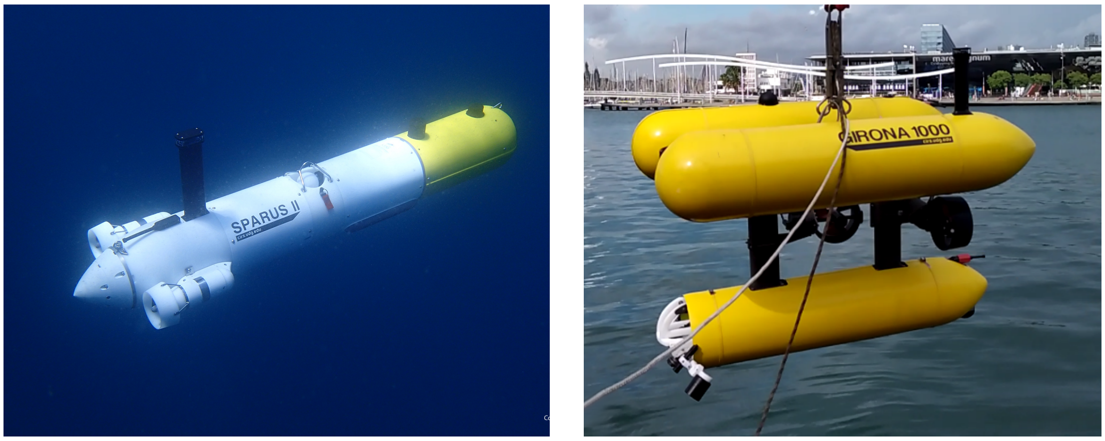
3. **To advance in cooperative manipulation capabilities for I-AUVs:** The use of multiple robots is a natural step forward in the research on autonomous underwater manipulation, as complex tasks may require an increased load capacity and dexterity, such as the cooperative transportation of large and bulky pipes.

4. **Validate experimentally the proposals:** To demonstrate the potential of using I-AUVs, it is important that all the works are validated experimentally with real robots.

1.4 Context

The research presented in this thesis has been developed at Centre d'Investigació en Robòtica Submarina (CIRS), which is part of the Institut de Recerca en Visió per Computador i Robòtica (ViCOROB) institute of the Universitat de Girona (UdG). The group started researching underwater vision and robotics in 1992 and has become a leading team in the underwater robotics and computer vision community. It has participated in several European and national projects (of both basic and applied research). It has also been involved in technology transfer projects and contracts with companies and institutions around the world. Moreover, ViCOROB has won several European multi-domain robotic competitions: SAUC-E (2006 and 2010) and Eurathlon (2014, 2015, and 2017).

During this time, the group has developed several AUV prototypes, being three of them currently fully operative: the Sparus II [23], the Girona500 [24], and the recently developed Girona1000, an upgraded version of the Girona500 (see Fig. 1.1). The Sparus II is a torpedo-shaped vehicle rated for up to 200 m depth that can cover long distances, making it ideal for missions such as photogrammetric sea-bottom surveying. The Girona500, rated for depths up to 500 m, is composed of an aluminum frame that supports three torpedo-shaped hulls of 0.3 m in diameter and 1.5 m in length. The overall dimensions of the vehicle are 1 m in height, 1 m in width, 1.5 m in length, and can carry a heavier payload, such as robotic manipulators.



(a) The Sparus II in open sea.

(b) The Girona1000 being deployed in the port of Barcelona (Catalonia).

Figure 1.1: AUVs developed at CIRS.

The line of research on autonomous underwater manipulation started at CIRS in 2009, when the Girona500 was equipped with a 4 DoF CSIP 5E manipulator from the ECA group [1] (see Fig. 1.2). Before this thesis, most of the work on underwater manipulation conducted in the lab was performed using this setup. These works include *Learning by Demonstration* [17, 25, 26], *Control Theory* [18, 19, 22], and *Motion Planning* [20, 27], among others.

However, this manipulator has a very limited workspace, as it only has 4 DoFs and it is driven by electric screw drives, which limits the range of each rotational joint and also makes it slow, although very strong. Consequently, even if the target is within the arm's reach, it is generally required to actuate the floating base to achieve the desired 6 DoF end-effector (EE) pose. Since the motion of the arm and the mobile base cannot be decoupled, the accuracy of the EE positioning gets reduced. Moreover, a faster manipulator could potentially counteract

[1] https://www.ecagroup.com/en/solutions/arm-5e-micro

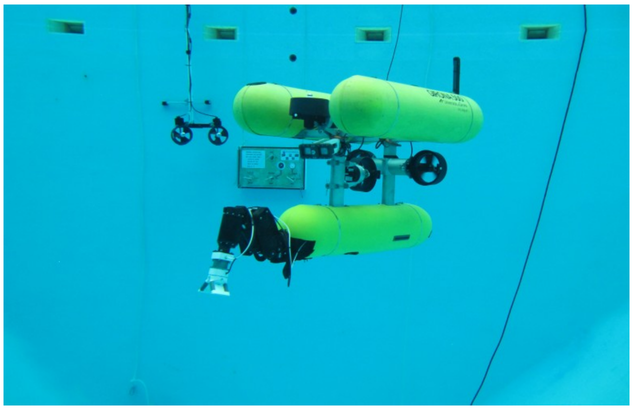


Figure 1.2: The Girona500 equipped with the ECA/CSIP 5E manipulator.

disturbances produced in the floating-based with fast, precise arm motions to have a smaller impact on the EE.

During the course of this Ph.D., the laboratory has progressively acquired two Bravo 7 manipulators from Reach Robotics [2]; a dexterous 6 DoF manipulator faster and more accurate, allowing precise manipulation (see Fig. 1.3). Moreover, recently, the Girona1000 AUV has been developed in the lab, an advanced version of the Girona500. Among its improvements, it is rated to reach depths of up to 1000 m (instead of 500 m) and incorporates a high-end inertial navigation system (INS).

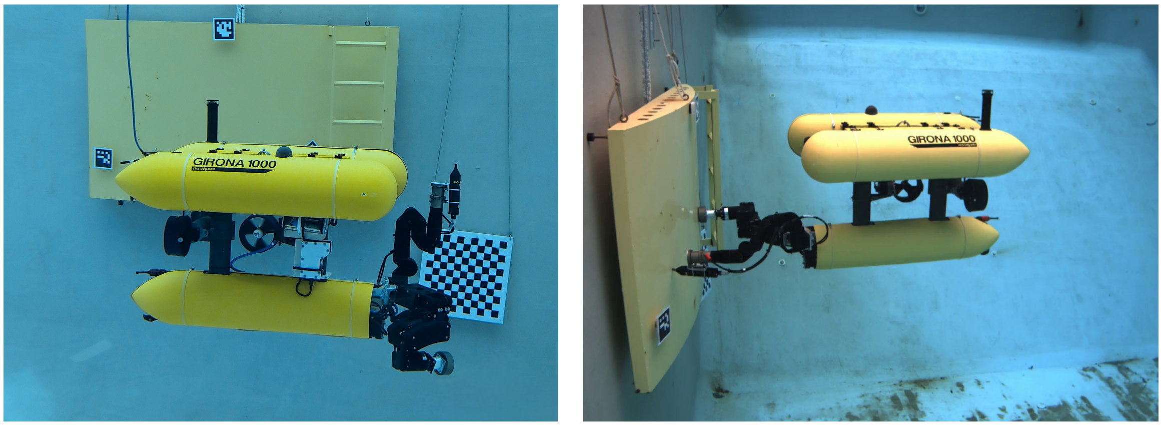
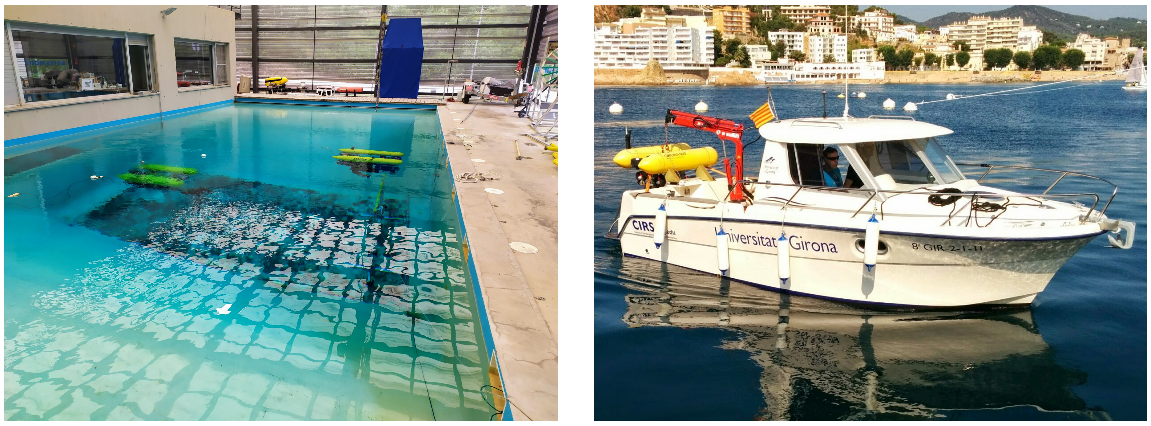


Figure 1.3: The Girona1000 is equipped with both ECA/CSIP and Bravo manipulators.

Furthermore, the laboratory counts with a fresh water tank of dimensions 16m × 8m × 5m (length × width × depth) and a crane-equipped boat named *Sextant* at St. Feliu de Guixols harbor (see Fig. 1.4). These infrastructures allow for relatively easy experimental data collection.

In this context, the work presented in this thesis shows a progression towards advanced intervention applications in the field of autonomous underwater manipulation.

[2] https://reachrobotics.com/products/manipulators/reach-bravo/



(a) CIRS water tank.

(b) *Sextant* boat.

Figure 1.4: Infrastructure for experiments.

This thesis was financed mainly by the Doctoral Grant of the Secretaria d'Universitats i Recerca del Departament d'Economia i Coneixement de la Generalitat de Catalunya (grant 2019FI_B_00812). The experiments, equipment, and infrastructure resources used in this thesis have been partially funded by the following projects:

- Project TWINBOT-GIRONA1000 (ref. DPI2017-86372-C3-2-R), funded by the Spanish Ministry of Economy, Industry. and Competitiveness.
- Project ATLANTIS (ref. H2020-ICT-2019-2-871571), funded by the European Commission.
- Project COOPERAMOS (ref. PID2020-115332RB-C32), funded by the Spanish Ministry of Science and Innovation.
- Project OPTIHROV (ref. PDC2021-120791-C21), funded by the Spanish Ministry of Science and Innovation.

1.5 Document structure

This document is structured into the following chapters:

- **Chapter 2. Robust calibration of multi-sensor multi-manipulator mobile robots using nonlinear on-manifold optimization**, covers the calibration procedure of the I-AUV sensors extrinsics and intrinsics.
- **Chapter 3. Autonomous underwater cooperative transportation**, demonstrates a fully autonomous cooperative pipe transportation using two Girona500 I-AUVs.
- **Chapter 4. Compliant manipulation with quasi-rigid docking for underwater structure inspection**, proposes a methodology for contact-based operation on semi-submergible structures. The work is experimentally demonstrated in the context of cathodic protection (CP) inspection of a wind turbine structure on a mock-up structure using the Girona1000 I-AUV.
- **Chapter 5. Results and Discussion** lays out the results obtained during this thesis, following the objectives presented previously.

- **Chapter 6. Conclusions and Future Work** concludes this thesis by summarizing the main contributions and future research directions.

Note that the arrangement of the articles within this compendium deviates from the original chronological order of publication. This intentional reordering aims to facilitate a smoother progression of ideas and concepts, ensuring a clearer understanding for the reader.

2

Robust calibration of multi-sensor multi-manipulator mobile robots using nonlinear on-manifold optimization

In this chapter, we present a robot calibration framework that estimates the extrinsic parameters of multiple cameras and manipulators, and the intrinsic parameters of the manipulators within a mobile robotic system. The calibration procedure not only optimizes the inter-relationships between the payload components but also references them to the inertial frame of the robot. The calibration problem is modeled as a factor graph and solved using iterative nonlinear optimization over Lie groups. The approach is validated through simulation and experimental implementation on the Girona500 I-AUV.

Title: Robust calibration of multi-sensor multi-manipulator mobile robots using nonlinear on-manifold optimization
Authors: **Roger Pi**, Pau Vial, Narcís Palomeras, and Pere Ridao
Submitted to: IEEE Transactions on Robotics
Quality index: JCR2022 Robotics, Impact Factor: 7.8, Q1 (5/42)

Robust calibration of multi-sensor multi-manipulator mobile robots using nonlinear on-manifold optimization

Roger Pi, Pau Vial, Narcís Palomeras, and Pere Ridao, *Member, IEEE*.

***Abstract*—This paper describes a calibration procedure that estimates the extrinsic parameters of multiple cameras and manipulators, and the intrinsic parameters of the manipulators, within a mobile robotic frame. The proposed method precisely calibrates the cameras and manipulators with respect to each other, providing high-accuracy inter-relationships between the payload components. Then, an ego-motion inertial calibration fuses the navigation sensors of the robot (accelerometers, gyroscopes, Linear Velocity Sensor (LVS), and pressure sensor) with visual feedback of a camera, thus referencing the payload components to the robot's inertial frame. The calibration problem is modeled as a factor graph and solved using iterative nonlinear optimization over Lie groups. Alongside the formulation of the factors, we provide the development of their analytic jacobians with respect to the estimated variables. We validate our proposal in simulation and on the Girona500 Intervention Autonomous Underwater Vehicle (I-AUV) from the Universitat de Girona, showing the convergence of the calibration towards the true values of the parameters.**

***Index Terms*—Mobile Robotics, Extrinsic Calibration, Factor Graphs, Lie Theory, Underwater Robotics**

I. Introduction

In the rapidly evolving field of robotics, the calibration of multiple sensors and manipulators of a robot is a critical aspect that significantly influences the performance and efficiency of the system. The inter-relationships between vision sensors and manipulators need to be accurately known in order to perform precise manipulation tasks. For instance, if the 3D pose of a camera is not well characterized relative to the robot's arm, the arm may completely miss or even collide with the target object. For even more accuracy, it is important to model misalignments such as manipulator joint offsets, which may originate due to manual assembly processes. Moreover, on a mobile robot it is necessary to estimate the inter-relationships of the aforementioned devices with respect to the inertial and motion sensors of the robot. Precise calibration ensures accurate perception, manipulation, and navigation, which are vital for the success of any robotic task. However, the calibration process may be challenging due to the different error characteristics of each sensor, how they perceive the world, and the high dimensionality of the problem.

An important aspect of any calibration methodology is the proper parameterization of the estimation variables, as they include rotations that do not form an Euclidean space. Common optimization methods like Gradient Descent, Gauss-Newton, or Newton's Method, as well as filtering methods such as the Extended Kalman Filter, are specifically designed for flat Euclidean spaces ($\mathbb{R}^N$). Consequently, the application of these methods to non-Euclidean spaces is not straightforward.

In optimization problems involving orientations, 3D poses are typically parameterized as:

- 3D translation + Euler Angles (6 elements),
- 3D translation + unit quaternion (7 elements), or
- 3D translation + rotation matrix $\in \mathbb{R}^{3\times 3}$ (12 elements),

being none of them ideal solutions. Firstly, Euler angles achieve a minimal representation (6 elements for 6 DOFs) but are prone to gimbal lock, where one DOF is lost. This situation affects the optimization, breaking the continuity of the problem and thus losing the conditions to solve the problem. Furthermore, after an optimization step $(x \leftarrow x+\delta)$, the angles may go out of range and require renormalization. Secondly, the use of quaternions, on the other hand, avoids gimbal lock. However, the extra DOF requires the addition of a normalization constraint in the optimization problem, typically managed by adding a Lagrange multiplier to the state vector. This constrained problem can slow down the convergence of the minimization process and reduce its precision. Moreover, unit quaternions must be kept in the same hemisphere to ensure a smooth optimization trajectory, adding another layer of complexity. Finally, using a rotation matrix implies defining an even more constrained optimization, which negatively impacts the dimensionality and convergence of the optimization problem.

To address these algebraic issues involving orientations and poses, we can acknowledge that these variables form Lie groups [1*]. In particular, the Lie group of rigid 3D transformations is the Special Euclidean group $SE(3)$, which can be parameterized by a homogeneous matrix $\in \mathbb{R}^{4\times 4}$ or, in a more disruptive form, using dual quaternions [2], [3]. By formulating the optimization problem directly on Lie groups instead of on Euclidean spaces, we can increase the convergence and precision of the solver, avoiding a constrained optimization problem. This methodology consists of a specialization of Riemannian optimization [4] for Lie groups, which ensures the continuity and mathematical correctness of the problem. Note that even though there are general-purpose solvers that can solve constrained optimization problems, Riemannian optimization reformulates the estimation problem to become unconstrained, achieving theoretical and practical

This work is supported by the Secretaria d'Universitats i Recerca del Departament d'Economia i Coneixement de la Generalitat de Catalunya under grant 2019FI_B_00812.

advantages; unconstrained optimization is more efficient and scalable since it leverages the natural geometry of the problem, the formulation becomes more elegant, and provides a robust handling of uncertainties, defining probability distributions at the tangent spaces of the Lie groups [5].

Finally, special attention must be directed towards defining derivatives for the optimization policy required by the solver algorithm in Riemannian optimization. Using numerical or automatic differentiation requires careful consideration to ensure that the algorithm adequately defines and utilizes the problem geometry, preventing differentiation in a purely Euclidean space. Additionally, numerical differentiation demands a well-defined problem geometry to incorporate perturbations at the tangent space of the group element and avoid skipping the group. Another option is to compute symbolic derivatives by applying the chain rule, leveraging already defined Jacobians of well-studied Lie groups, and reaching an analytical form for the derivatives. In the case of 3D rigid transformations, two possibilities exist. On the one hand, the Jacobians of the $SE(3)$ group can be used to treat 3D transformations as a block. On the other hand, translation and rotation can be considered independently within the $<\mathbb{R}^3, SO(3)>$ composite manifold and use the derivative of the Rodrigues equation provided in [6]. However, it's important to acknowledge that handling rotation and translation independently implies that the solver also updates them separately, a practice that may not be considered ideal.

In this study, we opt for symbolic differentiation considering the $SE(3)$ group to reduce the computational complexity of the calibration algorithm and ensure the appropriate geometry is utilized for derivative computations.

A. Related Work

Many studies have focused on developing specific techniques for pairwise calibration between sensors. In [7], the authors propose calibrating a 2D laser rangefinder with a camera using line constraints on a chessboard pattern. They treat translation and rotation independently, the latest parameterized using Rodrigues parameters, and solve the problem using the Levenberg-Marquardt (LM) method applying numerical differentiation. A similar work was done in [8], but using plane constraints from a 3D laser rangefinder. In this work, they opted for Nelder-Mead simplex optimization [9] and parameterized rotations as quaternions. However, they do not mention how they constrain its unit norm. In [10], the authors propose a calibration for range and image sensors that exploits natural features in the scene to avoid the use of calibration patterns. The estimation is formulated as a nonlinear least squares (NLS) problem solved by applying numerical differentiation and parameterizing rotations in axis-angle form (Rodrigues). All these methods ignore modeling sensors' noise.

Camera-Manipulator calibration has been well studied for cameras attached to the End Effector (EE), usually referred to as eye-in-hand, and for fixed external cameras.

The work addressed in [11] estimates the pose of the camera in the EE while assuming that the EE pose can be perfectly measured using forward kinematics. In [12], they propose an online hand-eye calibration method, which eliminates the dependence on calibration objects. Both works parameterize rotation variables with unit quaternions constrained using a Lagrange multiplier. A few works have addressed the Camera-Manipulator calibration using on-manifold optimization, such as [13] and [14]. Still, all the aforementioned works on Camera-Manipulator calibration ignore modeling process noise and do not contemplate estimating manipulator intrinsic parameters, such as joint offsets.

The need to estimate the relative poses of several cameras is also a common problem. In fields like photogrammetry and computer vision, this is addressed by concurrently determining the camera poses and 3D features observed by these cameras. Known as bundle adjustment [15], this method sets a Gaussian distribution to each camera measurement, treating the system as a substantial nonlinear least squares problem for resolution.

In [16] they formulate a bundle adjustment that generalizes to combine various different sensors measurements to calibrate their locations and manipulators' joint offsets. Rotations are represented as axis-angle, and the problem is solved using the LM algorithm with numerical differentiation on a NLS problem. In the documentation of their open-source code[1], they remark that the treatment of rotation is not ideal: *"Ideally, we would completely avoid optimizing in a specific rotation space, and instead, operate directly on the rotations themselves, using incremental rotations."*

The extrinsic calibration of a visual-inertial sensor pair is also a very relevant topic in autonomous mobile robotics, as localization and mapping techniques require to accurate fuse inertial and exteroceptive measurements. The most popular estimation procedures rely on the execution of a predetermined trajectoy of the mobile base, combining their respective ego-motions in a nonlinear minimization problem.

In [17], the authors propose a robust algorithm to calibrate multi-sensor arrays using incremental motions. This method divides the problem in different steps: finding an initialization for each sensor pair, estimating first the rotational components, removing outliers, estimating the translation, and finally combining the readings from all the sensors.

Some authors use filter-based approaches (e.g., the Extended Kalman Filter), like [18], [19] among others. In [20], [21] the authors propose a solution based on incremental motions that optimize not only the relative sensor transform but also the Inertial Measurement Unit (IMU) true path, as it suffers from integration drift. Their minimization is based on NLS, parameterizing 3D poses with dual quaternions. In our previous work [22], we use the measurements of a closed Inertial Navigation System (INS) providing absolute pose measurements of the robot inertial frame, and build a pose-graph, which is more restrictive as it constraints consecutive incremental poses. However, one drawback of this method is that the uncertainty associated with consecutive incremental poses is unknown, as this information is not provided by the INS. In this work, we improve the visual-inertial calibration by preintegrating the raw data of the internal IMU sensor to build a Graph Simultaneous Localization and Mapping (SLAM)

[1] http://wiki.ros.org/pr2_calibration_estimation

with a correct propagation of uncertainties. This approach is based on recent work in our lab [23] that develops a general-purpose preintegration methodology for Graph SLAM problems. The methodology is tested using the raw data of the Girona1000 INS and proved to replicate the dead-reckoning performance of the commercial INS, with the advantage of opening a graph-SLAM framework for advanced applications.

B. Contribution

This work provides simulated and experimental results of a complete calibration procedure estimating not only the extrinsic inter-relationships between sensors (and the robot inertial frame) but also the intrinsic (joint offsets) parameters of the manipulators. We use factor graphs as a tool to model the optimization problem and exploit the properties of Lie theory to perform a true on-manifold optimization with a proper handling of sensor noise and uncertainties. A special effort has been made to detail the mathematical development to reach analytical expressions of the required derivatives of the different defined factors, which is essential for optimal estimators. Finally, the proposed inertial calibration methodology is fast enough to be solved online, thus enabling SLAM while also optimizing the poses of the exteroceptive sensors.

The remainder of the paper is organized as follows. Section II describes the Girona500 I-AUV and its equipment. Section III introduces the general problem formulation. In section IV, the geometric characteristics of the used sensors are described. Then, section V formulates the different factors used for the estimation procedure. The simulated and experimental results are presented in sections VI and VII. Finally, the drawn conclusions are summarized in section VIII.

II. Platform Description

The Girona500 [24] is a reconfigurable Autonomous Underwater Vehicle (AUV) designed and built by the Underwater Robotics Research Center[2] (CIRS) at the University of Girona. It is designed for a maximum operating depth of up to 500 meters.[3] The vehicle is composed of an aluminum frame that supports three torpedo-shaped hulls of 0.3 m in diameter and 1.5 m in length. The vehicle's overall dimensions are 1 m in height, 1 m in width, and 1.5 m in length, and it weighs less than 200 Kg in its basic configuration.

The Girona500 is equipped with typical navigation sensors (DVL, IMU, pressure gauge, and USBL). In addition to these sensors, almost half the volume of the lower hull is reserved for mission-specific payload.

This available payload can be used to enhance the Girona 500 into an I-AUV (see fig. 1).

The current intervention payload is equipped with two 6 DOF Bravo7 manipulators from Reach Robotics[4], one having a camera attached at its tip. Additionally, it features a custom-made 3D laser scanner [25] and a down-looking camera to provide a different perspective, particularly useful for tasks such as seabed mapping (which can be used to improve the vehicle's navigation) or object detection.

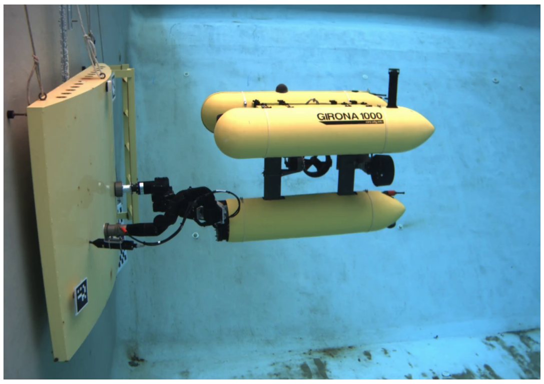


Fig. 1. The Girona1000 I-AUV performing inspection trials on a mockup structure in the CIRS facilities (Girona, Catalonia).

III. Problem Formulation

By measuring features with multiple sensors, we can build minimization policies that establish the relationship between these sensors in the robotic system. In our experiments, we rely on Charuco boards for detection. Charuco boards are traditional checkerboards that feature vacant white areas with Aruco markers [26]. The Aruco markers facilitate fast corner detection and allow to uniquely identify the corners. Thus, a Charuco board can be identified even with partial occlusion, and they don't suffer problems of rotation invariance while keeping the subpixel accuracy of traditional checkerboard patterns.

Suppose that two cameras, one held by a manipulator and the other fixed in the robot frame, detect a marker in the world (see fig. 2a). These combined (and synchronized) detections provide information to help estimate the cameras and manipulator poses, along with the manipulator joint offsets. Considering two manipulators, the same principles can be used in a dual-arm setup to estimate both EE camera poses, the relative pose between both manipulators, and their joint offsets (see fig. 2b). Another situation is when the pattern is held by one manipulator (see figs. 2c and 2d). Here, the relative pose between the pattern and the EE must also be estimated as it is not precisely known a priori.

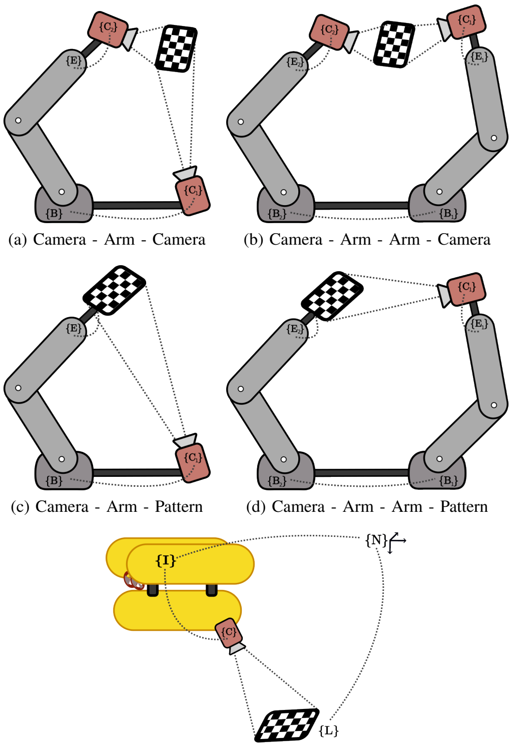
While these constraints allow to estimate all the inter-relationships between all cameras and manipulators in the robotic system, it does not relate them to the inertial frame of the robot. To establish a connection with the inertial frame of the robot, an additional step is needed in the calibration process. This involves incorporating information from inertial and motion sensors, such as accelerometers, gyroscopes, Doppler Velocity Log (DVL), or pressure sensors, and relating them with camera measurements.

Note that in section II, we introduce a custom-made 3D laser scanner based on a laser pointer reflected on a bi-axial MEMS mirror and a high-speed acquisition camera [25]. Since

[2] https://cirs.udg.edu

[3] Note that the Girona1000 has recently been developed at CIRS, an advanced version of the Girona500, rated for 1000 meters depth.

[4] https://reachrobotics.com/products/manipulators/reach-bravo/



(a) Camera - Arm - Camera

(b) Camera - Arm - Arm - Camera

(c) Camera - Arm - Pattern

(d) Camera - Arm - Arm - Pattern

(e) Camera - INS Calibration

Fig. 2. The set of constraints considered in this work

the intrinsic and local extrinsic parameters are calibrated in a closed-form optimization process [27], and the reference frame of the scanner is modeled at the camera frame, in this work, we treat it as a regular camera using only its raw images to calibrate its extrinsic parameters.

The optimization stage processes the defined minimization equations (or factors) encoded in a factor graph [5]. An example of a factor graph for the Camera - Arm - Camera (fig. 2a) estimation problem is shown in fig. 3. Measurements are assumed to be affected by a-priori characterized Gaussian noise.

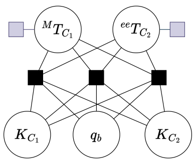


Fig. 3. Factor graph for the Camera - Arm - Camera calibration problem. 5 parameters can be calibrated: The pose of the fixed camera w.r.t the base link (${}^{M}T_{C_1}$), the pose of the EE camera (${}^{ee}T_{C_2}$), the joint offsets (q_b), and each camera intrinsic parameters. These variables are linked by a number of factors (black squares), each defining a constraint. Prior factors (grey) keep some variables close to their nominal values, modeling the "prior knowledge" of the variable state.

In this work, all noises and Jacobians are defined following Lie theory: defining probability distributions and adding perturbations on a tangent space of each group element. In this way, probability distributions and derivatives are consistent with the underlying geometry of the problem.

Lie theory is by no means simple, but in estimation for robotics, a small subset of basic principles have been proven to be enough to manage uncertainties, derivatives, and integrals with precision and ease. This mathematical corpus enables elegant and rigorous designs of optimal estimators, which is crucial to achieve precision, consistency, and stability of the solutions in modern estimation algorithms for robotics. The work done in [1*] provides a clear and useful small selection of material from Lie theory in the context of robotics and is our main reference for Lie theory. In this work, we develop (and simplify if possible) the Jacobians of all factors through the proper use of the chain rule.

IV. Sensors

For each sensor type in our robot, we need to formulate its geometric characteristics and Jacobians in order to lay the foundations to build factors for the optimization process. For the Girona500 we define three types of sensors: kinematic chains, cameras, and a Graph-based INSs.

A. Kinematic Chain

Let M be a manipulator, a kinematic chain consisting of consecutive actuated joints rigidly attached. We can then consider a set of joint variables q to be our measurement.

The forward kinematics model is computed as a sequence of rigid transformations, where each transform has two components; a fixed, link transformation, and an action, or motion, actuating 1 DOF:

$$Fk(q) = {}^{0}F_1 \ \dots \ {}^{i\text{-}1}F_i \ \dots {}^{n\text{-}1}F_n, \tag{1}$$

$${}^{i\text{-}1}F_i = T_i \cdot Exp(s_i \ q_i) \tag{2}$$

where $T \in SE(3)$ defines the link transform and $s = [\rho, \theta]^T \in \mathbb{R}^6$ is a twist defining the direction of the joint action. Thus, the action sq in the tangent space is transferred back to the group by the exponential map (Exp()). For instance, a revolute joint rotating around the z axis is modeled by $s = [0,0,0,0,0,1]^T$ or a prismatic joint sliding along the x axis is modeled by $s = [1,0,0,0,0,0]^T$. Note that a fixed joint can also be defined if $s = [0,0,0,0,0,0]^T$.

In this work, we assume that the measured joint angles may be offset due to inaccuracies during the assembly process. thus, we further define:

$$q_i = \hat{q}_i - q_{ib} \tag{3}$$

being $\hat{q}_i$ the true state of joint i, q_i the measured angle and q_{ib} the angle offset, or bias.

It's worth mentioning that we could potentially model additional kinematic biases, such as variations in link lengths. Nevertheless, these parameters are assumed to be accurate enough as they are precision machined.

The Jacobian of $Fk(q)$ is developed as follows. The full Jacobian is the concatenation of the partial Jacobians with respect to each joint:

$$\frac{\partial Fk(q)}{\partial q} = \left[\frac{\partial Fk(q)}{\partial q_0}, \frac{\partial Fk(q)}{\partial q_1} \dots \frac{\partial Fk(q)}{\partial q_m}\right] \tag{4}$$

For a specific joint i, we can split eq. (2) in three parts:

$$Fk(q) = \underbrace{{}^{0}F_1 \ \dots}_{A} \ {}^{i\text{-}1}F_i \ \underbrace{\dots {}^{n\text{-}1}F_n}_{B}, \tag{5}$$

and derive using the chain rule:

$$\frac{\partial Fk(q)}{\partial q_i} = \underbrace{\frac{\partial (A\ {}^{i\text{-}1}F_i)B}{\partial A\ {}^{i\text{-}1}F_i}}_{Ad^{-1}(B)} \underbrace{\frac{\partial A\ {}^{i\text{-}1}F_i}{\partial\ {}^{i\text{-}1}F_i}}_{I_6} \underbrace{\frac{\partial T_i Exp(s_i q_i)}{\partial Exp(s_i q_i)}}_{I_6} \cdot \underbrace{\frac{\partial Exp(s_i q_i)}{\partial s_i q_i}}_{J_r(s_i q_i)} \underbrace{\frac{\partial s_i q_i}{\partial q_i}}_{s_i} \tag{6}$$

$$= Ad(B^{\text{-}1}) \cdot J_r(s_i q_i) \cdot s_i \tag{7}$$

where:

$Ad(M)$ is the ***adjoint matrix*** of $M \in SE(3)$, and

$Jr()$ is the ***Right Jacobian*** of the group,

whose definitions can be found in [1*] (Appendix D).

Note that the adjoint matrix of a Lie group has two properties that are exploited in this work:

$$Ad(X^{\text{-}1}) = Ad(X)^{\text{-}1} \tag{8}$$

$$Ad(X \cdot Y) = Ad(X)\ Ad(Y) \tag{9}$$

where in both cases, the left parts of the equality are more efficient to compute than the right ones, since X and Y have group properties.

Finally, as $q = \hat{q} - q_b$, the Jacobian of $Fk(q)$ w.r.t q_b can be easily derived:

$$\frac{\partial Fk(q)}{\partial q_b} = -\frac{\partial Fk(q)}{\partial q} \tag{10}$$

Note that $n \neq m$ since fixed joints do not have an associated action.

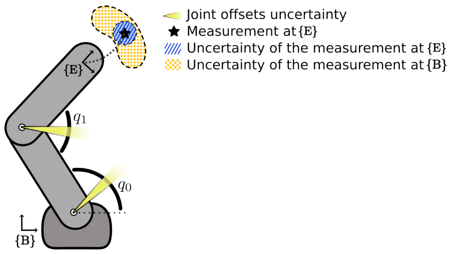

Joint offsets uncertainty
★ Measurement at {E}
▨ Uncertainty of the measurement at {E}
▨ Uncertainty of the measurement at {B}
{E}
q_1
q_0
{B}

Fig. 4. Joint offsets uncertainty affects the uncertainty of measurements made by sensors attached to the tip.

Having defined the forward kinematics and its Jacobian, it is now possible to formulate any sensor (or pattern) as being attached to the kinematic chain (see fig. 4).

B. Camera

Let C be a pinhole camera with a lens distortion model, and let $K_c = [f_x, f_y, c_x, c_y]$ be the camera's intrinsic parameters.

In order to find the pixel coordinates of a point ${}^{c}p$ in the camera frame, we start with a simple perspective projection, where the height of an object in the image is inversely proportional to its distance from the camera pupil:

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \frac{1}{{}^{c}p_z} \begin{bmatrix} {}^{c}p_x \\ {}^{c}p_y \end{bmatrix} \tag{11}$$

The *normalized* coordinates $p_n = [x_n, y_n]^T$ are then transformed by a distortion model dm, which describes the image distortion due to the optical system, which can be parameterized in different ways based on the lens type.

$$p_d = dm(p_n) \tag{12}$$

Finally, the normalized, distorted coordinates $p_d = [x_d, y_d]^T$ are related to the actual pixel coordinates $[u, v]$ by:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \tag{13}$$

The whole process of projecting a point ${}^{c}p$ to pixel coordinates can be formulated as

$$\begin{bmatrix} u \\ v \end{bmatrix} = proj({}^{c}p) = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} dm(p_n) + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \tag{14}$$

The Jacobian of $proj$ with respect to ${}^{c}p$ is given by:

$$\frac{\partial proj({}^{c}p)}{\partial {}^{c}p} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \frac{\partial dm(p_n)}{\partial p_n} \frac{\partial p_n}{\partial {}^{c}p} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \frac{\partial dm(p_n)}{\partial p_n} \frac{1}{{}^{c}p_z} \begin{bmatrix} 1 & 0 & -{}^{c}p_x \\ 0 & 1 & -{}^{c}p_y \end{bmatrix} \tag{15}$$

The two most common types of camera distortion models are the rectilinear lens distortion model and the fisheye lens distortion model. The image distortion model for rectilinear lenses takes into account radial distortion (barrel or pincushion) and tangential distortions (usually when the lens is not perfectly aligned parallel to the image plane). In contrast, the fisheye distortion generally includes only radial distortion. Still, its polynomial is a function of an angular distance from the center of perspective, rather than a linear distance in the image.

In this work, we consider a rectilinear lens distortion model with two radial distortion coefficients (k_1, k_2), and two tangential distortion coefficients $(p1, p2)$.

The equation relating the undistorted and distorted normalized image coordinates in this model is defined by the radial distortion as:

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \begin{bmatrix} x_n(1+k_1r^2+k_2r^4) \\ y_n(1+k_1r^2+k_2r^4) \end{bmatrix} \tag{16}$$

where $r^2 = x_n^2 + y_n^2$.
and the tangential distortion is represented as:

$$\begin{bmatrix} \partial x_t \\ \partial y_t \end{bmatrix} = \begin{bmatrix} 2p_1x_ny_n + p_2(r^2+2x^2) \\ 2p_2x_ny_n + p_1(r^2+2y^2) \end{bmatrix} \tag{17}$$

Combining both radial and tangential distortions, we can define:

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = dm\left(\begin{bmatrix} x_n \\ y_n \end{bmatrix}\right) = \begin{bmatrix} x_r \\ y_r \end{bmatrix} + \begin{bmatrix} \partial x_t \\ \partial y_t \end{bmatrix} \tag{18}$$

Being its Jacobian w.r.t the point ${}^{c}p$:

$$\frac{\partial dm(u,v)}{\partial u,v} = \frac{\partial rad(u,v)}{\partial u,v} + \frac{\partial tan(u,v)}{\partial u,v}, \tag{19}$$

$$\frac{\partial rad(u,v)}{\partial u,v} = \begin{bmatrix} g+2u^2(k_1+2k_2r^2), 2uv(k_1+2k_2r^2) \\ 2uv(k_1+k_22r^2), g+2v^2(k_1+k_22r^2) \end{bmatrix} \tag{19a}$$

$$\frac{\partial tan(u,v)}{\partial u,v} = \begin{bmatrix} 2(p_1v+3p_2u), 2(p_1u+p_2v) \\ 2(p_2v+p_1u), 2(p_2u+3p_1v) \end{bmatrix} \tag{19b}$$

where $g = 1 + k_1r^2 + k_2r^4$

A camera can be referenced to the base link of the system (the robot frame or manipulator base frame) or to a link of the manipulator (see fig. 5).

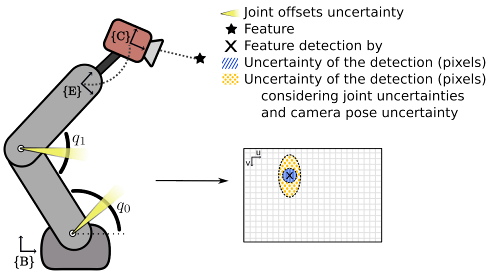

Joint offsets uncertainty
★ Feature
X Feature detection by
Uncertainty of the detection (pixels)
Uncertainty of the detection (pixels) considering joint uncertainties and camera pose uncertainty
{C}
{E}
q_1
q_0
{B}
u
v

Fig. 5. Joint offsets uncertainty and camera pose uncertainty influence the uncertainty of measurements in pixel space.

C. Graph-based inertial navigation system (G-INS)

An essential step of our approach is achieving an accurate calibration between the external sensors and the inertial reference frame of the robot. To do so, external sensor data is to be combined with inertial (or motion) information; that is, data collected from sensors such as accelerometers, gyroscopes (or magnetometer), DVL, or pressure gauge.

In previous work in our lab [23], the inertial and motion sensors of the robot are combined in a factor graph, modeling the robot's trajectory, as illustrated in fig. 6. Consecutive robot states are constrained by a pre-integration factor combining accelerometers, gyroscopes, and linear velocity measurements from the DVL. Additionally, each robot state, or keyframe, is constrained by DVL and pressure measurements. The problem also considers the earth's global effects of gravity and earth rotation and estimates sensor biases of accelerometers, gyroscopes, and DVL.

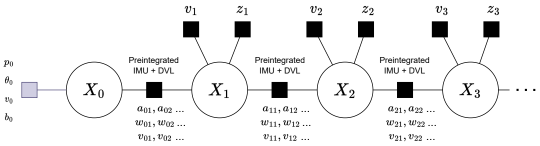


Fig. 6. Factor graph for robot navigation (dead reckoning). It applies preintegrated factors between keyframes, and depth and linear velocity prior factors. A robot state X_i is defined by a robot pose, linear velocity, and bias of the accelerometer, gyroscope, and DVL sensors.

The resulting factor graph can be extended with additional sensor information, including their relative poses w.r.t the robot's inertial frame, performing SLAM while also calibrating the extrinsic parameters of the sensors.

V. Factors

While the previous section was concerned with modeling the sensor measurements, now we discuss the factors that will be used to link the set of unknown (or uncalibrated) variables given all measurements available to us.

We classified the proposed factors into 5 groups:

- Camera - Arm - Camera (fig. 2a)
- Camera - Arm - Pattern (fig. 2c)
- Camera - Arm - Arm - Camera (fig. 2b)
- Camera - Arm - Arm - Pattern (fig. 2d)
- Camera - INS (fig. 2e)

Since we know the geometric properties of the pattern, measuring the pixel coordinates of the pattern corners ($z = [z_0, z_1, \cdots, z_n]^T$ where $z_i = [u_i, v_i]^T$) it is possible also to compute the 3D pose of the pattern or the 3D position of each corner.

In the following subsections, we will develop factors that relate measurements in 3D (pose to pose in $SE(3)$ and point to point in $\mathbb{R}^3$) and by projecting 3D points onto the camera image. Although the latest proved to achieve the best accuracy (see section VI-A4), developing all of them provides completeness to the work and serves three purposes:

- It allows the use of other feature detectors as a black box.
- Some factors serve as a basis of others.
- Projecting a point to the image plane requires better initialization to avoid back projection. Thus, it may be convenient to follow a two-step calibration, first relating measurements in 3D and then enhancing it through re-projection.

A. Camera - Arm - Camera (Pose - Pose)

Let's assume a manipulator M, a fixed camera C_1 mounted on the robot frame and a camera C_2 mounted on the manipulator's EE. Both C_1 and C_2 provide 3D measurements ($z \in SE(3)$) of the same landmark L. These measurements are denoted ${}^{C_1}T_L$ and ${}^{C_2}T_L$ respectively.

In this set-up, the system unknowns to be calibrated are:

- ${}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}$: The pose of the fixed camera (C_1) w.r.t the manipulator frame (M).
- ${}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}$: The pose of the EE camera (C_2) w.r.t the manipulator's EE frame.
- $\boldsymbol{q_b}$: The manipulator joints offsets.

We can relate these synchronized measurements with the following factor:

$$\eta_1 = Log({}^{C_1}T_L^{-1}\ {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}\ Fk(q - \boldsymbol{q_b})\ {}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}\ {}^{C_2}T_L) \in \mathbb{R}^6 \tag{20}$$

where $Log()$ is the **logarithmic map** of the $SE(3)$ group (see [1*] (Appendix D)).

Note that since the landmark is not assumed fixed, its location is not a variable to be estimated. This allows to move the robot or the pattern (or even use different ones), to obtain a rich and distributed set of measurements.

The Jacobians of η_1 w.r.t the unknowns are developed in the following subsections:

1) Jacobian w.r.t ${}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}$:

$$\begin{aligned}\frac{\partial \eta_1}{\partial {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}} &= \underbrace{\frac{\partial Log(Q_1)}{\partial Q_1}}_{J_r^{-1}(\eta_1)}\ \underbrace{\frac{\partial Q_2\ Q_3}{\partial Q_2}}_{Ad(Q_3)^{-1}}\ \underbrace{\frac{\partial Q_2}{\partial {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}}}_{I_6}\ \underbrace{\frac{\partial {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}}{\partial {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}}}_{-Ad({}^{M}T_{C_1})};\\ &= -J_r^{-1}(\eta_1)\ Ad(Q_3^{-1})\ Ad({}^{M}T_{C_1})\\ &= -J_r^{-1}(\eta_1)\ Ad(Q_3^{-1}\,{}^{M}T_{C_1})\end{aligned} \tag{21}$$

where

$Q_1 = {}^{C_1}T_L^{-1}\ {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}\ Fk(q - \boldsymbol{q_b})\ {}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}\ {}^{C_2}T_L$,

$Q_2 = {}^{C_1}T_L^{-1}\ {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}$, and

$Q_3 = Fk(q - \boldsymbol{q_b})\ {}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}\ {}^{C_2}T_L$.

However, note that in the final expression of the derivative, the inclusion of Q_2 is ruled out as it becomes unnecessary.

2) Jacobian w.r.t ${}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}$:

$$\begin{aligned}\frac{\partial \eta_1}{\partial {}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}} &= \underbrace{\frac{\partial Log(Q_1)}{\partial Q_1}}_{J_r^{-1}(\eta_1)}\ \underbrace{\frac{\partial Q_4\ Q_5}{\partial Q_5}}_{I_6}\ \underbrace{\frac{\partial Q_5}{\partial {}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}}}_{Ad({}^{C_2}T_L)^{-1}};\\ &= J_r^{-1}(\eta_1)\ Ad({}^{C_2}T_L^{-1})\end{aligned} \tag{22}$$

where

$Q_4 = {}^{C_1}T_L^{-1}\ {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}\ Fk(q - \boldsymbol{q_b})$ and

$Q_5 = {}^{C_1}T_L^{-1}\ {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}$.

3) Jacobian w.r.t $\boldsymbol{q_b}$:

$$\begin{aligned}\frac{\partial \eta_1}{\partial \boldsymbol{q_b}} &= \underbrace{\frac{\partial Log(Q_1)}{\partial Q_1}}_{J_r^{-1}(\eta_1)}\ \underbrace{\frac{\partial Q_4\ Q_5}{\partial Q_4}}_{Ad(Q_5)^{-1}}\ \underbrace{\frac{\partial Q_2\ Fk(q - \boldsymbol{q_b})}{\partial Fk(q - \boldsymbol{q_b})}}_{I_6}\ \underbrace{\frac{\partial Fk(q - \boldsymbol{q_b})}{\partial \boldsymbol{q_b}}}_{eq.\ (10)}\\ &= J_r^{-1}(\eta_1)\ Ad(Q_5^{-1})\ \frac{\partial Fk(q - \boldsymbol{q_b})}{\partial \boldsymbol{q_b}}\end{aligned} \tag{23}$$

Note that Q_2 and Q_4 are not required to be computed in any derivative. See eq. (24) to visualize how the residual (eq. (20)) is split to derive the Jacobian through the rule of chain.

$$\eta_1 = Log(\underbrace{\underbrace{\overbrace{{}^{C_1}T_L^{-1}\ {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}}^{Q_2}\ \mathrlap{\overbrace{\phantom{Fk(q - \boldsymbol{q_b})\ {}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}\ {}^{C_2}T_L}}^{Q_3}}Fk(q - \boldsymbol{q_b})}_{Q_4}\ \underbrace{{}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}\ {}^{C_2}T_L}_{Q_5}}_{Q_1}) \tag{24}$$

B. Camera - Arm - Camera (Point - Point)

In this situation, each camera (C_1 and C_2) provide measurements of a landmark L point ($\mathbb{R}^3$). These measurements are denoted ${}^{C_1}p_L$ and ${}^{C_2}p_L$, respectively.

Each pair of point measurements from both cameras is related with the following factor:

$$\eta_2 = {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}\ Fk(q - \boldsymbol{q_b})\ {}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}\ {}^{C_2}p_L - {}^{C_1}p_L \in \mathbb{R}^3 \tag{25}$$

Note that in our experiments, each detected landmark is a corner of the Charuco pattern. Thus, each image pair contains several measurements.

1) Jacobian w.r.t ${}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}$:

$$\begin{aligned}\frac{\partial \eta_2}{\partial {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}} &= \underbrace{\frac{\partial Q_1 {}^{C_2}p_L}{\partial Q_1}}_{Ja(Q_1,{}^{C_2}p_L)}\ \underbrace{\frac{\partial {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}\ Q_2}{\partial {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}}}_{Ad(Q_2)^{-1}}\ \underbrace{\frac{\partial {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}}{\partial {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}}}_{-Ad({}^{M}T_{C_1})}\\ &= -Ja(Q_1, {}^{C_2}p_L)\ Ad(Q_2^{-1})\ Ad({}^{M}T_{C_1})\\ &= -Ja(Q_1, {}^{C_2}p_L)\ Ad(\underbrace{Q_2^{-1}\ {}^{M}T_{C_1}}_{Q_1^{-1}})\\ &= -Ja(Q_1, {}^{C_2}p_L)\ Ad(Q_1^{-1})\end{aligned} \tag{26}$$

where

$Q_1 = {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}\ Fk(q - \boldsymbol{q_b})\ {}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}$,

$Q_2 = Fk(q - \boldsymbol{q_b})\ {}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}$, and

$Ja(T, p)$ is the jacobian of the $T \in SE(3)$ action on a point $p \in \mathbb{R}^3$ w.r.t T, which definition can be found in [1*] (Appendix D).

Note that we use the properties of the adjoint matrix (see eqs. (8, 9)) to simplify the Jacobian and make it more efficient.

2) Jacobian w.r.t ${}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}$:

$$\frac{\partial \eta_2}{\partial {}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}} = \underbrace{\frac{\partial Q_1 {}^{C_2}p_L}{\partial Q_1}}_{Ja(Q_1,{}^{C_2}p_L)}\underbrace{\frac{\partial Q_3 {}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}}{{}^{ee}\boldsymbol{T}_{\boldsymbol{C_2}}}}_{I_6} = Ja(Q_1, {}^{C_2}p_L) \tag{27}$$

where

$Q_3 = {}^{M}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}\ Fk(q - \boldsymbol{q_b})$.

3) Jacobian w.r.t $\boldsymbol{q_b}$:

$$\frac{\partial\eta_2}{\partial \boldsymbol{q_b}} = \overbrace{\frac{\partial Q_1{}^{C_2}p_L}{\partial Q_1}}^{Ja(Q_1,{}^{C_2}p_L)} \overbrace{\frac{\partial Q_3\ {}^{ee}\boldsymbol{T_{C_2}}}{\partial Q_3}}^{Ad({}^{ee}\boldsymbol{T_{C_2}})^{-1}} \cdot \underbrace{\frac{\partial {}^{M}\boldsymbol{T}^{-1}_{\boldsymbol{C_1}}\ Fk(q - \boldsymbol{q_b})}{\partial Fk(q - \boldsymbol{q_b})}}_{I_6} \frac{\partial Fk(q - \boldsymbol{q_b})}{\partial \boldsymbol{q_b}}$$

$$= Ja(Q_1, {}^{C_2}p_L)\ Ad({}^{ee}\boldsymbol{T}^{-1}_{\boldsymbol{C_2}})\ \frac{\partial Fk(q - \boldsymbol{q_b})}{\partial \boldsymbol{q_b}} \tag{28}$$

Note that after simplifying the Jacobians, only Q_1 is required. See eq. (29) to visualize how the residual (eq. (25)) is split.

$$\eta_2 = \underbrace{\underbrace{{}^{M}\boldsymbol{T}^{-1}_{\boldsymbol{C_1}}\ \overbrace{Fk(q - \boldsymbol{q_b})}^{Q_2}}_{Q_3}\ \overbrace{{}^{ee}\boldsymbol{T_{C_2}}}\ {}^{C_2}p_L}_{Q_1} - {}^{C_1}p_L \tag{29}$$

C. Camera - Arm - Camera Reprojection (Point - Pixel)

In this situation, the minimization function is performed by projecting the position of a point detected in one camera ($\mathbb{R}^3$) to the camera image of the other ($\mathbb{R}^2$) (see fig. 7).

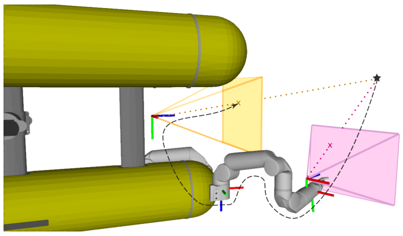


Fig. 7. The 3D point detected by each camera is reprojected in the camera image of the other to compute the error in pixel space

Lets consider that C_2 measures the landmark as a 3D point and C_1 in pixels ($z_2 = [u_L, v_L]^T$). We can relate these synchronized measurements with the following factor:

$$\eta_3 = proj({}^{M}\boldsymbol{T}^{-1}_{\boldsymbol{C_1}}\ Fk(q - \boldsymbol{q_b})\ {}^{ee}\boldsymbol{T_{C_2}}\ {}^{C_2}p_L) - \begin{bmatrix} u_L \\ v_L \end{bmatrix} \tag{30}$$

Note that this factor is an extension of eq. (25), being expected measurement projected onto the camera image. Thus, applying the rule of chain, the Jacobians can be easily derived:

$$\frac{\partial\eta_3}{\partial {}^{M}\boldsymbol{T_{C_1}}} = \underbrace{\frac{\partial proj(Q_1)}{\partial Q_1}}_{eq.(15)} \underbrace{\frac{\partial\eta_2}{\partial {}^{M}\boldsymbol{T_{C_1}}}}_{eq.(26)} \tag{31}$$

$$\frac{\partial\eta_3}{\partial {}^{ee}\boldsymbol{T_{C_2}}} = \frac{\partial proj(Q_1)}{\partial Q_1} \underbrace{\frac{\partial\eta_2}{\partial {}^{ee}\boldsymbol{T_{C_2}}}}_{eq.(27)} \tag{32}$$

$$\frac{\partial\eta_3}{\partial \boldsymbol{q_b}} = \frac{\partial proj(Q_1)}{\partial Q_1} \underbrace{\frac{\partial\eta_2}{\partial \boldsymbol{q_b}}}_{eq.(28)} \tag{33}$$

where

$$Q_1 = {}^{M}\boldsymbol{T}^{-1}_{\boldsymbol{C_1}}\ Fk(q - \boldsymbol{q_b})\ {}^{ee}\boldsymbol{T_{C_2}}\ {}^{C_2}p_L$$

In order to model the noise of both camera detections, the reprojection must be performed in both directions. Let Fk' be the reversed forward kinematics of M; that is, from the tip to the base. Then, a factor with the same structure can be used to formulate the opposite reprojection:

$$\eta_4 = proj({}^{ee}\boldsymbol{T}^{-1}_{\boldsymbol{C_2}}\ Fk'(q - \boldsymbol{q_b})\ {}^{M}\boldsymbol{T_{C_1}}\ {}^{C_1}p_L) - \begin{bmatrix} u_L \\ v_L \end{bmatrix} \in \mathbb{R}^2 \tag{34}$$

D. Camera - Arm - Pattern

In this situation, let's assume a manipulator M, a fixed camera C_1 mounted on the robot frame, and a pattern L being held by the manipulator's EE. The relative position of the pattern w.r.t the EE is not known a-priori.

In this setup, the system unknowns to be calibrated are:

- ${}^{M}\boldsymbol{T_{C_1}}$: The pose of the fixed camera (C_1) w.r.t the manipulator frame (M)
- ${}^{ee}\boldsymbol{T_L}$: The pose of the pattern L w.r.t the manipulator's EE frame
- $\boldsymbol{q_b}$: The manipulator joints offsets

which is analogous to the calibration problem for the Camera - Arm - Camera case, where C_2 becomes L. The factor using Pose measurements can be derived from eq. (20) as follows:

$$\eta_1 = Log({}^{C_1}T_L^{-1}\ {}^{M}\boldsymbol{T}^{-1}_{\boldsymbol{C_1}}\ Fk(q - \boldsymbol{q_b})\ \underbrace{{}^{ee}\boldsymbol{T_{C_2}}}_{{}^{ee}\boldsymbol{T_L}}\underbrace{{}^{C_2}T_L}_{I_6}),$$

$$\eta_1' = Log({}^{C_1}T_L^{-1}\ {}^{M}\boldsymbol{T}^{-1}_{\boldsymbol{C_1}}\ Fk(q - \boldsymbol{q_b})\ {}^{ee}\boldsymbol{T_L}). \tag{35}$$

Similarly, the factor considering point measurements can be derived from eq. (25). For each Charuco corner ${}^{L}p$ (which position is known w.r.t the pattern origin) and its detection ${}^{C_1}p_L \in \mathbb{R}^3$ by the C_1 camera, the factor can be formulated as:

$$\eta_2 = {}^{M}\boldsymbol{T}^{-1}_{\boldsymbol{C_1}}\ Fk(q - \boldsymbol{q_b})\ \underbrace{{}^{ee}\boldsymbol{T_{C_2}}}_{{}^{ee}\boldsymbol{T_L}}\underbrace{{}^{C_2}p_L}_{{}^{L}p} - {}^{C_1}p_L,$$

$$\eta_2' = {}^{M}\boldsymbol{T}^{-1}_{\boldsymbol{C_1}}\ Fk(q - \boldsymbol{q_b})\ {}^{ee}\boldsymbol{T_L}\ {}^{L}p - {}^{C_1}p_L \tag{36}$$

Or the latest can be projected onto the camera image:

$$\eta_3 = proj({}^{M}\boldsymbol{T}^{-1}_{\boldsymbol{C_1}}\ Fk(q - \boldsymbol{q_b})\ \underbrace{{}^{ee}\boldsymbol{T_{C_2}}}_{{}^{ee}\boldsymbol{T_L}}\underbrace{{}^{C_2}p_L}_{{}^{L}p}) - \begin{bmatrix} u_L \\ v_L \end{bmatrix},$$

$$\eta_3' = proj({}^{M}\boldsymbol{T}^{-1}_{\boldsymbol{C_1}}\ Fk(q - \boldsymbol{q_b})\ {}^{ee}\boldsymbol{T_L}\ {}^{L}p) - \begin{bmatrix} u_L \\ v_L \end{bmatrix} \tag{37}$$

E. Camera - Arm - Arm - Camera (Point - Point)

In this situation, a camera C_1 is mounted on the M_1 manipulator's EE, and a second camera C_2 is mounted on the M_2 manipulator's EE. Both C_1 and C_2 provide measurements of a landmark L point ($\mathbb{R}^3$). These measurements are denoted ${}^{C_1}p_L$ and ${}^{C_2}p_L$, respectively. Please note that the Pose-Pose factor is not formulated, as our tests indicated that it leads to inferior results compared the other approaches (refer to section VI for details).

In this set-up, the system unknowns to be calibrated are:

- ${}^{M_1}\boldsymbol{T}_{\boldsymbol{M_2}}$: The pose of the manipulator (M_1) w.r.t the manipulator (M_2).
- ${}^{ee1}\boldsymbol{T}_{\boldsymbol{C_1}}$: The pose of the EE camera (C_1) w.r.t the M_1 manipulator's EE frame.
- ${}^{ee2}\boldsymbol{T}_{\boldsymbol{C_2}}$: The pose of the EE camera (C_2) w.r.t the M_2 manipulator's EE frame.
- $\boldsymbol{q_{b1}}$: The M_1 manipulator joints offsets.
- $\boldsymbol{q_{b2}}$: The M_2 manipulator joints offsets.

Each pair of point measurements from both cameras is related with the following factor:

$$\begin{aligned}\eta_4 = {}^{ee1}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}\ Fk(q - \boldsymbol{q_{b1}})^{-1}\ {}^{M_1}\boldsymbol{T}_{\boldsymbol{M_2}}\ \cdot \\ Fk(q - \boldsymbol{q_{b2}})\ {}^{ee2}\boldsymbol{T}_{\boldsymbol{C_2}}\ {}^{C_2}p_L - {}^{C_1}p_L \in \mathbb{R}^3 \quad (38)\end{aligned}$$

1) Jacobian w.r.t ${}^{ee1}\boldsymbol{T}_{\boldsymbol{C_1}}$*:*

$$\begin{aligned}\frac{\partial\eta_4}{\partial{}^{ee1}\boldsymbol{T}_{\boldsymbol{C_1}}} &= \underbrace{\frac{\partial Q_1{}^{C_2}p_L}{\partial Q_1}}_{Ja(Q_1,{}^{C_2}p_L)}\ \underbrace{\frac{\partial{}^{ee1}T_{C_1}^{-1}\ Q_2}{\partial{}^{ee1}T_{C_1}^{-1}}}_{Ad(Q_2)^{-1}}\ \underbrace{\frac{\partial{}^{ee1}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}}{\partial{}^{ee1}\boldsymbol{T}_{\boldsymbol{C_1}}}}_{-Ad({}^{ee1}T_{C_1})}; \\ &= -Ja(Q_1, {}^{C_2}p_L)\ Ad(\underbrace{Q_2^{-1}\ {}^{ee1}T_{C_1}}_{Q_1^{-1}}) \\ &= -Ja(Q_1, {}^{C_2}p_L)\ Ad(Q_1^{-1}) \qquad (39)\end{aligned}$$

where

$Q_1 = {}^{ee1}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}\ Q_2$, and

$Q_2 = Fk(q - \boldsymbol{q_{b1}})^{-1}\,{}^{M_1}\boldsymbol{T}_{\boldsymbol{M_2}} Fk(q - \boldsymbol{q_{b2}})\ {}^{ee2}\boldsymbol{T}_{\boldsymbol{C_2}}$.

2) Jacobian w.r.t $\boldsymbol{q_{b1}}$*:*

$$\begin{aligned}\frac{\partial\eta_4}{\partial\boldsymbol{q_{b1}}} &= \overbrace{\frac{\partial Q_1{}^{C_2}p_L}{\partial Q_1}}^{Ja(Q_1,{}^{C_2}p_L)}\ \overbrace{\frac{\partial Q_3 Q_4}{\partial Q_3}}^{Ad(Q_4)^{-1}}\ \overbrace{\frac{\partial\boldsymbol{Q_3}}{\partial Fk(q - \boldsymbol{q_{b1}})^{-1}}}^{I_6}\ \cdot \\ &\quad \underbrace{\frac{\partial Fk(q - \boldsymbol{q_{b1}})^{-1}}{\partial Fk(q - \boldsymbol{q_{b1}})}}_{-Ad(Fk(q - \boldsymbol{q_{b1}}))}\ \underbrace{\frac{\partial Fk(q - \boldsymbol{q_{b1}})}{\partial\boldsymbol{q_{b1}}}}_{eq.\ (10)}; \\ &= -Ja(Q_1, {}^{C_2}p_L) Ad(\underbrace{Q_4^{-1} Fk(q - \boldsymbol{q_{b1}})}_{Q_2^{-1}}) \frac{\partial Fk(q - \boldsymbol{q_{b1}})}{\partial\boldsymbol{q_{b1}}} \\ &= -Ja(Q_1, {}^{C_2}p_L)\ Ad(Q_2^{-1})\ \frac{\partial Fk(q - \boldsymbol{q_{b1}})}{\partial\boldsymbol{q_{b1}}} \qquad (40)\end{aligned}$$

where

$Q_3 = {}^{ee1}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}\ Fk(q - \boldsymbol{q_{b1}})^{-1}$, and

$Q_4 = {}^{M_1}\boldsymbol{T}_{\boldsymbol{M_2}} Fk(q - \boldsymbol{q_{b2}})\ {}^{ee2}\boldsymbol{T}_{\boldsymbol{C_2}}$.

3) Jacobian w.r.t ${}^{M_1}\boldsymbol{T}_{\boldsymbol{M_2}}$*:*

$$\begin{aligned}\frac{\partial\eta_4}{\partial{}^{M_1}\boldsymbol{T}_{\boldsymbol{M_2}}} &= \underbrace{\frac{\partial Q_1{}^{C_2}p_L}{\partial Q_1}}_{Ja(Q_1,{}^{C_2}p_L)}\ \underbrace{\frac{\partial Q_5 Q_6}{\partial Q_5}}_{Ad(Q_6)^{-1}}\ \underbrace{\frac{\partial Q_3{}^{M_1}\boldsymbol{T}_{\boldsymbol{M_2}}}{\partial{}^{M_1}\boldsymbol{T}_{\boldsymbol{M_2}}}}_{I_6}; \\ &= Ja(Q_1, {}^{C_2}p_L) Ad(Q_6^{-1}) \qquad (41)\end{aligned}$$

where

$Q_5 = {}^{ee1}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}\ Fk(q - \boldsymbol{q_{b1}})^{-1}\ {}^{M_1}\boldsymbol{T}_{\boldsymbol{M_2}}$, and

$Q_6 = Fk(q - \boldsymbol{q_{b2}})\ {}^{ee2}\boldsymbol{T}_{\boldsymbol{C_2}}$.

4) Jacobian w.r.t $\boldsymbol{q_{b2}}$*:*

$$\begin{aligned}\frac{\partial\eta_4}{\partial\boldsymbol{q_{b2}}} &= \underbrace{\frac{\partial Q_1{}^{C_2}p_L}{\partial Q_1}}_{Ja(Q_1,{}^{C_2}p_L)}\underbrace{\frac{\partial Q_7{}^{ee2}T_{C_2}}{\partial Q_7}}_{Ad({}^{ee2}T_{C_2})^{-1}}\underbrace{\frac{\partial Q_5 Fk(q - \boldsymbol{q_{b2}})}{\partial Fk(q - \boldsymbol{q_{b2}})}}_{I_6}\underbrace{\frac{\partial Fk(q - \boldsymbol{q_{b2}})}{\partial\boldsymbol{q_{b2}}}}_{eq.\ (10)}; \\ &= Ja(Q_1, {}^{C_2}p_L)\ Ad({}^{ee2}T_{C_2}^{-1})\ \frac{\partial Fk(q - \boldsymbol{q_{b2}})}{\partial\boldsymbol{q_{b2}}} \qquad (42)\end{aligned}$$

where

$Q_7 = {}^{ee1}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}\ Fk(q - \boldsymbol{q_{b1}})^{-1}\ {}^{M_1}\boldsymbol{T}_{\boldsymbol{M_2}}\ Fk(q - \boldsymbol{q_{b2}})$.

5) Jacobian w.r.t ${}^{ee2}\boldsymbol{T}_{\boldsymbol{C_2}}$*:*

$$\begin{aligned}\frac{\partial\eta_4}{\partial{}^{ee2}\boldsymbol{T}_{\boldsymbol{C_2}}} &= \underbrace{\frac{\partial Q_1{}^{C_2}p_L}{\partial Q_1}}_{Ja(Q_1,{}^{C_2}p_L)}\underbrace{\frac{\partial Q_7\ {}^{ee2}T_{C_2}}{\partial{}^{ee2}T_{C_2}}}_{I_6}; \\ &= Ja(Q_1, {}^{C_2}p_L) \qquad (43)\end{aligned}$$

Note that after simplifying, only Q_1, Q_2, and Q_7 are required to be computed. See eq. (44) to visualize how the residual (eq. 38) is split to derive the Jacobian through the rule of chain.

$$\eta_4 = \mathrlap{\overbrace{\phantom{{}^{ee1}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}} Fk(q - \boldsymbol{q_{b1}})^{-1}\,{}^{M_1}\boldsymbol{T}_{\boldsymbol{M_2}} Fk(q - \boldsymbol{q_{b2}})}}^{Q_7}}\mathrlap{\overbrace{\phantom{{}^{ee1}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}} Fk(q - \boldsymbol{q_{b1}})^{-1}\,{}^{M_1}\boldsymbol{T}_{\boldsymbol{M_2}}}}^{Q_5}}\mathrlap{\overbrace{\phantom{{}^{ee1}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}} Fk(q - \boldsymbol{q_{b1}})^{-1}}}^{Q_3}}\underbrace{{}^{ee1}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}\,\underbrace{Fk(q - \boldsymbol{q_{b1}})^{-1}\,\underbrace{{}^{M_1}\boldsymbol{T}_{\boldsymbol{M_2}}\,\underbrace{Fk(q - \boldsymbol{q_{b2}}){}^{ee2}\boldsymbol{T}_{\boldsymbol{C_2}}}_{Q_6}}_{Q_4}}_{Q_2}}_{Q_1}\ \ldots \qquad (44)$$

F. Camera - Arm - Arm - Camera Reprojection (Point - Pixel)

As in section V-C, the minimization function can be modeled by projecting the position of a point detected in one camera ($\mathbb{R}^3$) to the camera image of the other ($\mathbb{R}^2$).

Considering C_2 measures the landmark as a 3D point and C_1 in pixels ($z_2 = [u_L, v_L]^T$). We can relate these synchronized measurements with the following factor:

$$\begin{aligned}\eta_5 = proj({}^{ee1}\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{-1}}\ Fk(q - \boldsymbol{q_{b1}})^{-1}\ {}^{M_1}\boldsymbol{T}_{\boldsymbol{M_2}}\ \cdot \\ Fk(q - \boldsymbol{q_{b2}})\ {}^{ee2}\boldsymbol{T}_{\boldsymbol{C_2}}\ {}^{C_2}p_L) - \begin{bmatrix} u_L \\ v_L \end{bmatrix} \in \mathbb{R}^2.\end{aligned}$$

Therefore, the jacobians of η_5 w.r.t the unknowns can be easily derived through the rule of chain, as shown in section V-C.

G. *Camera - Arm - Arm - Pattern*

In this situation, let's assume a manipulator M, a fixed camera C_1 mounted on the robot frame, and a pattern L being held by the manipulator's EE. The relative position of the pattern w.r.t the EE is not known a-priori.

In this setup, the system unknown ${}^{ee2}T_{C_2}$ is replaced by the pattern pose w.r.t the M_2 EE (${}^{ee2}T_L$). Therefore, for each Charuco corner ${}^{C_2}T_L$, (which position is known w.r.t the pattern origin) the factor from sections V-E and V-F can be reformulated as:

$$\begin{aligned}\eta_4' = {}^{\mathbf{ee1}}\mathbf{T}_{\mathbf{C_1}}^{\textbf{-1}}\ Fk(q\,\text{-}\,\mathbf{q_{b1}})^{\text{-}1}\ {}^{\mathbf{M_1}}\mathbf{T}_{\mathbf{M_2}}\ \cdot \\ Fk(q\,\text{-}\,\mathbf{q_{b2}})\ {}^{\mathbf{ee2}}\mathbf{T}_{\mathbf{L}}\ {}^{L}p - {}^{C_1}p_L \in \mathbb{R}^3 \end{aligned} \quad (45)$$

$$\begin{aligned}\eta_5' = proj({}^{\mathbf{ee1}}\mathbf{T}_{\mathbf{C_1}}^{\textbf{-1}}\ Fk(q\,\text{-}\,\mathbf{q_{b1}})^{\text{-}1}\ {}^{\mathbf{M_1}}\mathbf{T}_{\mathbf{M_2}}\ \cdot \\ Fk(q\,\text{-}\,\mathbf{q_{b2}})\ {}^{\mathbf{ee2}}\mathbf{T}_{\mathbf{L}}\ {}^{L}p) - \begin{bmatrix} u \\ v \end{bmatrix} \in \mathbb{R}^2 \end{aligned} \quad (46)$$

H. *Robot Inertial frame - Camera (Point - Point)*

Let's assume a fixed camera C mounted on the robot frame, denoted as R. Note that the robot origin is set to the IMU frame to avoid frame transform of high-rate IMU measurements as suggested in [23]. C can provide measurements of a fixed landmark L located in the world frame[5] W. The robot inertial sensors cannot measure directly L. Thus, this estimation must be based on the relative ego-motion of the sensors.

Here, we build a NLS problem extending our graph-based INS to a full SLAM problem by constraining the robot trajectory with detections of a landmark, considered fixed in the world, as illustrated in fig. 8. In this setup, the unknowns that must be calibrated are:

- ${}^{\mathbf{R}}\mathbf{T}_{\mathbf{C_1}}$: The pose of the fixed camera C w.r.t the inertial frame.
- ${}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}}$: The pose of the landmark L w.r.t the world frame.
- ${}^{\mathbf{W}}\mathbf{T}_{\mathbf{R}}\boldsymbol{(t)}$: The trajectory of the robot w.r.t the world frame.

Additionally, along with the robot trajectory, the optimization problem also estimates the history of robot velocities and sensor biases as they are encoded in the state X.

Here, we consider the camera to detect each Charuco corner as a 3D point. However, treating each point as an (independent) landmark would allow the intrinsic properties of the pattern to be violated. Instead of defining relative constraints between each point (which are not guaranteed to be kept), we can implicitly constraint the pattern shape since it is known a-priori and introduce only the 3D pose of the landmark:

$${}^{W}p_i = {}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}}\ {}^{L}p_i \quad (47)$$

where ${}^{L}p_i$ is the *i-th* corner position in the pattern frame, which is constant.

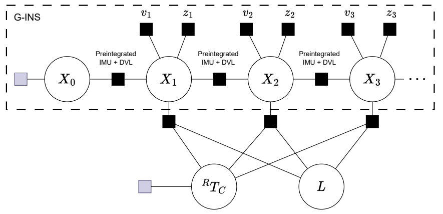


Fig. 8. An example of the inertial calibration problem. The robot trajectory of the graph-based INS (fig. 6) is constrained with exteroceptive measurements of a camera C. Note that multiple landmarks can be mapped, and each keyframe may have several factors.

For each detected corner, we can define the following factor:

$$\eta_7 = {}^{\mathbf{R}}\mathbf{T}_{\mathbf{C_1}}^{\textbf{-1}}\ {}^{\mathbf{W}}\mathbf{T}_{\mathbf{R}}^{\textbf{-1}}\ {}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}}\ {}^{L}p\ -\ {}^{C_1}p_L \in \mathbb{R}^3 \quad (48)$$

1) Jacobian w.r.t ${}^{\mathbf{W}}\mathbf{T}_{\mathbf{R}}$*:*

$$\begin{aligned}\frac{\partial \eta_7}{\partial {}^{\mathbf{W}}\mathbf{T}_{\mathbf{R}}} &= \underbrace{\frac{\partial Q_1({}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}}{}^{L}p)}{\partial Q_1}}_{Ja(Q_1,{}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}})} \underbrace{\frac{\partial Q_1}{\partial {}^{\mathbf{W}}\mathbf{T}_{\mathbf{R}}^{\textbf{-1}}}}_{I_6} \underbrace{\frac{\partial {}^{\mathbf{W}}\mathbf{T}_{\mathbf{R}}^{\textbf{-1}}}{\partial {}^{\mathbf{W}}\mathbf{T}_{\mathbf{R}}}}_{-Ad({}^{W}T_R)} \\ &= -Ja(Q_1, {}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}})Ad({}^{\mathbf{W}}\mathbf{T}_{\mathbf{R}}) \end{aligned} \quad (49)$$

where
$Q_1 = {}^{\mathbf{R}}\mathbf{T}_{\mathbf{C_1}}^{\textbf{-1}}\ {}^{\mathbf{W}}\mathbf{T}_{\mathbf{R}}^{\textbf{-1}}$

2) Jacobian w.r.t ${}^{\mathbf{R}}\mathbf{T}_{\mathbf{C_1}}$*:*

$$\begin{aligned}\frac{\partial \eta_7}{\partial {}^{\mathbf{R}}\mathbf{T}_{\mathbf{C_1}}} &= \underbrace{\frac{\partial Q_1({}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}}{}^{L}p)}{\partial Q_1}}_{Ja(Q_1,{}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}})} \underbrace{\frac{\partial Q_1}{\partial {}^{\mathbf{R}}\mathbf{T}_{\mathbf{C_1}}^{\textbf{-1}}}}_{Ad({}^{W}T_R^{\textbf{-1}})^{\textbf{-}1}} \underbrace{\frac{\partial {}^{\mathbf{R}}\mathbf{T}_{\mathbf{C_1}}^{\textbf{-1}}}{\partial {}^{\mathbf{R}}\mathbf{T}_{\mathbf{C_1}}}}_{-Ad({}^{R}T_{C_1})} \\ &= -Ja(Q_1, {}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}})\ Ad({}^{\mathbf{W}}\mathbf{T}_{\mathbf{R}}{}^{\mathbf{R}}\mathbf{T}_{\mathbf{C_1}}) \end{aligned} \quad (50)$$

3) Jacobian w.r.t ${}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}}$*:*

$$\frac{\partial \eta_7}{\partial {}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}}} = \frac{\partial Q_1({}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}}{}^{L}p)}{\partial {}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}}{}^{L}p} \frac{\partial {}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}}{}^{L}p}{\partial {}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}}} \quad (51)$$

I. *Robot Inertial frame - Camera (Point - Pixel)*

Similarly, the previous factor can be modeled in the camera image:

$$\eta_8 = proj({}^{\mathbf{R}}\mathbf{T}_{\mathbf{C_1}}^{\textbf{-1}}\ {}^{\mathbf{W}}\mathbf{T}_{\mathbf{R}}^{\textbf{-1}}\ {}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}}\ {}^{L}p)\ -\ \begin{bmatrix} u \\ v \end{bmatrix} \in \mathbb{R}^2 \quad (52)$$

As in section V-C, here the Jacobians can be easily derived from eqs.(50), (49), and (51):

$$\frac{\partial \eta_8}{\partial {}^{\mathbf{W}}\mathbf{T}_{\mathbf{R}}} = \frac{\partial proj(Q_1)}{\partial Q_1} \frac{\partial \eta_7}{\partial {}^{\mathbf{W}}\mathbf{T}_{\mathbf{R}}} \quad (53)$$

$$\frac{\partial \eta_8}{\partial {}^{\mathbf{R}}\mathbf{T}_{\mathbf{C_1}}} = \frac{\partial proj(Q_1)}{\partial Q_1} \frac{\partial \eta_7}{\partial {}^{\mathbf{R}}\mathbf{T}_{\mathbf{C_1}}} \quad (54)$$

$$\frac{\partial \eta_8}{\partial {}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}}} = \frac{\partial proj(Q_1)}{\partial Q_1} \frac{\partial \eta_7}{\partial {}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}}} \quad (55)$$

where
$Q_1 = {}^{\mathbf{R}}\mathbf{T}_{\mathbf{C_1}}^{\textbf{-1}}\ {}^{\mathbf{W}}\mathbf{T}_{\mathbf{R}}^{\textbf{-1}}\ {}^{\mathbf{W}}\mathbf{T}_{\mathbf{L}}\ {}^{L}p$

[5] In underwater robotics, the most common tangent plane coordinate system is the North-East-Down (NED).

VI. SIMULATED RESULTS

This section provides a detailed overview of the outcomes obtained from the implementation of the proposed calibration method on a simulated environment using the open-source Stonefish simulator [28].

The simulated environment provides a controlled setting for evaluating the proposed calibration method's performance and assessing its accuracy and effectiveness against known ground truth values, initializing the calibration far from the nominal values.

To emulate the real set-up, the simulated Girona500 features two Reach Bravo7 manipulators, one having a camera attached to its last link, one down-looking camera, and one forward-looking camera, which is replaced with our custom 3D laser scanner in the real set-up. As navigation sensors, it includes an IMU, a pressure sensor, and a DVL.

A. *Camera - Arm - Camera*

In this experiment, we compare the camera-arm-camera calibration in the three modalities explained in sections V-A to V-C. The loop is closed with the forward-looking camera, the left arm, and its EE camera detecting a pattern in the environment, as illustrated in fig. 9. The calibration procedure is tested with a dataset of 21 samples.

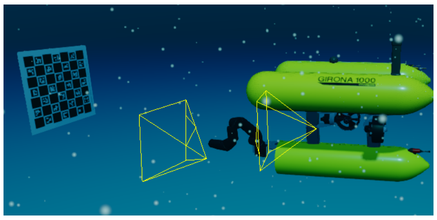


Fig. 9. The simulated Girona1000 with a forward-looking camera and an EE camera detecting a pattern.

The known, ground truth parameters are[6]:

$$
\begin{aligned}
{}^{M}\breve{T}_{C_1} &= \begin{cases} pos: & (-0.5, 0.2, -0.1) \\ rpy: & (0.0, 88.59, 0.0) \end{cases} \\
{}^{ee}\breve{T}_{C_2} &= \begin{cases} pos: & (0.15, 0.1, 0.1) \\ rpy: & (0.0, -28.6, 0.0) \end{cases} \\
q_b &= [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]^T
\end{aligned}
$$

The initial parameters for the calibration are set to:

$$
\begin{aligned}
{}^{M}T_{C_1} &= \begin{cases} pos: & (-0.41, 0.33, 0.0) \\ rpy: & (3.817, 73.071, 7.253) \end{cases} \\
{}^{ee}T_{C_2} &= \begin{cases} pos: & (0.25, 0.0, -0.05) \\ rpy: & (9.81, -17.5, 9.33) \end{cases} \\
q_b &= [0.0, 4.864, 3.145, 6.291, -9.145, 0.0]^T,
\end{aligned}
$$

with the uncertainty of each camera pose set to $\sigma = 1.5m$ in position for each axis and $\sigma = 1rad$ in orientation for each axis, and the uncertainty of the joint state bias set to $\sigma = 0.2rad$ for all joints. However, the biases of the first and last joints remain fixed, i.e., they are not subject to optimization. This restriction is imposed due to the redundancy of each joint, which is concatenated by a variable in $SE(3)$, rendering them non-observable. The initial information of the estimated variables is encoded in the graph through prior factors.

The measurement uncertainties will be defined for each case in the appropriate units.

1) Pose - Pose (sec. V-A): In this experiment, the measurement uncertainty, defined in the logarithmic map of $SE(3)$, is set to $\sigma = 0.04\ rad$ for each orientation axis, $\sigma = 0.1\ m$ for x and y position, and $\sigma = 0.2\ m$ for z position. The reason behind this is that flat pattern detectors are known to be more uncertain in measuring the relative distance. The orientation is assumed to be more precise since the pattern has 36 corners for interpolation. However, note that all these decisions are subjective, as it is not natural to argue on precision in terms of relative poses between two detections. Moreover, each detection is treated equally, although it should not be the case (for instance, detecting a nearby pattern is inherently more accurate than detecting one that is far away). These details condition the performance of the estimated solution, as detailed in section VI-A4.

The built graph had 24 nodes, including prior factors, and took 296 ms[7] for the calibration procedure to converge. To assess the accuracy of the solution, we compare the outcome of the solution with the ground truth, both defined in their tangent spaces[8]:

$$
\begin{aligned}
{}^{M}T_{C_1}\ er. &= \begin{cases} pos: & (0.0063, 0.0063, 0.0108) \\ rot: & (0.0017, 0.0004, -0.0018) \end{cases} \\
{}^{ee}T_{C_2}\ er. &= \begin{cases} pos: & (0.0058, 0.0032, 0.0086) \\ rot: & (0.0001, 0.0005, -0.0024) \end{cases} \\
q_b\ er. &= [0.0, 1.864, -1.207, -0.108, -0.045, 0.0]^T.
\end{aligned}
$$

Please note that the error (ground truth - result), is calculated as ($Log(\breve{T}^{-1} \cdot T^{*}$) for 3D poses, where $\breve{T}$ is the ground truth pose and T^{*} the optimized one. Hence, the error is calculated in the reference frame of the ground truth. Errors in position are in the order of a few millimeters, the z-axis being even less accurate.

2) Point - Point (sec. V-B): In this experiment, the measurement uncertainty, in 3D position, is set to $\sigma = 0.1\ m$ for x and y axis and $\sigma = 0.2\ m$ for z axis, the same uncertainty defined in the previous experiment.

The built graph had 729 nodes and took 627 ms for the calibration procedure to converge. Please note that for each pair of acquired images, there is one node for each detected corner in the pattern. The resulting error with respect to the ground truth is:

[6] For enhanced clarity, pose orientation is conveyed in Euler angles (in degrees).

[7] On an Intel Core i7-9700K CPU @ 3.60GHz

[8] Pose orientation error is represented in Rodrigues form (tangent space)

$$
\begin{aligned}
{}^{M}T_{C_1}\ er. &= \begin{cases} pos: & (-0.0010, 0.0009, 0.0013) \\ rot: & (0.0012, 0.0015, -7.e^{\text{-}6}) \end{cases} \\
{}^{ee}T_{C_2}\ er. &= \begin{cases} pos: & (-0.001, 0.0004, -0.0004) \\ rot: & (0.0004, 0.001, 5.1e^{\text{-}5}) \end{cases} \\
q_b\ er. &= [0.0, 2.2e^{\text{-}4}, -3.2e^{\text{-}5}, 1.5e^{\text{-}5}, -4.8e^{\text{-}5} 0.0]^T
\end{aligned}
$$

In this experiment, errors in position are reduced significantly. One possible explanation is the fact that the uncertainty of the measurement is defined only for points ($\mathbb{R}^3$ and not full poses $SE(3)$, while information is retained as each pair of images generates multiple factors that also constrain the rotation of the measurements. Consequently, we no longer operate under the assumption that the measurement errors in position and orientation are uncorrelated.

3) Point - Pixels (sec. V-C): Finally, we provide results based on point reprojection onto the camera images. The measurement uncertainty, defined in pixels $[u, v]$, is set to $\sigma = 3\ px$ for both cameras.

The built graph had 1455 nodes and took 1.117 s for the calibration procedure to converge, with errors:

$$
\begin{aligned}
{}^{M}T_{C_1}\ er. &= \begin{cases} pos: & (-8.1e^{\text{-}4}, 6.5e^{\text{-}4}, 5.2e^{\text{-}4}) \\ rot: & (9.7e^{\text{-}4}, 1.4e^{\text{-}3}, 1.8e^{\text{-}4}) \end{cases} \\
{}^{ee}T_{C_2}\ er. &= \begin{cases} pos: & (-2.5e^{\text{-}4}, 3.8Ze^{\text{-}4}, -2.2e^{\text{-}4}) \\ rot: & (8.1e^{\text{-}4}, 6.7e^{\text{-}4}, 1.1e^{\text{-}4}) \end{cases} \\
q_b\ er. &= [0.0, 7.3e^{\text{-}4}, -7.5e^{\text{-}4}, -1.2e^{\text{-}5}, 4.1e^{\text{-}5}, 0.0]^T
\end{aligned}
$$

Here, the errors in position are reduced to below millimeter accuracy. An alternative method to assess the results, without relying on ground truth information, can be achieved by reprojecting the detected 3D points from one camera onto the image of the other, followed by calculating the error in pixels. The visual representation of this error is illustrated in Figure 10, which offers valuable insight into the effectiveness of the estimation procedure. The reprojection error is provided first using the initial values of the estimated variables and then using the optimized ones. Moreover, fig. 11 visually represents the 3D pattern detections with initial and estimated parameters.

4) Comparisson: The accuracy of the results, all derived from the same dataset, consistently improves with each successive experiment (refer to fig. 12). Projecting 3D points onto the camera leverages the inherent characteristics of the measurements, providing the advantage of easier quantification of measurement uncertainties. This is particularly advantageous because, in general, the covariance matrix is defined solely on the diagonal, assuming independence between variables. By focusing on the 2D projection, the assumption of independence is simplified to only encompass the u and v pixel coordinates. In contrast, dealing with complete 3D poses, derived as a function of u and v pixel coordinates, introduces complexity due to the interdependence of these variables. As a result, the calibration process becomes more practical and more interpretable, contributing to the observed enhancement in accuracy across successive calibration experiments.

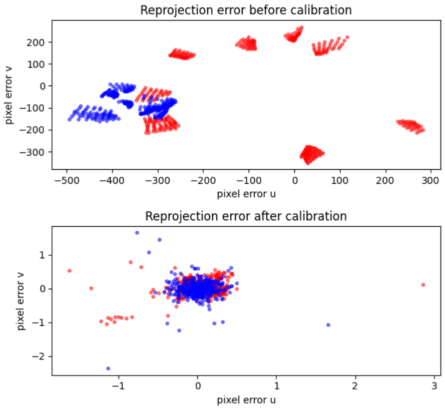


Fig. 10. Reprojection error in pixels by the fixed camera (red) and end-effector camera (blue). Note that the scales are different in the top and bottom plots

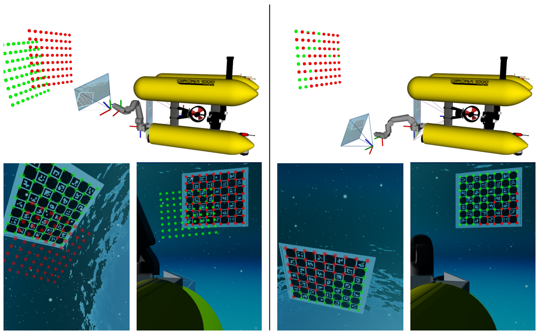


Fig. 11. The detected pattern by each camera using initial parameters (left) and the calibrated ones (right). The pattern detections are marked in green for the end-effector camera and red for the fixed camera.

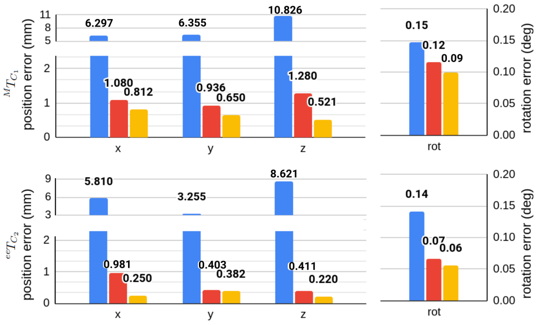


Fig. 12. Errors with respect to the ground truth values. In blue using Pose-Pose factors, in red using Point-Point factors, and in yellow, using Point-Pixel factors. The top plot corresponds to ${}^{M}T_{C_1}$, and the bottom plot corresponds to ${}^{ee}T_{C_2}$. Note that the plot has a discontinuous y-axis for a clearer visualization of the data.

However, it is crucial to highlight that although the projection method yields more accurate results, the initial parameters must be measured more carefully; otherwise, the 3D points may project behind the camera, introducing chirality issues. This issue does not arise in the other two methods. Consequently, the calibration process can be approached in two steps: initializing with measurements of 3D points (or poses) to achieve a near-optimal solution, followed by refining the calibration using the projection method. This methodology ensures a robust and accurate estimation, addressing concerns related to the precision of initial parameters and the solution's optimality.

B. Camera - Arm - Pattern

In this experiment, a Charuco board is attached to the EE of a manipulator, and it is detected by the forward-looking camera (see fig. 13). The calibration procedure is tested with a dataset of 31 samples. The Pose - Pose calibration is not considered here as it leads to similar inaccurate results as in the previous experiments.

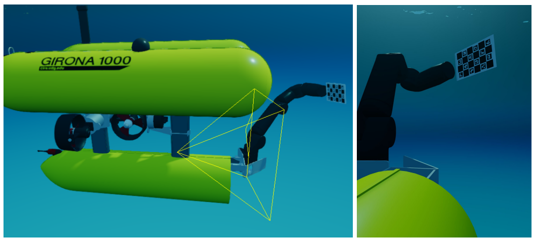


Fig. 13. The simulated Girona500 holding a Charuco pattern, and an image rendered from the forward-looking camera.

The known, ground truth parameters are:

$$
\begin{aligned}
{}^{M}\breve{T}_{C_1} &= \begin{cases} pos: & (-0.5, 0.2, -0.1) \\ rpy: & (0, 90, 0.0) \end{cases} \\
{}^{ee}\breve{T}_{L} &= \begin{cases} pos: & (-0.15, 0.0, 0.25) \\ rpy: & (-1.65, 85, 175) \end{cases} \\
q_b &= [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]^T
\end{aligned}
$$

The initial parameters for the calibration are set to:

$$
\begin{aligned}
{}^{M}T_{C_1} &= \begin{cases} pos: & (-0.36, 0.33, 0.0) \\ rpy: & (18, 72, 13) \end{cases} \\
{}^{ee}T_{L} &= \begin{cases} pos: & (0.25, -0.10, -0.05) \\ rpy: & (143, 86, 10) \end{cases} \\
q_b &= [0.0, 0.05, 0.02, 0.04, -0.02, 0.0]^T
\end{aligned}
$$

1) Point - Point (sec. V-B): In this experiment, the measurement uncertainty, in 3D position, is set to $\sigma = 0.1\ m$ for x and y axis and $\sigma = 0.2\ m$ for z axis, as in previous experiments. The built graph had 699 nodes and took 724 ms for the calibration procedure to converge, with error:

$$
\begin{aligned}
{}^{M}T_{C_1}\ er. &= \begin{cases} pos: & (-2.4e^{-4}, -5.2e^{-4}, 6.9e^{-4}) \\ rot: & (6.7e^{-5}, 3.3e^{-4}, 1.5e^{-4}) \end{cases} \\
{}^{ee}T_{L}\ er. &= \begin{cases} pos: & (5.0e^{-4}, 4.5e^{-4}, -5.4e^{-4}) \\ rot: & (2.8e^{-3}, -1.0e^{-3}, -5.9e^{-3}) \end{cases} \\
q_b\ er. &= [0.0, -5.1e^{-4}, -3.1e^{-4}, 6.3e^{-4}, 1.6e^{-3}, 0.0]^T
\end{aligned}
$$

2) Point - Pixels (sec. V-C): In this experiment, the measurement uncertainty, in image coordinates, was set to $\sigma = 3\ px$ for both u and v coordinates. The built graph had 418 nodes and took 582 ms for the calibration procedure to converge, being the error with respect to the ground truth:

$$
\begin{aligned}
{}^{M}T_{C_1}\ er. &= \begin{cases} pos: & (1.2e^{-5}, -1.2e^{-4}, 1.5e^{-4}) \\ rot: & (3.1e^{-4}, 4.6e^{-5}, 9.8e^{-5}) \end{cases} \\
{}^{ee}T_{L}\ er. &= \begin{cases} pos: & (1.2e^{-4}, -5.4e^{-5}, -4.6e^{-4}) \\ rot: & (2.2e^{-4}, 5.6e^{-5}, 5.6e^{-4}) \end{cases} \\
q_b\ er. &= [0.0, -1.2e^{-5}, -4.3e^{-5}, -1.4e^{-4}, 3.1e^{-4}, 0.0]^T
\end{aligned}
$$

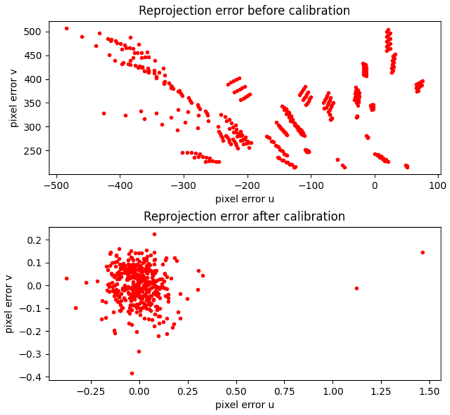


Fig. 14. Pattern reprojection error in pixels. Note that the scales are different in the top and bottom plots

Figure 14 depicts the reprojection error of the pattern 3D points in the camera image, with the initial parameters and after the calibration procedure. Finally, both Point-Point and Point-Pixels methods are compared in fig. 15, revealing again that performing the measurements in the minimum representation improves the accuracy of the estimation.

C. Reversed Camera-Arm-Pattern (eye-in-hand)

Suppose we assume both the robot and the pattern to be static in the N frame. In that case, we can formulate the calibration problem as a Reverse Camera-Arm-Pattern, where the pattern is considered fixed w.r.t the manipulator's base link and the camera fixed in the manipulator's EE. Hence, the same formulation can be used with the kinematic chain reversed.

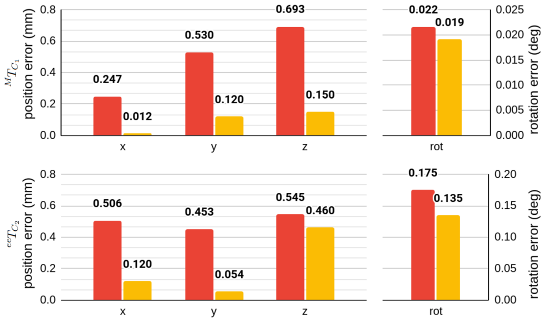


Fig. 15. Error with respect to the ground truth values. In red using Point-Point factors and in yellow, using Point-Pixel factors. The top plot corresponds to ${}^{M}T_{C_1}$ and the bottom plot corresponds to ${}^{ee}T_{C_2}$.

This type of calibration is usually referred to as eye-in-hand calibration.

For this experiment, we took a subset of the data acquired for the Camera-Arm-Camera experiment where the pattern and the robot are static (10 samples). The measurement uncertainty, in image coordinates, was set to $\sigma = 3\ px$ for both u and v coordinates. The built graph had 387 nodes and took 489 ms for the calibration procedure to converge with an error of:

$$
\begin{aligned}
{}^{ee}T_{C_2}\ er. &= \begin{cases} pos: & (-3.5e^{-5}, 1.4e^{-4}, 1.3e^{-4}) \\ rot: & (4.4e^{-4}, 5.7e^{-4}, 6.1e^{-6}) \end{cases} \\
{}^{M}T_{L}\ er. &= \begin{cases} pos: & (1.2e^{-5}, -1.2e^{-4}, 1.5e^{-4}) \\ rot: & (3.1e^{-4}, 4.6e^{-4}, 9.8e^{-5}) \end{cases} \\
q_b\ er. &= [0.0, -1.2e^{-5}, -4.3e^{-5}, -1.4e^{-4}, 3.1e^{-4}, 0.0]^T
\end{aligned}
$$

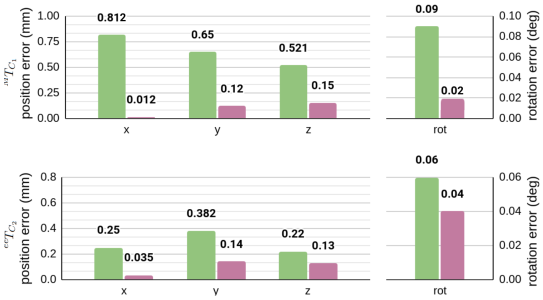


Fig. 16. Comparison of the accuracy of the Camera - Arm - Camera (green) versus the Camera - Arm - Pattern (magenta) methods. The top plot corresponds to the ${}^{M}T_{C_1}$ 3D pose, and the bottom plot corresponds to the ${}^{ee}T_{C_2}$ 3D pose.

Being each ${}^{M}T_{C_1}$ and ${}^{ee}T_{C_2}$ calibrated using both Camera-Arm-Camera and Camera-Arm-Pattern (or reversed) approaches, fig. 16 depicts a comparison of the accuracy accomplished in each calibration method. Note that we ignore the pattern estimated location as it is not relevant. While both methods offer a satisfactory level of precision, it is noteworthy that employing a fixed or manipulator-held pattern significantly enhances calibration accuracy. One key distinction is that, although both procedures involve estimating two 3D poses (plus the joint offsets), the latter situation uses only one camera measurement for each factor, whereas the former requires two camera measurements, each with its associated detection errors. Additionally, the need for more measurements complicates the acquisition of a rich dataset, particularly if it is obtained manually.

However, it is important to note that employing the pattern in hand for calibration necessitates the manipulator's EE to move within the fixed camera field of view, which may not always be feasible. In our robot setup, for instance, the manipulator's EE can hardly be seen by the down-looking camera. As a result, their inter-relationships must be established through the Camera-Arm-Camera approach.

D. Camera - Arm - Arm - Pattern

In this experiment, a Charuco board is attached to the EE of one manipulator, and it is detected by the camera attached to the other manipulator.(see fig. 17). Note that, using a dual-arm setup, is highly unlikely to miss the EE pattern in the camera's field of view, as both arms can be coordinated to ensure visibility. Thus, we do not consider Camera-Camera calibration, as the analysis presented in section VI-C is equally applicable in this scenario.

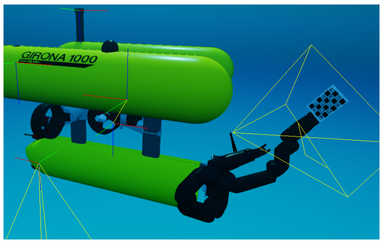


Fig. 17. The simulated Girona1000 with an EE pattern and an EE camera.

The calibration procedure is tested with a dataset of 42 samples, using the Point - Pixel projection factor.

The known, ground truth parameters are:

$$
\begin{aligned}
{}^{M_1}\check{T}_{M_2} &= \begin{cases} pos: & (-0.048, -0.272, 0.0) \\ rpy: & (0.0, 0.031, -20.05) \end{cases} \\
{}^{ee1}\check{T}_{C_1} &= \begin{cases} pos: & (0.1, 0.1, 0.1) \\ rpy: & (0.0, -11.45, 0.0) \end{cases} \\
{}^{ee2}\check{T}_{L} &= \begin{cases} pos: & (-0.1, 0.1, 0.175) \\ rpy: & (90, 0, -90) \end{cases} \\
q_{b1} &= [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]^T \\
q_{b2} &= [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]^T
\end{aligned}
$$

The initial parameters for the calibration are set to:

$$\begin{aligned} {}^{M_1}T_{M_2} &= \begin{cases} pos: & (0.15, 0.1, -0.1)\\ rpy: & (0.0, 0.0, -10)\end{cases}\\ {}^{ee1}T_{C_1} &= \begin{cases} pos: & (-0.15, 0.0, 0.25)\\ rpy: & (-12.0, -3, 10.0)\end{cases}\\ {}^{ee2}T_{L} &= \begin{cases} pos: & (0.0, 0.0, 0.05)\\ rpy: & (75, 8, -80)\end{cases}\\ q_{b1} &= [0, 4, -3, 2, -2, 0.0]^T,\\ q_{b2} &= [0, -3, -3, -4, 2, 0.0]^T \end{aligned}$$

The measurement uncertainty, in image coordinates, was set to $\sigma = 3px$ for both u and v coordinates. The built graph had 564 nodes and took 603 ms for the calibration procedure to converge, being the error with respect to the ground truth:

$$\begin{aligned} {}^{M}T_{C_1}\ er. &= \begin{cases} pos: & (1.4e^{-5}, -1.6e^{-4}, 1.5e^{-4})\\ rot: & (3.3e^{-4}, 3.9e^{-5}, 8.8e^{-5})\end{cases}\\ {}^{ee}T_{L}\ er. &= \begin{cases} pos: & (1.1e^{-4}, -4.4e^{-5}, 4.5e^{-4})\\ rot: & (2.1e^{-4}, 4.6e^{-5}, 4.7e^{-4})\end{cases}\\ q_b\ er. &= [0.0, -1.1e^{-5}, -4.6e^{-5}, -2.1e^{-5}, 2.6e^{-4}, 0.0]^T \end{aligned}$$

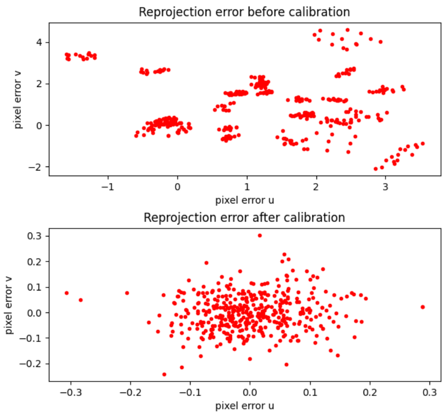


Fig. 18. Pattern projection error in pixels. Note that the scales are different in the top and bottom plots

Figure 18 depicts the reprojection error of the pattern 3D points in the camera image, with the initial parameters and after the calibration procedure.

E. Inertial Calibration

For this experiment, the simulated robot is driven along a trajectory (see fig. 19) in order to calibrate the pose of the down-looking camera w.r.t the inertial frame of the robot (${}^{R}T_{C_1}$).

The down-looking camera ground truth pose is:

$${}^{R}\check{T}_{C_1} = \begin{cases} pos: & (0.5, 0.2, 0.2)\\ rpy: & (0, 0, 0)\end{cases}$$

And the initial parameters for the calibration are set to:

$${}^{R}T_{C_1} = \begin{cases} pos: & (0.2, 0.4, 0.3)\\ rpy: & (15, -10, -18)\end{cases}$$

The accelerometers were simulated with $\sigma = 0.3\ m/s^2$ and the gyroscopes with $\sigma = 0.01\ rad/s$. Note that Stonefish does not currently simulate acceleration and angular velocity bias. The measurement uncertainty, in image coordinates, was set to $\sigma = 3px$ for both u and v coordinates.

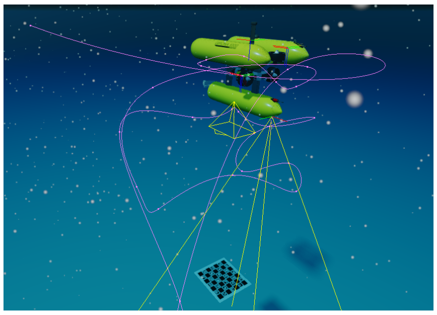


Fig. 19. The simulated setup for the inertial calibration. The trajectory of the robot is shown in purple. The axis of each sensor is shown, in addition to the field of view of the down-looking camera and the 4 beams of the DVL.

It is important to note that the full 6 DOF pose cannot be recovered if the robot rotates only around a single axis [21], [29]. This observation is significant, as the Girona500 is designed to be controllable in surge, sway, heave, and yaw while being passively stable in roll and pitch. Acknowledging this limitation, the calibration procedure has been evaluated on the same x, y, z, yaw trajectory with different ranges of roll & pitch motion.

	$err.x$	$err.y$	$err.z$	$err.rot$
0 deg	0.0011	0.0003	3.9152	0.0132
3 deg	0.0189	0.0052	0.1151	0.0045
12 deg	0.0112	0.0099	0.0225	0.0034
18 deg	0.0043	0.0024	0.0382	0.0025
28 deg	0.0008	0.0006	0.0021	0.0027

TABLE I
Comparison of calibration error on trajectories with different roll and pitch motion ranges.

The errors in translation and rotation w.r.t the ground truth shown in table I demonstrate that the vertical lever arm is not observable without some roll/pitch motion of the vehicle. However, a small range of movement in these axes is enough for the estimation procedure. Consequently, during the experimental trials (section VII-A), the vehicle is slightly induced to roll and pitch motions by attaching a weight at the tip of one manipulator, effectively moving its center of mass (see fig. 28).

Finally, fig. 20 illustrates the outcome of the calibration procedure compared with pure dead-reckoning, considering a maximum range of motion of 18 degrees in roll and pitch of

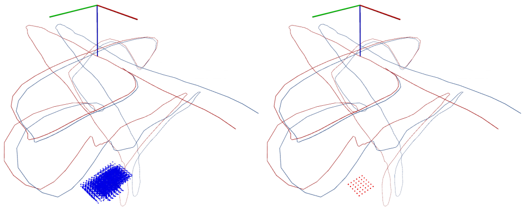


Fig. 20. Outcome of the simulated inertial calibration. In blue, the dead-reckoning path and the detected pattern using the initial camera transform guess. In red, the smoothened path and the detected pattern using the optimized camera pose.

the vehicle. The pattern detections have been integrated over the robot trajectory in both cases, considering the initial camera transform for the dead-reckoning case and the optimized transform for the latter, offering a comprehensive comparison of the impact of the calibration procedure.

VII. Experimental Results

The proposed methodology is also experimentally validated on the Girona500 I-AUV (see fig. 21). The goal is to estimate the poses of the laser scanner, the down-looking camera, the left-arm EE camera, and both manipulators, in addition to their joint offsets. Since the custom-made 3D laser scanner has its own closed-form intrinsic calibration [27], being its origin fixed at the camera frame, we treat it as a regular camera using only its raw images.

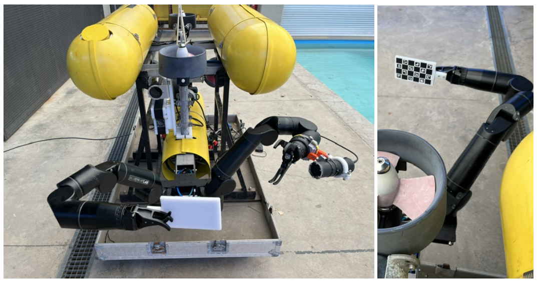


Fig. 21. The Girona500 holding a calibration target.

The calibration procedure is divided in two phases. First, it takes place the calibration of the payload components: all cameras and manipulators. Once their inter-relationships are well established, the down-looking camera is referenced to the robot's frame through the inertial calibration. Based on the simulated results, the following datasets are acquired for the 1st calibration procedure:

- 23 samples of Camera - Arm - Pattern calibration using the forward-looking laser camera and the right arm with a pattern in hand.
- 23 samples of Reversed Camera - Arm - Pattern calibration using the left arm, its EE camera, and a static pattern.
- 30 samples of Camera - Arm - Camera calibration using the left arm, its EE camera, and the forward-looking laser camera.
- 35 samples of Camera - Arm - Arm - Pattern calibration using the left arm, its EE camera, and the right arm with a pattern in hand.
- 30 samples of Camera - Arm - Camera calibration using the left arm, its EE camera, and the down-looking camera[9].

In section VI each method is tested independently to assess its accuracy. However, as some variables are estimated by various methods (e.g., joint offsets), it is preferable to build a single graph containing all the factors and optimize them all together. Combining the proposed factors requires the manipulators and fixed cameras to be referenced to a common reference frame. This is addressed in appendix A.

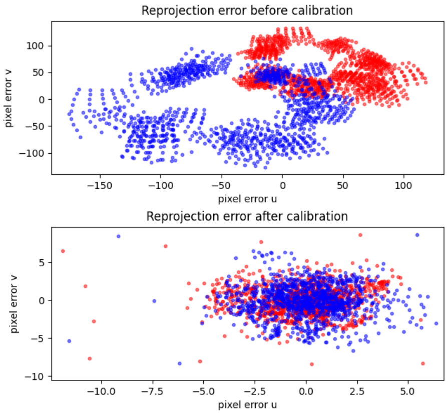
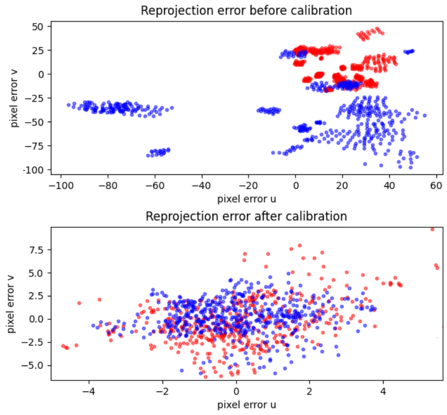


Fig. 22. Forward-looking laser camera (red) / EE camera (blue) reprojection error in pixels. Note that the scales are different in the top and bottom plots.

To assess the estimation error, the registration error between sensor pairs is calculated. Figure 22 shows the pixel error from reprojecting the EE camera to the laser camera and vice versa by evaluating the Camera - Arm - Camera factors using the estimated values. Similarly, fig. 23 shows the reprojection error between the EE camera and the down-looking camera. Figure 24 shows the pixel error from the fixed pattern to the laser camera, through the Reversed Camera - Arm - Pattern, and fig. 25 shows the pixel error from the reprojection of the EE pattern onto the laser camera. To highlight the importance of estimating joint offsets, fig. 25 shows additionally the outcome of the solution without considering them. Finally, fig. 26 shows the registration error from the dual-arm factors, projecting the right-arm pattern-in-hand onto the left-arm EE camera.

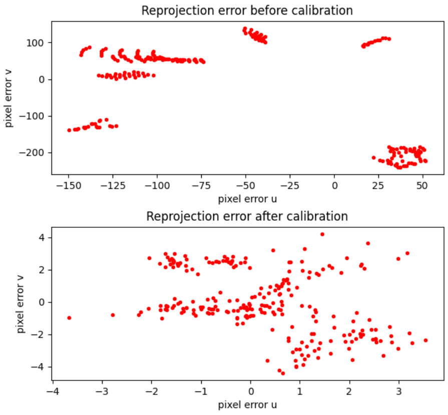
To visually assess the outcome of the calibration procedure, fig. 27 shows the robot 3D model projected onto the laser camera image in addition to the detected pattern points. Note

[9] The Girona500 was elevated with a crane to utilize the down-looking camera.


Reprojection error before calibration

Reprojection error after calibration

Fig. 23. Down-looking camera (red) / EE camera (blue) reprojection error in pixels. Note that the scales are different in the top and bottom plots.

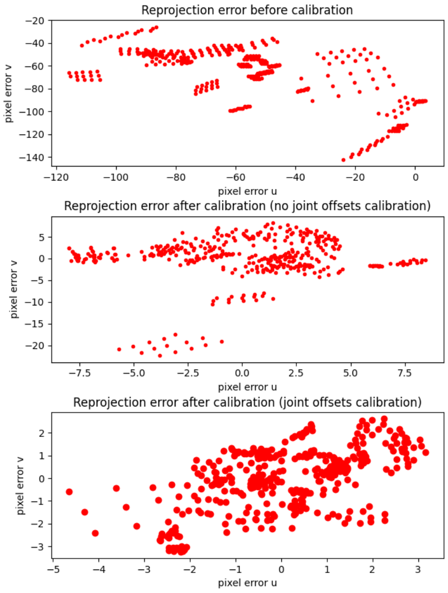

Reprojection error before calibration

Reprojection error after calibration

Fig. 24. Pattern - EE camera reprojection error in pixels. Note that the scales are different in the top and bottom plots.

how there is a perceptible misalignment within the manipulator robot model and its location in the acquired image shown in fig. 27c when the joint offsets are not calibrated. When joint offsets are considered in the calibration procedure, there is no appreciable misalignment between the robot model and the image, as shown in fig. 27d.

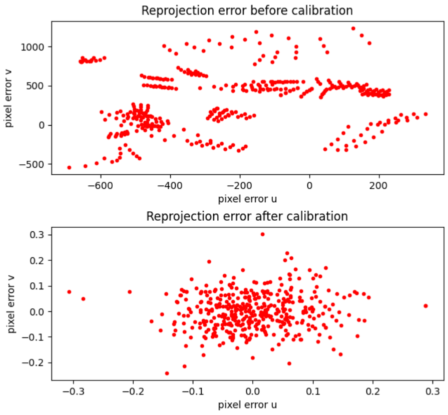
It is worth mentioning that during the experimental trials, it was observed that the payload frame of the Girona500 holding the manipulators slightly bent when the manipulators moved. This deformation was attributed to the mechanical design of the payload's aluminum frame. Consequently, the measurements are influenced by an elastic deformation that induces unmodeled noise in the optimization problem, which is reflected in the reprojection plots. While this situation suggests that the payload frame should be stiffened, the estimation procedure robustly manages this noise, yielding a highly accurate outcome despite this mechanical elasticity.


Reprojection error before calibration

Reprojection error after calibration (no joint offsets calibration)

Reprojection error after calibration (joint offsets calibration)

Fig. 25. Pattern - laser camera reprojection error in pixels. Note that the scales are different in the top and bottom plots.

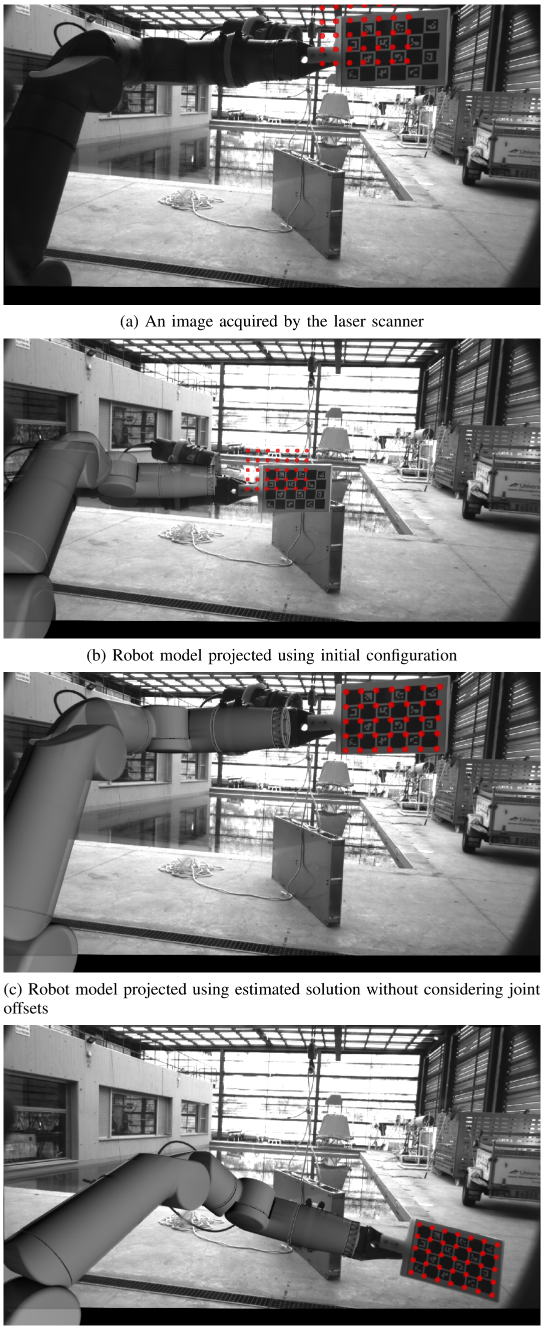

Reprojection error before calibration

Reprojection error after calibration

Fig. 26. Dual-arm pattern - EE camera reprojection error in pixels. Note that the scales are different in the top and bottom plots.

A. *Inertial calibration*

The calibration of the down-looking camera w.r.t the inertial frame of the robot was performed in the CIRS water tank. The IMU of the vehicle provides measurements at 100Hz, the DVL at 4 Hz, and the pressure gauge at 8Hz. This information is preintegrated between camera keyframes, which are set at a rate of 2 Hz. A Charuco pattern was placed in the bottom of the tank, and the robot was manually teleoperated along



(a) An image acquired by the laser scanner

(b) Robot model projected using initial configuration

(c) Robot model projected using estimated solution without considering joint offsets

(d) Robot model projected using estimated solution considering joint offsets

Fig. 27. Robot 3D model projected onto the laser camera.

a non-specific path, attempting to be sufficiently diverse in motions. A weight was attached to the tip of one manipulator, allowing it to move its center of mass during the teleoperation (see fig. 28) and induce up to 15 degrees of pitch and roll to the vehicle. Results are shown in fig. 30 where the plots show the time evolution of the estimated camera pose.

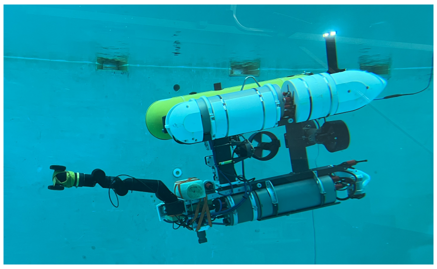


Fig. 28. The Girona500 with a weight attached to the manipulator's tip, shifting its center of mass to generate roll and pitch motion.

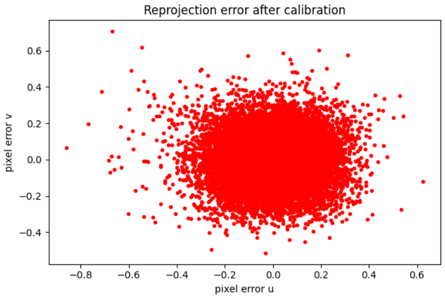
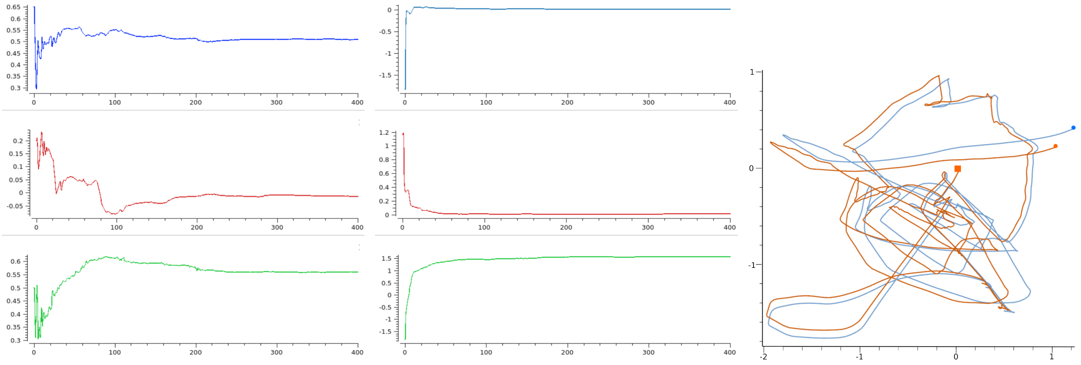


Fig. 29. Inertial calibration: camera reprojection error in pixels.

Finally, fig. 29 shows the reprojection error of the camera detections after calibration. It is important to acknowledge that the reprojection error of the inertial calibration compared to the reprojection errors of the first phase, such as Camera - Arm - Pattern (fig. 25), suggests, at first sight, that the inertial calibration achieves quite more accurate results. However, note that the inertial calibration problem has more flexibility, as each camera measurement is linked to a different robot state within the trajectory. Therefore, the optimizer can compensate for camera measurement errors by propagating them through the robot's trajectory, virtually improving the reprojection error. Hence, it is important that the uncertainties of the inertial and motion sensors are well-characterized to ensure a balance between navigation and exteroceptive costs and avoid overfitting during the calibration process.

VIII. Conclusions

We have presented an approach for calibrating a robotic framework composed of multiple cameras, manipulators, and inertial and motion sensors. We use Lie theory systematically, making this work a true on-manifold calibration approach,



(a) Evolution of the camera pose parameters (xyz).

(b) Evolution of the camera pose parameters (roll - pitch - yaw).

(c) Integrated odometry before and after calibration projected in the XY plane - respectively blue and red.

Fig. 30. Evolution of the inertial calibration.

improving the accuracy, reliability, and efficiency of the estimation process. This contrasts the most common works in the literature that usually rely on numeric differentiation and quaternion parameterization, which requires a Lagrange multiplier to keep the unit norm constraint. Our approach not only calibrates sensor positions but also estimates manipulator joint offsets, enhancing the overall performance of the robot. Additionally, by solving the inertial calibration online, our method not only calibrates sensor positions but also improves the robot's navigation, which can be used for SLAM applications. The method is validated in simulation to assess its accuracy and then tested experimentally on the Girona500 I-AUV, demonstrating its practical applicability and effectiveness.

Appendix A
Robot reference frame

It is worth noting that the minimization equations proposed in section V do not share a common reference frame. As a result, combining the single arm and dual arm factors for the calibration of the Girona500, results on the following unknowns:

- ${}^{M_1}T_{M_2}$: The pose of the manipulator M_1 w.r.t the manipulator M_2.
- ${}^{M_1}T_{C_1}$: The pose of the fixed camera C_1 w.r.t the M_1 manipulator's frame.
- ${}^{M_2}T_{C_1}$: The pose of the fixed camera C_1 w.r.t the M_2 manipulator's frame.

with one clear redundancy since ${}^{M_1}T_{M_2} \equiv {}^{M_1}T_{C_1}\,{}^{M_2}T_{C_1}^{-1}$. Hence, this parameterization would require an extra constraint, which can induce small errors.

This issue can be easily addressed by referencing the manipulators and fixed cameras to a common frame: The robot's inertial frame R. Additionally, this configuration also matches the inertial calibration. Therefore, we can define:

- ${}^{M_1}T_{M_2} := {}^{R}T_{M_1}^{-1}\,{}^{R}T_{M_2}$
- ${}^{M_1}T_{C_1} := {}^{R}T_{M_1}^{-1}\,{}^{R}T_{C_1}$
- ${}^{M_2}T_{C_1} := {}^{R}T_{M_2}^{-1}\,{}^{R}T_{C_1}$

Since all these new definitions are of the form $A = B^{-1}C$ where $B, C \in SE(3)$, the jacobians for the new factors can be easily derived by the chain rule. For a given factor η:

$$\frac{\partial\eta}{\partial B} = \frac{\partial\eta}{\partial A}\underbrace{\frac{\partial A}{\partial B^{-1}}}_{Ad(C)^{-1}}\underbrace{\frac{\partial B^{-1}}{\partial B}}_{-Ad(B)} = -\frac{\partial\eta}{\partial A}\,Ad(\underbrace{C^{-1}B}_{A^{-1}}) \tag{56}$$

$$\frac{\partial\eta}{\partial C} = \frac{\partial\eta}{\partial A}\underbrace{\frac{\partial A}{\partial C}}_{I_6} = \frac{\partial\eta}{\partial A} \tag{57}$$

Acronyms

AUV	Autonomous Underwater Vehicle
DOF	degree of freedom
DVL	Doppler Velocity Log
EE	End Effector
I-AUV	Intervention Autonomous Underwater Vehicle
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LM	Levenberg-Marquardt
LVS	Linear Velocity Sensor
NLS	nonlinear least squares
SLAM	Simultaneous Localization and Mapping

References

[1] J. Sola, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537*, 2018.

[2] R. Goldman, *Dual Quaternions and Their Associated Clifford Algebras*. CRC Press, Taylor & Francis Group, 2023.

[3] E. Bayro-Corrochano, "A survey on quaternion algebra and geometric algebra applications in engineering and computer science 1995–2020," *IEEE Access*, vol. 9, pp. 104 326–104 355, 2021.

[4] P.-A. Absil, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

[5] F. Dellaert, M. Kaess *et al.*, "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[6] G. Gallego and A. Yezzi, "A compact formula for the derivative of a 3-d rotation in exponential coordinates," *Math Imaging Vis*, vol. 51, pp. 378–384, 2015.

[7] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2301–2306.

[8] R. Unnikrishnan and M. Hebert, "Fast extrinsic calibration of a laser rangefinder to a camera," *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-09*, 2005.

[9] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the nelder–mead simplex method in low dimensions," *SIAM Journal on optimization*, vol. 9, no. 1, pp. 112–147, 1998.

[10] P. Moghadam, M. Bosse, and R. Zlot, "Line-based extrinsic calibration of range and image sensors," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3685–3691.

[11] R. Horaud and F. Dornaika, "Hand-eye calibration," *The international journal of robotics research*, vol. 14, no. 3, pp. 195–210, 1995.

[12] N. Andreff, R. Horaud, and B. Espiau, "On-line hand-eye calibration," in *Second International Conference on 3-D Digital Imaging and Modeling (Cat. No. PR00062)*. IEEE, 1999, pp. 430–436.

[13] F. C. Park and B. J. Martin, "Robot sensor calibration: solving ax= xb on the euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994.

[14] K. H. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *2006 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2006, pp. 4647–4653.

[15] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*. Springer, 2000, pp. 298–372.

[16] V. Pradeep, K. Konolige, and E. Berger, "Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach," in *Experimental Robotics: The 12th International Symposium on Experimental Robotics*. Springer, 2014, pp. 211–225.

[17] Z. Taylor and J. Nieto, "Motion-based calibration of multimodal sensor arrays," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June. Institute of Electrical and Electronics Engineers Inc., jun 2015, pp. 4843–4850.

[18] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, 2015, pp. 298–304. [Online]. Available: https://doi.org/10.3929/ethz-a-010566547

[19] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, 2017. [Online]. Available: http://doi.org/10.1177/0278364917728574www.sagepub.com/

[20] J. Brookshire and S. Teller, "Automatic calibration of multiple coplanar sensors," in *Robotics: Science and Systems*, vol. 7, 2012, pp. 33–40. [Online]. Available: http://rvsn.csail.mit.edu/calibration.

[21] ——, "Extrinsic calibration from per-sensor egomotion," in *Robotics: Science and Systems*, vol. 8, 2013, pp. 25–32. [Online]. Available: http://rvsn.csail.mit.edu/calibration3d.

[22] M. Castillón, R. Pi, N. Palomeras, and P. Ridao, "Extrinsic visual–inertial calibration for motion distortion correction of underwater 3d scans," *IEEE Access*, vol. 9, pp. 93 384–93 398, 2021.

[23] P. Vial, N. Palomeras, and M. Carreras, "On lie group imu and linear velocity preintegration for autonomous navigation considering the earth rotation compensation," 2024, [Manuscript submitted for publication].

[24] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 auv: From survey to intervention," *IEEE/ASME Transactions on mechatronics*, vol. 17, no. 1, pp. 46–53, 2011.

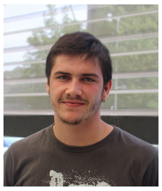
[25] M. Castillón, A. Palomer, J. Forest, and P. Ridao, "Underwater 3d scanner model using a biaxial mems mirror," *IEEE Access*, vol. 9, pp. 50 231–50 243, 2021.

[26] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and Vision Computing*, vol. 76, pp. 38 – 47, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885618300799

[27] M. Castillón, J. Forest, and P. Ridao, "Underwater 3d scanner to counteract refraction: Calibration and experimental results," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 4974–4982, 2022.

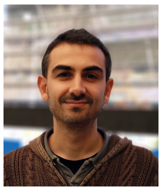
[28] P. Cieślak, "Stonefish: An Advanced Open-Source Simulation Tool Designed for Marine Robotics, With a ROS Interface," in *OCEANS 2019 - Marseille*, Jun. 2019.

[29] H. H. Chen, "A screw motion approach to uniqueness analysis of head-eye geometry," in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 1991, pp. 145–146.



Roger Pi received the B.S. degree in Computer Engineering from Universitat de Girona in 2017 and the M.Sc. degree in computer vision and robotics from the University of Burgundy, Universitat de Girona, and Heriot-Watt University in 2019, where he also received the best Master student award. He joined the Underwater Robotics Research Center (CIRS) as a Ph.D. candidate under the supervision of Dr. Pere Ridao. His research interests are focused on intervention operations for autonomous underwater vehicles.

Pau Vial received the B.Sc. degree in industrial engineering from the Polytechnic University of Catalonia in 2020. He is currently a Ph.D. candidate at the Computer Vision and Robotics Research Institute (VICOROB) from the University of Girona. His research interests include optimization techniques, Machine Learning, and Lie Theory applied to robotic navigation.



Narcís Palomeras (MSc 2004, PhD 2011) is a postdoctoral researcher at the University of Girona (UdG) and a member of the Computer Vision and Robotics Group (VICOROB). He has participated in several research projects, all related to underwater robotics, both national and European (TRIDENT, PANDORA, MORPH, MERBOTS, LOONDOCK, TWINBOT, 3DAUV, ATLANTIS...), as well as in different European competitions for AUVs such as SAUC-E and ERL. His research activity focuses mainly on underwater robotics in topics like planning, exploration, intelligent control architectures, mission control, and localization. Currently he is the coordinator of a Joint Degree Erasmus Mundus on Intelligent Field Robotic Systems.

Pere Ridao received his Ph.D. degree in industrial engineering in 2001 from the University of Girona. Since 1997, he has participated in 24 research projects (15 European and 9 National), he is the author of more than 100 publications, and he has directed 9 Ph.D. thesis (4 more under direction) and 14 MS thesis. His research activity focuses on designing and developing Autonomous Underwater Vehicles for 3D Mapping and Intervention. He is the director of the Computer Vision and Robotics Research Institute (VICOROB) and the head of the Underwater Robotics Research Center (CIRS) and an Associate Professor with the Department of Computer Engineering of the University of Girona. Dr. Ridao has served as the chair of the IFAC's Technical Committee on Marine Systems. He is also coauthor of 4 licenses and 1 Spanish/European patent, being co-founder of Iqua Robotics S.L. spin-off company.

3

Autonomous underwater cooperative transportation

In this chapter, we present a decentralized cooperative transportation scheme for a team of I-AUVs. The approach focuses on a decentralized kinematic control of two I-AUVs, simultaneously controlling their EEs velocities to achieve a common goal, with minimal data transfer between robots. The proposed approach is validated through both simulated and water tank experiments, performing the complete sequence of pick, transport, and place operations. For a visual demonstration of the experiment, please refer to the attached video, also available at https://youtu.be/epnU4v3Hz44.

Title: Twinbot: Autonomous underwater cooperative transportation
Authors: **Roger Pi**, Patryk Cieślak, Pere Ridao, and Pedro J Sanz
Journal: IEEE Access
Volume: 9, Pages: 37668–37684, Published: 2021
DOI: 10.1109/ACCESS.2021.3063669
Quality index: JCR2021 Engineering, Electrical & Electronic, Impact Factor: 3.476, Q2 (105/276)

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 19, 2021, accepted February 27, 2021, date of publication March 2, 2021, date of current version March 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3063669

TWINBOT: Autonomous Underwater Cooperative Transportation

ROGER PI[1], PATRYK CIEŚLAK[1], PERE RIDAO[1], (Member, IEEE), AND PEDRO J. SANZ[2], (Senior Member, IEEE)
[1]Computer Vision and Robotics Research Institute (VICOROB), Universitat de Girona, 17003 Girona, Spain
[2]Interactive and Robotic Systems Lab (IRS-Lab), Jaume I University, 12071 Castellón de la Plana, Spain

Corresponding author: Roger Pi (roger.pi@udg.edu)

This work was supported in part by the TWINBOT "TWIN ROBOTS FOR COOPERATIVE UNDERWATER INTERVENTION MISSIONS" Project funded by the Spanish Ministry of economy, industry, and competitiveness, under Grant DPI2017-86372-C3, in part by the Valencian Government (CIRTESU Project) under Grant IDIFEDER/2018/013, and in part by the Secretaria d'Universitats i Recerca del Departament d'Economia i Coneixement de la Generalitat de Catalunya under Grant 2019FI_B_00812.

ABSTRACT Underwater Inspection, Maintenance, and Repair operations are nowadays performed using Remotely Operated Vehicles (ROV) deployed from dynamic-positioning vessels, having high daily operational costs. During the last twenty years, the research community has been making an effort to design new Intervention Autonomous Underwater Vehicles (I-AUV), which could, in the near future, replace the ROVs, significantly decreasing these costs. Until now, the experimental work using I-AUVs has been limited to a few single-vehicle interventions, including object search and recovery, valve turning, and hot stab operations. More complex scenarios usually require the cooperation of multiple agents, i.e., the transportation of large and heavy objects. Moreover, using small, autonomous vehicles requires consideration of their limited load capacity and limited manipulation force/torque capabilities. Following the idea of multi-agent systems, in this paper we propose a possible solution: using a group of cooperating I-AUVs, thus sharing the load and optimizing the stress exerted on the manipulators. Specifically, we tackle the problem of transporting a long pipe. The presented ideas are based on a decentralized Task-Priority kinematic control algorithm adapted for the highly limited communication bandwidth available underwater. The aforementioned pipe is transported following a sequence of poses. A path-following algorithm computes the desired velocities for the robots' end-effectors, and the on-board controllers ensure tracking of these setpoints, taking into account the geometry of the pipe and the vehicles' limitations. The utilized algorithms and their practical implementation are discussed in detail and validated through extensive simulations and experimental trials performed in a test tank using two 8 DOF I-AUVs.

INDEX TERMS Autonomous underwater intervention, cooperative robots, cooperative manipulation, task priority control.

I. INTRODUCTION

Inspection, Maintenance, and Repair(IMR) operations at sea remain extremely costly and time-consuming, requiring the use of heavy-weight Remotely Operated Vehicles (ROV) supported by large Dynamic Positioning (DP) vessels and complex Tether Management Systems (TMS). In the last three decades, research in autonomous underwater robots and robotic intervention has been slowly gaining speed, aiming to tackle some of the IMR tasks that could in future be performed by Intervention Autonomous Underwater Vehicles (I-AUV). The research started with the pioneering works of OTTER [1], ODIN [2], UNION [3] and AMADEUS [4], which contributed in developing core technologies. The first field demonstrations of actual autonomous IMR operation arrived with the ALIVE project [5], where an I-AUV docked with a subsea control panel using hydraulic grasps and performed fixed-base valve turning. Similar experiments were reproduced later in the TRITON [6] project, using a significantly lighter I-AUV. Valve turning using a floating I-AUV was demonstrated during the PANDORA European project, first using learning-by-demonstration techniques [7] and later using Task Priority (TP) [8]. Valve turning in the presence of obstacles was tackled in [9] using the TP framework

The associate editor coordinating the review of this manuscript and approving it for publication was Cheng Chin.

and assuming a prior knowledge of the obstacles' positions. In [10], a method is presented employing laser scanning to build an occupancy grid used for motion planning. That paper reports experimental results obtained on autonomous manipulation in the presence of *a priori* unknown obstacles. Another typical IMR task already demonstrated is pipe inspection [11]. In this case, compliant control methods were used to ensure robust contact between the inspection tool and the inspected surface.

Object search and recovery has been another area of I-AUV research. The first object recovery from a floating vehicle was achieved in the SAUVIM [12] project. The I-AUV autonomously located and hooked an *a priori* known object, while hovering to recover it from the seabed. Similar results were recently reported in the MARIS project [13], where an I-AUV was able to detect and grasp a small pipe from the bottom of a water tank, using a specially designed 3-finger hand and a single 7 DOF robotic arm. A multipurpose object search and recovery strategy was first proposed in a Spanish project RAUVI [14], extended later in the European project TRIDENT [13], [15]. This strategy was organized in two steps: 1) First, the I-AUV performed an optical survey of the area of interest, building a photo-mosaic, and 2) The user selected a target object in the photo-mosaic and the robot was sent to recover it autonomously.

The use of multiple robots is a natural step forward in the research on autonomous underwater manipulation. Small robots do not require DP vessels and can be deployed from virtually any ship, significantly reducing the operational costs. Nevertheless, such small vehicles have limited capabilities in terms of their load capacity and the magnitude of the forces/torques that they can exert. A multi-robot system can share the load and optimize the manipulator's stress, allowing for manipulation of long, bulky, and/or heavy objects. Additional benefits of a multi-robot system are its increased robustness and coverage. In the presence of group redundancy, it may be possible to complete a task even if one of the robots fails, especially if there is no distinctive leader and the group is homogeneous [16]. On the other hand, better efficiency and shorter mission time can be achieved in search and recovery scenarios. The obvious drawback of a multi-robot system is the need to develop and implement more complex control algorithms, able to deal with the coordination of robots, taking into account the available communication links.

A. RELATED WORK

A recent survey of multi-robot manipulation, focused on cooperative transportation using ground robots, is presented in [16]. In the case of underwater robots, we are naturally interested in the concepts developed for grasping-based transportation. Very few works exist in this field. After completing the TRIDENT project in 2013, our consortium developed the idea of evolving its concept into a complete cooperative system involving two cooperative I-AUVs for load transportation and object assembly: the ''Cooperative Robots for Autonomous Underwater Intervention Missions'' (CRAUNIM) concept. Although this proposal did not mature at the EU level, it was later developed in the MARIS and TWINBOT national projects. MARIS was an Italian project [17] which experimentally demonstrated single-vehicle floating-base object recovery, as well as the kinematic simulation of a cooperative load transportation task. On the other hand, TWINBOT is a Spanish project devoted to the experimental demonstration of cooperative transportation. Simulated results of underwater cooperative object transportation have been reported in [18] and [19]. In the first case, the MARIS control strategy was based on the TP kinematic control algorithm, running independently on two I-AUVs, cooperating to transport a rigid pipe. First, each robot computes its optimal End-Effector (EE) velocity. Later, a consensus is reached, i.e., an average pipe velocity is computed, for which the robots recalculate their configuration space velocities. The second investigation treats the robots as particles immersed in a potential field generated by the goal (attracting field) and the obstacles (repelling field). Only simulation results about formation control, including I-AUV-object and I-AUV-object-environment interactions, are reported. To the authors' knowledge, there is only one previous study reporting experimental results on cooperative transportation underwater, which was presented in [20]. There, a decentralized impedance control method is proposed, where each robot is equipped with a wrist-mounted force-torque sensor. The control strategy is based on the leader-follower concept, where the leader knows the environment and commands the motion of the transported object. The followers estimate the object's motion using the force and torque readings. All robots implement an impedance control strategy based on their dynamical models.

B. CONTRIBUTION

This paper presents simulation and experimental results of the TWINBOT project, demonstrating the complete cooperative transportation sequence of pick, transport, and placement of a bulky object. The main contributions of this paper are:

1) The proposal of a simple decentralized TP kinematic control architecture, using a master-slave organization, suitable for the limited available communication bandwidth.
2) The proposal of a distributed velocity normalization method to accomplish the velocity limits among all the I-AUVs, which are required to ensure the direction of the desired object's velocity.
3) The use of a high accuracy, drift-less, vision-based navigation, using an *a priori* known optical map.
4) The use of a visual servoing method to achieve the accuracy required for grasping.
5) Experimental validation in a water tank environment.

The choice of the TP kinematic control, for solving the cooperative transportation problem, connects our work with the MARIS project, where a similar approach was used.

Therefore, it is important to understand the differences between the algorithm used in MARIS and the one proposed in this work.

In MARIS, the transportation phase requires three steps: 1) Independent optimization of full TP hierarchies for each of the robots separately, with safety tasks at the top, based on the same desired EE velocity, 2) Averaging of velocities achievable by the systems, 3) Second independent optimization of TP hierarchies with the EE velocity tracking tasks as the highest priority. As the authors underline, this strategy leads to the best effort solution, because in step 2 of the algorithm a simple average is used, which does not guarantee that the resulting desired EE velocity can be achieved in step 3. The authors provide a discussion on how a more advanced fusion policy may solve the problem, but they leave the actual derivation of such a policy for future work.

In the present work, the transportation phase requires only two steps: 1) Independent optimization of the full TP hierarchies for each of the robots separately, with safety tasks at the top, based on the EE velocities required to move a transported pipe to a specific position with a specific orientation, 2) Normalization of the resulting velocities which ensures that each of the robots can achieve EE velocity in all directions (linear and angular). While the first step is equivalent in both algorithms, our velocity normalization (step 2) allows solving the problem with one TP computation only, ensuring that the desired EE velocities are possible to achieve while satisfying all the safety tasks.

II. UVMS KINEMATICS

In this section, we present the kinematics of an Underwater Vehicle Manipulator System (UVMS). First, we define its generalized coordinates and then develop the position and velocity kinematics.

A. GENERALIZED COORDINATES

The generalised coordinates of an I-AUV system can be defined as follows:

$$\mathbf{q} = [\boldsymbol{\eta}\ \boldsymbol{q}]^T \tag{1}$$

where $\boldsymbol{\eta}$ is the generalized coordinates vector of the AUV (pose vector) and $\boldsymbol{q}$ is the generalized coordinate vector of the arm (configuration vector):

$$\boldsymbol{\eta} = [\boldsymbol{\eta}_1^T\ \boldsymbol{\eta}_2^T]^T = [x\ y\ z\ \phi\ \theta\ \psi]^T \tag{2}$$

$$\boldsymbol{q} = [q_1 \dots q_n]^T. \tag{3}$$

B. KINEMATICS OF POSITION

The position and attitude of the AUV *B–frame* with respect to the North East Down (NED) *N–frame* can be represented using the pose vector $\boldsymbol{\eta}$ or the related homogeneous matrix:

$$\begin{aligned} {}^N\boldsymbol{K_B}(\boldsymbol{\eta}) &= \begin{bmatrix} {}^N\boldsymbol{R_B}(\eta_2) & \eta_1 \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \\ {}^N\boldsymbol{R_B}(\eta_2) &= \boldsymbol{R}_{z,\psi}\boldsymbol{R}_{y,\theta}\boldsymbol{R}_{x,\phi}. \end{aligned} \tag{4}$$

The position and attitude of the arm, with respect to the 0*–frame* located at its base, can be represented with the pose vector:

$${}^0\boldsymbol{\eta_n} = [{}^0\boldsymbol{\eta_{0n_1}}^T\ {}^0\boldsymbol{\eta_{0n_2}}^T]^T = [x_{0n}\ y_{0n}\ z_{0n}\ \phi_{0n}\ \theta_{0n}\ \psi_{0n}]^T \tag{5}$$

which can be computed from the arm configuration vector ($\boldsymbol{q}$) computing the forward kinematics of the arm [21]:

$${}^0\boldsymbol{A_n}(\boldsymbol{q}) = \prod_{i=1}^{n} {}^{i-1}\boldsymbol{A_i}(\boldsymbol{q_i}) = \begin{bmatrix} {}^0\boldsymbol{R_n}({}^0\boldsymbol{\eta_{0n_2}}) & {}^0\boldsymbol{\eta_{0n_1}} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{6}$$

where ${}^{i-1}\boldsymbol{A_i}(\boldsymbol{q_i})$ are the link to link transformation matrices depending on the *DH* parameters. The Euler angles of the arm ($\boldsymbol{\eta_{0n_2}} = [\phi_{0n}\ \theta_{0n}\ \psi_{0n}]^T$) can be obtained from:

$${}^0\boldsymbol{R_n}({}^0\boldsymbol{\eta_{0n_2}}) = \boldsymbol{R}_{z,\psi_{0n}}\boldsymbol{R}_{y,\theta_{0n}}\boldsymbol{R}_{x,\phi_{0n}} \tag{7}$$

solving for ϕ_{0n}, θ_{0n} and ψ_{0n}.

Therefore, given the UVMS generalized coordinates $\mathbf{q}$, its end-effector pose:

$$\boldsymbol{\eta_{ee}} = [\boldsymbol{\eta_{ee_1}}\ \boldsymbol{\eta_{ee_2}}]^T = [x_{ee}\ y_{ee}\ z_{ee}\ \phi_{ee}\ \theta_{ee}\ \psi_{ee}]^T \tag{8}$$

defined with respect to *N–frame*, depends on the AUV pose $\boldsymbol{\eta}$ and the end-effector pose ${}^0\boldsymbol{\eta_{0n}}$ (Fig. 1). The UVMS pose, can also be expressed as a homogeneous matrix:

$$\begin{aligned} {}^N\boldsymbol{K_n}(\mathbf{q}) &= {}^N\boldsymbol{K_B}(\boldsymbol{\eta}) \cdot {}^B\boldsymbol{H_0} \cdot {}^0\boldsymbol{A_n}(\boldsymbol{q}) \\ &= \begin{bmatrix} {}^N\boldsymbol{R_n}(\boldsymbol{\eta_{ee_2}}) & \boldsymbol{\eta_{ee_1}} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}. \end{aligned} \tag{9}$$

where ${}^B\boldsymbol{H_0}$ is the transformation matrix from the *B–frame* to the 0*–frame*.

C. KINEMATICS OF VELOCITY

The UVMS Jacobian [22] relates the quasi-velocities $\boldsymbol{\zeta} = [\boldsymbol{\nu}^{\boldsymbol{T}}\ \dot{\boldsymbol{q}}^{\boldsymbol{T}}]^T$ to the end-effector rate of change $\dot{\boldsymbol{\eta}}_{\boldsymbol{ee}}$:

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{ee}} = \boldsymbol{J}(\mathbf{q})\boldsymbol{\zeta} \tag{10}$$

being $\boldsymbol{\nu} = [\boldsymbol{\nu_1}^T\ \boldsymbol{\nu_2}^T]^T = [u\ v\ w\ p\ q\ r]^T$ the AUV linear and angular velocity vector. Given the AUV Jacobian:

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{J_\nu}(\boldsymbol{\eta})\boldsymbol{\nu} \\ \boldsymbol{J_\nu}(\boldsymbol{\eta}) &= \begin{bmatrix} {}^N\boldsymbol{R_B}(\eta_2) & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \boldsymbol{J_{\nu_2}}(\eta_2) \end{bmatrix}, \end{aligned} \tag{11}$$

where $\boldsymbol{J_{\nu_2}}(\eta_2)$ is the matrix mapping the angular velocity into the Euler angle rates, and given the arm geometric Jacobian:

$$\begin{aligned} \begin{bmatrix} {}^0\dot{\boldsymbol{\eta}}_{\boldsymbol{0n_1}} \\ {}^0\boldsymbol{\nu_{0n_2}} \end{bmatrix} &= \boldsymbol{J_m}(\boldsymbol{q})\dot{\boldsymbol{q}} \\ \boldsymbol{J_m}(\boldsymbol{q}) &= \begin{bmatrix} \boldsymbol{J_{m,p}}(\boldsymbol{q}) \\ \boldsymbol{J_{m,o}}(\boldsymbol{q}) \end{bmatrix} \\ \boldsymbol{J_{m,p}}(\boldsymbol{q}) &= \frac{\partial \boldsymbol{\eta_{0n_1}}}{\partial \boldsymbol{q}} \\ \boldsymbol{J_{m_o}}(\boldsymbol{q}) &= \frac{\partial \boldsymbol{\nu_{0n_2}}}{\partial \boldsymbol{q}}, \end{aligned} \tag{12}$$

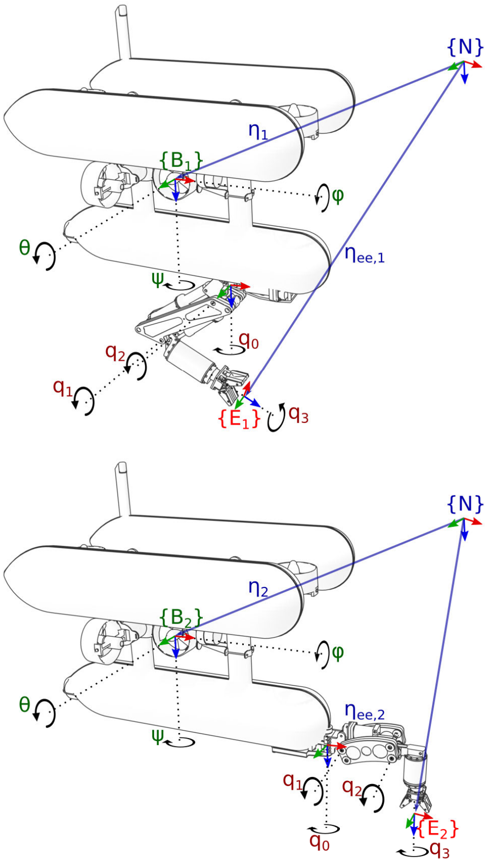


FIGURE 1. **Kinematic structure of the UVMS team.**

the UVMS geometric Jacobian $\boldsymbol{J}$ (10) can be computed as follows:

$$\begin{bmatrix}\dot{\boldsymbol{\eta}}_{\boldsymbol{ee}_1}\\ \boldsymbol{\nu}_{\boldsymbol{ee}_2}\end{bmatrix} = \begin{bmatrix}\boldsymbol{J_p}(\mathbf{q})\\ \boldsymbol{J_o}(\mathbf{q})\end{bmatrix}\begin{bmatrix}\boldsymbol{\nu}\\ \dot{\boldsymbol{q}}\end{bmatrix} = \boldsymbol{J}(\mathbf{q})\boldsymbol{\zeta} \tag{13}$$

where

$$\begin{aligned}\boldsymbol{J_p}(\mathbf{q}) &= \left[{}^{N}\boldsymbol{R_B}\ -\boldsymbol{H}{}^{N}\boldsymbol{R_B}\ \boldsymbol{J_{m,p}}(\boldsymbol{q})\right]\\ \boldsymbol{H}\ &= [{}^{N}\boldsymbol{R_B}{}^{B}\boldsymbol{r_{B0}}]_\times + [{}^{N}\boldsymbol{R_0}{}^{0}\boldsymbol{\eta_{0,ee}}]_\times\end{aligned} \tag{14}$$

and

$$\boldsymbol{J_o}(\mathbf{q}) = \left[\mathbf{0}_{3\times 3}\ {}^{N}\boldsymbol{R_B}\ \boldsymbol{J_{m,o}}(\boldsymbol{q})\right], \tag{15}$$

being $[\boldsymbol{a}]_\times$ the skew symmetric matrix so that $\boldsymbol{c} = \boldsymbol{a} \times \boldsymbol{b} = [\boldsymbol{a}]_\times \boldsymbol{b}$.

III. TASK-PRIORITY CONTROL

Tasks are designed to achieve goals, such as reaching an EE pose or avoiding joint limits. Exploiting the system redundancy, several tasks may run concurrently satisfying their goals simultaneously, e.g., achieving an EE pose while remaining within the joint limits. The priorities are used to define a strict hierarchy, stating, in case of conflicts among tasks' goals, which ones must be respected and which ones can be sacrificed. Two categories of tasks can be identified: 1) Equality tasks, whose goal is to drive the task variable to a desired value (e.g., EE pose task) and 2) Set tasks (also called inequality tasks) [23], [24], whose goal is to keep the task variable within a specified range (e.g., manipulator joint limits task).

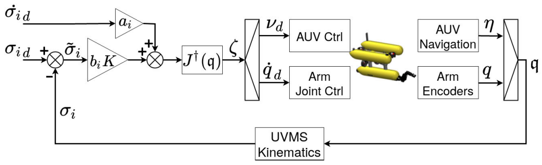


FIGURE 2. **Kinematic control for a general task $\dot{\boldsymbol{\sigma}}_{\boldsymbol{i}}$ according to** (20)**.**

A. TASK DEFINITION

A task is a m-dimensional functional constraint defined over the generalised coordinates $\mathbf{q} = [\boldsymbol{\eta}^{\boldsymbol{T}}\ \boldsymbol{q}^{\boldsymbol{T}}]^T = [x\ y\ z\ \phi\ \theta\ \psi\ q_1\ \ldots\ q_n]^T$:

$$\boldsymbol{\sigma_i} \equiv \boldsymbol{\sigma_i}(\mathbf{q}) \in \mathbb{R}^{\boldsymbol{m_i}}, \tag{16}$$

whose time derivative ($\dot{\boldsymbol{\sigma}}_{\boldsymbol{i}}$) is related to the system quasi-velocities ($\boldsymbol{\zeta} = [\boldsymbol{\nu}^{\boldsymbol{T}}\ \dot{\boldsymbol{q}}^{\boldsymbol{T}}]^{\boldsymbol{T}} = [u\ v\ w\ p\ q\ r\ q_1\ \ldots\ q_n]^T$) through the corresponding Jacobian $\boldsymbol{J_i}(\mathbf{q})$:

$$\dot{\boldsymbol{\sigma}}_{\boldsymbol{i}}(\mathbf{q}) = \boldsymbol{J_i}(\mathbf{q})\boldsymbol{\zeta}, \tag{17}$$

where

$$\boldsymbol{J_i}(\mathbf{q})) = \frac{\partial\boldsymbol{\sigma_i}(\mathbf{q})}{\partial\mathbf{q}}\frac{\partial\mathbf{q}}{\partial t} = \frac{\partial\boldsymbol{\sigma_i}(\mathbf{q})}{\partial\mathbf{q}}\begin{bmatrix}\boldsymbol{J_\nu}(\boldsymbol{\eta})\ \mathbf{0}_{\boldsymbol{6\times n}}\\ \mathbf{0}_{\boldsymbol{n\times 6}}\ \boldsymbol{I_{n\times n}}\end{bmatrix}\boldsymbol{\zeta} \tag{18}$$

and for which a corresponding error variable is defined:

$$\tilde{\boldsymbol{\sigma}}_{\boldsymbol{i}} = \tilde{\boldsymbol{\sigma}}_{\boldsymbol{i}}(\mathbf{q}) \in \mathbb{R}^{\boldsymbol{m}}\ ,\ \tilde{\boldsymbol{\sigma}}_{\boldsymbol{i}} = \boldsymbol{\sigma}_{\boldsymbol{i_d}} - \boldsymbol{\sigma_i}. \tag{19}$$

The desired task velocity vector $\dot{\boldsymbol{\sigma}}_{\boldsymbol{d\,i}}$ is then defined as:

$$\dot{\boldsymbol{\sigma}}_{\boldsymbol{i}} = a_i\dot{\boldsymbol{\sigma}}_{\boldsymbol{id}} + b_i\boldsymbol{K_i}\tilde{\boldsymbol{\sigma}}_{\boldsymbol{i}}, \tag{20}$$

where $a_i \equiv a_i \in \{0, 1\}$ and $b_i \equiv b_i(\mathrm{q}) \in \{0, 1\}$ are binary activation functions and $\boldsymbol{K_i}$ is a positive diagonal gain matrix. Equation (20) has two components: a feedback part ($\boldsymbol{K_i}\tilde{\boldsymbol{\sigma}}_{\boldsymbol{i}}$) to achieve zero regulation error and a feed-forward part ($\dot{\boldsymbol{\sigma}}_{\boldsymbol{i}}$) providing zero tracking error. Depending on the activation functions, it is possible to carry out regulation ($a_i = 0, b_i = 1$), tracking ($a_i = b_i = 1$) or optimization ($a_i = 1,\ b_i = 0$).

Generic tasks can be characterized as follows:

$$Task_i(\boldsymbol{\sigma}_{\boldsymbol{i_d}}, \dot{\boldsymbol{\sigma}}_{\boldsymbol{i_d}}, a, b) = \left\{\boldsymbol{\sigma_i}(\mathbf{q}), \tilde{\boldsymbol{\sigma}}_{\boldsymbol{i}}(\mathbf{q}), \boldsymbol{J_i}(\mathbf{q}), a_i, b_i\right\}, \tag{21}$$

where the arguments within parentheses represent time-variable parameters, whereas those within curly brackets represent arguments set up during the task instance.

B. EQUALITY TASKS

Equality tasks include two types of tasks: Regulation tasks and Tracking tasks. The first are used to drive the task variable to the desired value and are defined by five-element tuples:

$$\begin{aligned} &Regulation\_Task_i(\boldsymbol{\sigma_{i_d}}) \\ &\quad = Task_i\left\{\boldsymbol{\sigma_i}(\mathbf{q}), \boldsymbol{\tilde{\sigma}_i}(\mathbf{q}), \boldsymbol{J_i}(\mathbf{q}), a_i = 0, b_i = 1\right\}. \end{aligned} \tag{22}$$

Tracking tasks are used to drive the task variable to the desired value at a certain time and are defined by seven-element tuples:

$$\begin{aligned} &Tracking\_Task_i(\boldsymbol{\sigma_{i_d}}, \boldsymbol{\dot{\sigma}_{i_d}}) \\ &\quad = Task_i\left\{\boldsymbol{\sigma_i}(\mathbf{q}), \boldsymbol{\tilde{\sigma}_i}(\mathbf{q}), \boldsymbol{J_i}(\mathbf{q}), a_i = a, b_i = b\right\}. \end{aligned} \tag{23}$$

Several equality tasks have been reported in the literature to control the following: EE position, EE orientation, EE configuration (position and orientation), EE field of View, Robot Nominal Configuration, Vehicle orientation, Vehicle Yaw, and many others (see [22] for details). Hereafter, two examples are provided, which are used later on for the cooperative object transportation described in section IV.

1) ROBOT NOMINAL CONFIGURATION

Regulating the configuration space variables is used for multiple tasks, including keeping a specific vehicle yaw attitude with respect to a manipulated object, optimizing the shape of the manipulator, or locking the selected manipulator joints. This task can be used, for instance, to regulate the robot heading:

$$\begin{aligned} &T_{NC\psi}(\psi_d) \\ &\quad = Regulation\_Task\left\{\begin{matrix} \sigma_i(\mathbf{q}) = \psi \\ \tilde{\sigma}_i(\mathbf{q}) = \psi_d - \psi \\ \boldsymbol{J_i}(\mathbf{q}) = \left[\mathbf{0_{1\times5}}\ 1\ \mathbf{0_{1xn}}\right] \end{matrix}\right\} \end{aligned} \tag{24}$$

or to move the arm to a desired position:

$$\begin{aligned} &T_{NC_q}(\boldsymbol{q_d}) \\ &\quad = Regulation\_Task\left\{\begin{matrix} \boldsymbol{\sigma_i}(\mathbf{q}) = \boldsymbol{q} \\ \boldsymbol{\tilde{\sigma}_i}(\mathbf{q}) = \boldsymbol{q_d} - \boldsymbol{q} \\ \boldsymbol{J_i}(\mathbf{q}) = \left[\mathbf{0_{1\times6}}\ 1\ \dots\ 1\right] \end{matrix}\right\}. \end{aligned} \tag{25}$$

2) END-EFFECTOR CONFIGURATION

This task generates system velocities that result in following setpoints in EE velocity $\boldsymbol{\nu_{ee_d}}$, EE pose $\boldsymbol{\eta_{ee_d}}$, or EE trajectory $\boldsymbol{\eta_{ee_d}}$. The task variables are defined as follows:

$$\begin{aligned} T_{EEC}(\boldsymbol{\eta_{ee_d}}, \boldsymbol{\nu_{ee_d}}) &= Tracking\_Task\left\{\begin{matrix} \boldsymbol{\sigma_i}(\mathbf{q}) \\ \boldsymbol{\tilde{\sigma}_i}(\mathbf{q}) \\ \boldsymbol{J_i}(\mathbf{q}) \end{matrix}\right\} \\ \boldsymbol{\sigma_i}(\mathbf{q}) &= \boldsymbol{\eta_{ee}} \\ \boldsymbol{\tilde{\sigma}_i}(\mathbf{q}) &= \begin{bmatrix} \boldsymbol{\eta_{ee_{1_d}}} - \boldsymbol{\eta_{ee_1}} \\ \lambda_{ee}\boldsymbol{\varepsilon_d} - \lambda_d\boldsymbol{\varepsilon_{ee}} + [\boldsymbol{\varepsilon_d}]_\times\boldsymbol{\varepsilon_{ee}} \end{bmatrix} \\ \boldsymbol{J_i}(\mathbf{q}) &= \boldsymbol{J}(\mathbf{q}) \end{aligned} \tag{26}$$

where $\boldsymbol{J}(\mathbf{q})$ is the UVMS Jacobian (13) and $\boldsymbol{\tilde{Q}} = \{\boldsymbol{\tilde{\varepsilon}}, \tilde{\lambda}\}$ is the error quaternion:

$$\begin{aligned} \boldsymbol{\tilde{Q}} &= \boldsymbol{Q_d} * \boldsymbol{Q^{-1}} \\ &= \{\lambda_{ee}\boldsymbol{\varepsilon_d} - \lambda_d\boldsymbol{\varepsilon_{ee}} + [\boldsymbol{\varepsilon_d}]_\times\boldsymbol{\varepsilon_{ee}}\ ,\ \lambda_d\lambda + \boldsymbol{\varepsilon_D^T\varepsilon}\} \end{aligned} \tag{27}$$

being $\boldsymbol{Q_d} = \{\boldsymbol{\varepsilon_d}, \lambda_d\}$ the quaternion of the desired attitude and $\boldsymbol{Q} = \{\boldsymbol{\varepsilon_{ee}}, \lambda_{ee}\}$ the one corresponding to the current attitude. It is worth noting that the direction of $\boldsymbol{\tilde{\varepsilon}}$ defines the axis of rotation between the desired and current frames. Moreover the module of the vector vanishes when aligned ($\boldsymbol{\tilde{Q}} = \{\mathbf{0}, 1\}$). Therefore, it is sufficient to use $\boldsymbol{\tilde{\varepsilon}}$ to define the attitude error. The T_{EEC} task as defined above is used for EE trajectory tracking but can also be instantiated for EE configuration control alone:

$$T_{EEC_\eta}(\boldsymbol{\eta_{ee_d}}) = T_{EEC}(\boldsymbol{\eta_{ee_d}}, \mathbf{0}) \tag{28}$$

or EE velocity control alone:

$$T_{EEC_\nu}(\boldsymbol{\nu_{ee_d}}) = T_{EEC}(\mathbf{0}, \boldsymbol{\nu_{ee_d}}). \tag{29}$$

C. SET TASKS

In this paper, we follow the implementation of the set tasks proposed in [24] where a combination of several high and low priority set tasks can be used. Set tasks are scalar regulation only tasks defined as:

$$\begin{aligned} &Set\_Task_i(\sigma_{Ui}, \sigma_{L_i}, \alpha, \delta) \\ &\quad = Task_i\left\{\begin{matrix} \sigma_i(\mathbf{q}) \\ \tilde{\sigma}_i(\mathbf{q}) \\ J_i(\mathbf{q}) \\ \sigma_{i_d} \\ a_i \\ b_k \end{matrix}\right\} \\ &a_i = 0 \\ &\sigma_{i_d} = \left\{\begin{matrix} \sigma_{L_i} + \delta, & (\sigma \le \sigma_{L_i} + \alpha)) \\ \sigma_{U_i} - \delta, & (\sigma \ge \sigma_{U_i} - \alpha) \\ 0, & \textit{otherwise}\ \text{(not applicable)} \end{matrix}\right\} \\ &b_k = \left\{\begin{matrix} 0, & b_{k-1} \wedge ((\dot{\sigma} > 0) \wedge (\sigma \ge \sigma_{U_i} + \delta) \\ & \vee(\dot{\sigma} < 0) \wedge (\sigma \le \sigma_{U_i} - \delta)) \\ 1, & \neg b_{k-1} \wedge ((\dot{\sigma} < 0) \wedge (\sigma \le \sigma_{L_i} + \alpha) \\ & \vee(\dot{\sigma} > 0) \wedge (\sigma \ge \sigma_{U_i} - \alpha)) \\ & b_{k-1}, \quad \textit{otherwise} \end{matrix}\right\} \end{aligned} \tag{30}$$

where σ_{L_i}, σ_{U_i} can be defined as $-\infty$ or $+\infty$ respectively to define inequalities. Fig. 3 visually explains the Set Task concept. We are interested in keeping the task variable σ within a specified range (σ_L, σ_U). When σ approaches one of the limits, at a distance α from the respective limit, the task is activated and starts working to push the σ back to the desired set (top part of the figure). To avoid the chattering effect, the activated task is pushing the σ value to reach a distance of δ from the respective limit, with $\delta > \alpha$ (bottom part of the figure). When the σ reaches the desired set, depicted in green, the task is deactivated.

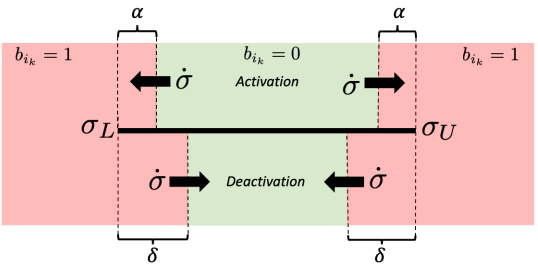


FIGURE 3. Set Task concept. The task is activated when σ approaches one of the limits, depicted in red (top), and deactivated when reaches the desired region, depicted in green (bottom).

Typical set tasks reported in the literature include Joint Limits, EE obstacle avoidance, Minimum Altitude, and Minimum Depth, to name but a few. The most representative task of this type is probably the one devoted to ensuring that the joint variables remain within their boundaries, which is described in the next subsection.

1) JOINT LIMITS

Ensuring that the control system generates position and velocity setpoints compliant with the manipulator's joint limits is a typical safety task put at the highest priority of the task hierarchy. The task is defined for each joint j as follows (written for upper limit q_{U_j} and lower limit q_{L_j}):

$$\begin{aligned} &T_{JL_j}(q_{L_j}, q_{U_j}, \alpha, \delta) \\ &\quad = Set\_Task \begin{Bmatrix} \sigma_i(\mathbf{q}) & = q_j \\ \tilde{\sigma}_i(\mathbf{q}) & = \sigma_{i_d} - \sigma_i(\mathbf{q}) \\ J_i(\mathbf{q}) & = [0_0 \; \dots \; 1_j \; \dots \; 0_{6+n}] \end{Bmatrix}. \qquad (31) \end{aligned}$$

where J_{q_j} is a single-entry row matrix, equal to 1 for the entry corresponding to the joint j. As an inequality task, its activation happens when the task variable q_j is outside its boundary, where α and δ are the activation and deactivation margins, respectively. It is required that $\delta > \alpha$ to avoid chatter.

D. PRIORITY-BASED EXECUTION OF MULTIPLE TASKS

Let us consider that we have a certain number of tasks: $\boldsymbol{\sigma_1}, \dots, \boldsymbol{\sigma_n}$ where, given two arbitrary tasks for which $\boldsymbol{\sigma_i}$ and $\boldsymbol{\sigma_j}$, $i < j \Rightarrow priority(\boldsymbol{\sigma_i}) > priority(\boldsymbol{\sigma_j})$. Then it is possible to use a recursive formulation [25] to compute the I-AUV quasi-velocities according to the established priorities:

$$\begin{aligned} i = 0: \quad & \boldsymbol{\zeta_0} = \mathbf{0}^n \\ & \boldsymbol{P_0} = \boldsymbol{I^{n\times n}} \\ i > 0: \quad & \boldsymbol{\bar{J}_i(\mathbf{q})} = \boldsymbol{J_i(\mathbf{q})P_{i-1}} \\ & \boldsymbol{P_i} = \boldsymbol{P_{i-1} - \bar{J}_i^{\dagger}(\mathbf{q})\bar{J}_i(\mathbf{q})} \\ & \boldsymbol{\zeta_i} = \boldsymbol{\zeta_{i-1} + \bar{J}_i^{\dagger}(\mathbf{q})\,(\dot{\sigma}_{id} - J_i(\mathbf{q})\zeta_{i-1})} \qquad (32) \end{aligned}$$

IV. COOPERATIVE UNDERWATER TRANSPORTATION

The goal of the cooperative underwater transportation is to perform a sequence of pick, transport, and place operations of a rigid object with a known geometry, using a group of floating I-AUVs. It is assumed that each of the robots has at least 6 DOF and can achieve any desired configuration of its EE (within the manipulator limits). Each robot has to be equipped with a navigation system, which is able to provide estimates of absolute vehicle position and orientation in the $N - frame$. The object grasping points are predefined and thus assumed to be known. To simplify the theoretical discussion, we assume that the cooperative transportation is performed using two robots transporting a pipe. The control strategy is to generate velocities in the $N - frame$, to be followed by the EEs of both robots, to move the pipe along a predefined path. Moreover, at each point of the path, a direction vector is defined reflecting the desired direction of the pipe axis. The aforementioned velocities are achieved by the robots separately by using a decentralized Task-Priority (TP) kinematic control algorithm. Although the whole system could be modeled and controlled using a centralized TP approach, that is not a practical solution. Running a control loop over underwater communications is not reliable, and it would require knowledge of the specifications of all robots, such as kinematic structure, joint limits, among others. The decentralized control permits each robot to run its own TP control, as when working independently. This embraces the modularity of the system and makes it easy to work with a non-homogeneous group of robots or even replace or reconfigure them if necessary. One of the robots is considered a master (**1**). It runs a state machine, controlling the sequence of operations, as well as the pipe path-following algorithm. It computes and communicates the desired EE velocity to the slave (**2**) and receives its EE configuration. It also sends simple commands to switch the mode of operation of the slave at certain moments of the sequence. Due to the low bandwidth of underwater acoustic communication channels, it is crucial to limit the amount of data that must be interchanged between the robots. Decentralized control permits running the sequencer and pipe setpoint controller at a lower rate than the internal robots' control, lowering the required bandwidth. Moreover, if the network fails, the robots' can continue with the last received setpoints, ensuring safety, and the mission control can recover if the network is briefly disrupted. Although it would be possible to use underwater Visual Light Communication (VLC) modems to overcome the bandwidth limitations, it would require careful planning of the modem placement as well as ensuring their line of sight, incurring additional costs. The control system scheme is presented in Fig. 4 and the following subsections describe the important blocks in detail.

A. THE SEQUENCER

A state machine, running on the master vehicle, defines the current mode of operation and changes to a new one when required. Each mode enables a different subset of TP tasks from the dynamic task hierarchy $\mathcal{H} = \left\{T_{JL_1}, T_{NC_{\psi_2}}, T_{EEC_{\eta_{3.1}}} \mid T_{EEC_{\nu_{3.2}}}, T_{NC_{q4}}\right\}$, where only one

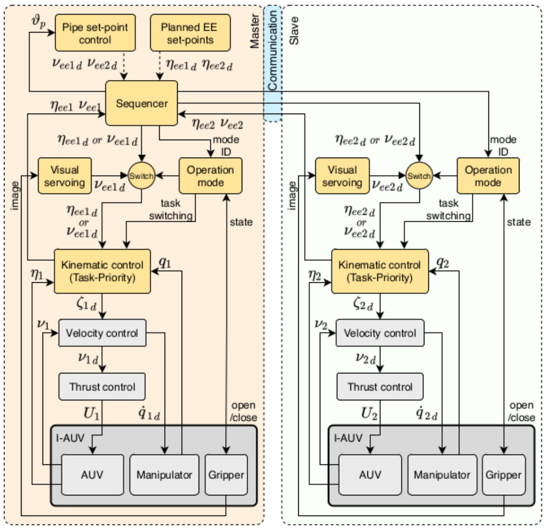


FIGURE 4. Cooperative transportation control scheme.

TABLE 1. Dynamic task priority hierarchy.

Hierarchy	Type	Description	Approach	Pick	Transport	Place
Task 1	T_{JL}	Joint Limits	✓	✓	✓	✓
Task 2	T_{NC_ψ}	AUV yaw attitude	✓	✓	✗	✓
Task 3_1	T_{EEC_η}	EE Pose	✓	✗	✗	✓
Task 3_2	T_{EEC_ν}	EE Velocity	✗	✓	✓	✗
Task 4	T_{NC_q}	Arm Configuration	✓	✓	✗	✓

T_{EEC} task is active at the time. Four modes of operation are defined (Table 1):

- **Approach**: used to approach the neighborhood of the pick-up points. This mode is triggered when both robots detect the pipe pose (see Sec. V-D), and both detections agree up to a margin error. Then, this mode is triggered, and each robot R receives its pre-grasping EE configuration setpoint $\boldsymbol{\eta_{eeR_d}}$ to be accomplished (task T_{EEC_η}). The AUV yaw (task T_{NC_ψ}) is fixed normal to the direction of the pipe, to ensure visibility of the target. It also keeps both robots parallel to each other, reducing the risk of collision.
- **Pick**: used to perform grasping using visual feedback from the end-effector camera (see Sec. V-E). During this mode of operation, the end-effector is controlled in velocity mode (task T_{EEC_ν}). Task T_{NC_ψ} and T_{NC_q} are maintained to keep a similar system configuration while approaching the pipe.
- **Transport**: intended for the cooperative transportation phase. As explained in Section IV, during the pipe transportation the robots operate in velocity mode to reduce the stress on the pipe and the manipulators (task T_{EEC_ν}). While this mode of operation is enabled, the waypoints are continuously updated by the pipe setpoint controller. This is the only mode that requires regular communication, as the master sends the desired slave's EE velocity ($\boldsymbol{\nu_{ee2_d}}$) and this reports its current EE configuration and velocity ($\boldsymbol{\eta_{ee2}}$ $\boldsymbol{\nu_{ee2}}$).
- **Place**: this mode is triggered when the robots' EEs are located at the placement pose. Then, the EE maintains position and attitude (T_{EEC_η}) while the T_{NC_ψ} task ensures AUV orthogonality with the pipe. At this point, the grippers open to drop the target in place.

In all modes of operation, the joint limits are secured by the T_{JL} task, which is always enabled.

B. PIPE PATH-FOLLOWING

The cooperative transportation occurs during the *transport mode*, when the path-following algorithm guides the pipe along a path defined as a sequence of waypoints $\boldsymbol{\vartheta_{p_i}} = \left[\boldsymbol{\eta_{p_1}^T}\ \boldsymbol{o_p^T}\right]^T$. The pipe position ($\boldsymbol{\eta_{p_1}}$) is the location of its origin ($\boldsymbol{P}$), defined as the point lying on the pipe axis which is equidistant to both holding points ($\boldsymbol{P_1}, \boldsymbol{P_2}$). During transport, the positions of the holding points should match those of the robots' EE ($\boldsymbol{P_1} = \boldsymbol{\eta_{ee2_1}}$ and $\boldsymbol{P_2} = \boldsymbol{\eta_{ee1_1}}$), and the pipe axis ($\boldsymbol{o_p}$) is given by the vector $\overrightarrow{\boldsymbol{P_1}\ \boldsymbol{P_2}}$. The control algorithm computes the linear ($\boldsymbol{\nu_p}$) and angular ($\boldsymbol{\omega_p}$) velocities that should be followed by the pipe in order to reach the target waypoint ($\boldsymbol{\vartheta_{p_i}}$). As shown in Fig. 5, the desired linear ($\boldsymbol{\nu_{p_d}}$) and angular velocities ($\boldsymbol{\omega_{p_d}}$) of the pipe can be chosen proportional to the position ($\boldsymbol{e_p}$) and orientation ($\boldsymbol{e_o}$) errors respectively:

$$\begin{aligned} \boldsymbol{e_p} &= \boldsymbol{P_d} - \boldsymbol{P}\ ;\ \boldsymbol{\nu_p} = K_p\boldsymbol{e_p} \\ \boldsymbol{e_o} &= \boldsymbol{o_p} \times \boldsymbol{o_{p_d}}\ ;\ \boldsymbol{\omega_p} = K_o\boldsymbol{e_o} \end{aligned} \tag{33}$$

Next, the desired linear velocities for $\boldsymbol{P_1}$ ($\boldsymbol{\nu_{1_d}}$) and $\boldsymbol{P_2}$ ($\boldsymbol{\nu_{2_d}}$) required to achieve the desired pipe velocity $[\boldsymbol{\nu_{p_d}^T}\ \boldsymbol{\omega_{p_d}^T}]^T$ are derived as follows:

$$\begin{aligned} \boldsymbol{\nu_{1_d}} &= \boldsymbol{\nu_{p_d}} + \boldsymbol{\nu_{o_1}}\ ;\ \boldsymbol{\nu_{o_1}} = \boldsymbol{\omega_{p_d}} \times (\boldsymbol{P_1} - \boldsymbol{P}) \\ \boldsymbol{\nu_{2_d}} &= \boldsymbol{\nu_{p_d}} + \boldsymbol{\nu_{o_2}}\ ;\ \boldsymbol{\nu_{o_2}} = \boldsymbol{\omega_{p_d}} \times (\boldsymbol{P_2} - \boldsymbol{P}) \end{aligned} \tag{34}$$

Finally, the desired EE velocity for each robot is computed:

$$\begin{aligned} \boldsymbol{\nu_{ee1_d}} &= \left[\boldsymbol{\nu_{1_d}^T}\ \boldsymbol{\omega_p^T}\right]^T \\ \boldsymbol{\nu_{ee2_d}} &= \left[\boldsymbol{\nu_{2_d}^T}\ \boldsymbol{\omega_p^T}\right]^T \end{aligned} \tag{35}$$

and fed to the corresponding T_{EE_ν} task running on both vehicles.

C. DECENTRALIZED VELOCITY NORMALIZATION

It is worth noting that the resulting EE velocities ($\boldsymbol{\nu_{ee1_d}}, \boldsymbol{\nu_{ee2_d}}$) might not be achievable by the robots due, for instance, to the velocity limits of the joints. Therefore, the achieved EE velocities may differ from the desired ones, resulting in incorrect tracking of the desired pipe velocity.

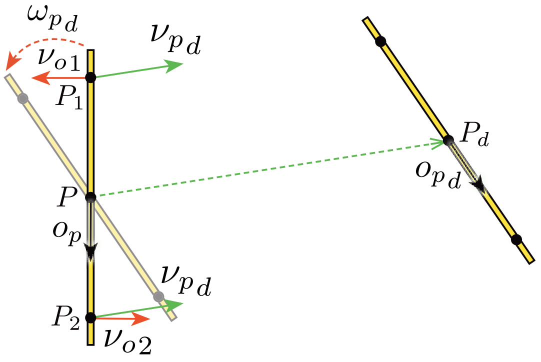


FIGURE 5. EE linear and angular velocity setpoints to reach a pipe waypoint.

In order to keep the system decentralized, a novel method is proposed to scale the desired EE velocity of both I-AUVs (Alg. 1). The method uses an inverse scaling factor s, which is iteratively adapted. First, both velocity setpoints computed by the task priority hierarchy of each robot are scaled according to the inverse scale factor of the previous iteration ($s_{k-1}^{-1}\boldsymbol{\nu}_{\boldsymbol{ee_1,d}}$, $s_{k-1}^{-1}\boldsymbol{\nu}_{\boldsymbol{ee_2,d}}$). Then, their discrepancies with the previously achieved end-effector velocities ($\boldsymbol{\Delta\nu_{ee_1}}$ and $\boldsymbol{\Delta\nu_{ee_2}}$) are computed (lines 1 and 2). If the discrepancies ($\Delta\nu_{ee_j}(i)$) of all the Cartesian components ($i \in \{1..6\}$) of both setpoints ($j \in \{1, 2\}$) are all smaller than a threshold (ε), then the inverse scaling factor is lowered (line 4), but never allowed to fall below 1. This ensures that the inverse scaling factor always scales down the requested velocities and never amplifies them. Otherwise, if any component discrepancy overpasses the threshold, the maximum discrepancy observed (m) in any of the (i) components of both setpoints (j) is computed (line 8), being used to increase the absolute value of the inverse scale s by a maximum increment of ϵ (line 9). Increasing s will tend to decrease the discrepancies in the next iteration. Finally, the new factor s is smoothed through an exponential filter (line 11) before applying it to scale both velocity setpoints (lines 12 and 13).

Algorithm 1: Adaptive Scaling of the Desired EEs Velocity

```
1  $\boldsymbol{\Delta\nu_{ee1}} = s_{k-1}^{-1}\boldsymbol{\nu_{ee1_d}} - \boldsymbol{\nu_{ee1,k-1}}$
2  $\boldsymbol{\Delta\nu_{ee2}} = s_{k-1}^{-1}\boldsymbol{\nu_{ee2_d}} - \boldsymbol{\nu_{ee2,k-1}}$
3  if $\forall j, i\ \Delta\nu_{eej}(i) < \varepsilon$ then
4      $s_k = \mathtt{Max}\,(0.95 \cdot s_{k-1},\ 1.0)$
5  else
6      $\boldsymbol{\Delta\nu_{ee1}} = \boldsymbol{\nu_{ee1_d}} - \boldsymbol{\nu_{ee1,k-1}}$
7      $\boldsymbol{\Delta\nu_{ee2}} = \boldsymbol{\nu_{ee2_d}} - \boldsymbol{\nu_{ee2,k-1}}$
8      $m = \mathtt{Max}\,(\forall j, i\ \Delta\nu_{eej}(i) < \varepsilon)$
9      $s_k = \begin{cases} s_{k-1} + \epsilon & \text{if } k_\epsilon(m - s_{k-1}) > \epsilon, \\ s_{k-1} - \epsilon & \text{if } k_\epsilon(m - s_{k-1}) < -\epsilon, \\ s_{k-1} + k_\epsilon(m - s_{k-1}) & \text{otherwise.} \end{cases}$
10 end
11 $s_k = \alpha s_k + (1 - \alpha)s_{k-1}$
12 $\boldsymbol{\nu}'_{\boldsymbol{ee1_d}} = \boldsymbol{\nu_{ee1_d}} \cdot s_k^{-1}$
13 $\boldsymbol{\nu}'_{\boldsymbol{ee2_d}} = \boldsymbol{\nu_{ee2_d}} \cdot s_k^{-1}$
```

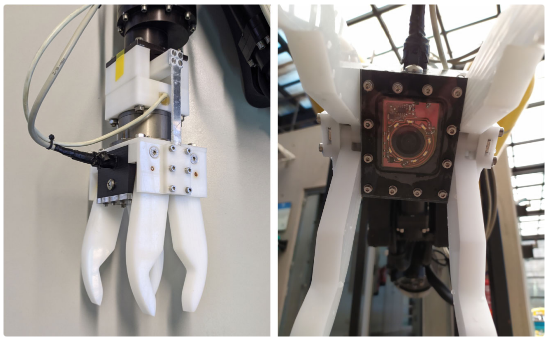


FIGURE 6. Customized end-effector with embedded camera.

V. IMPLEMENTATION

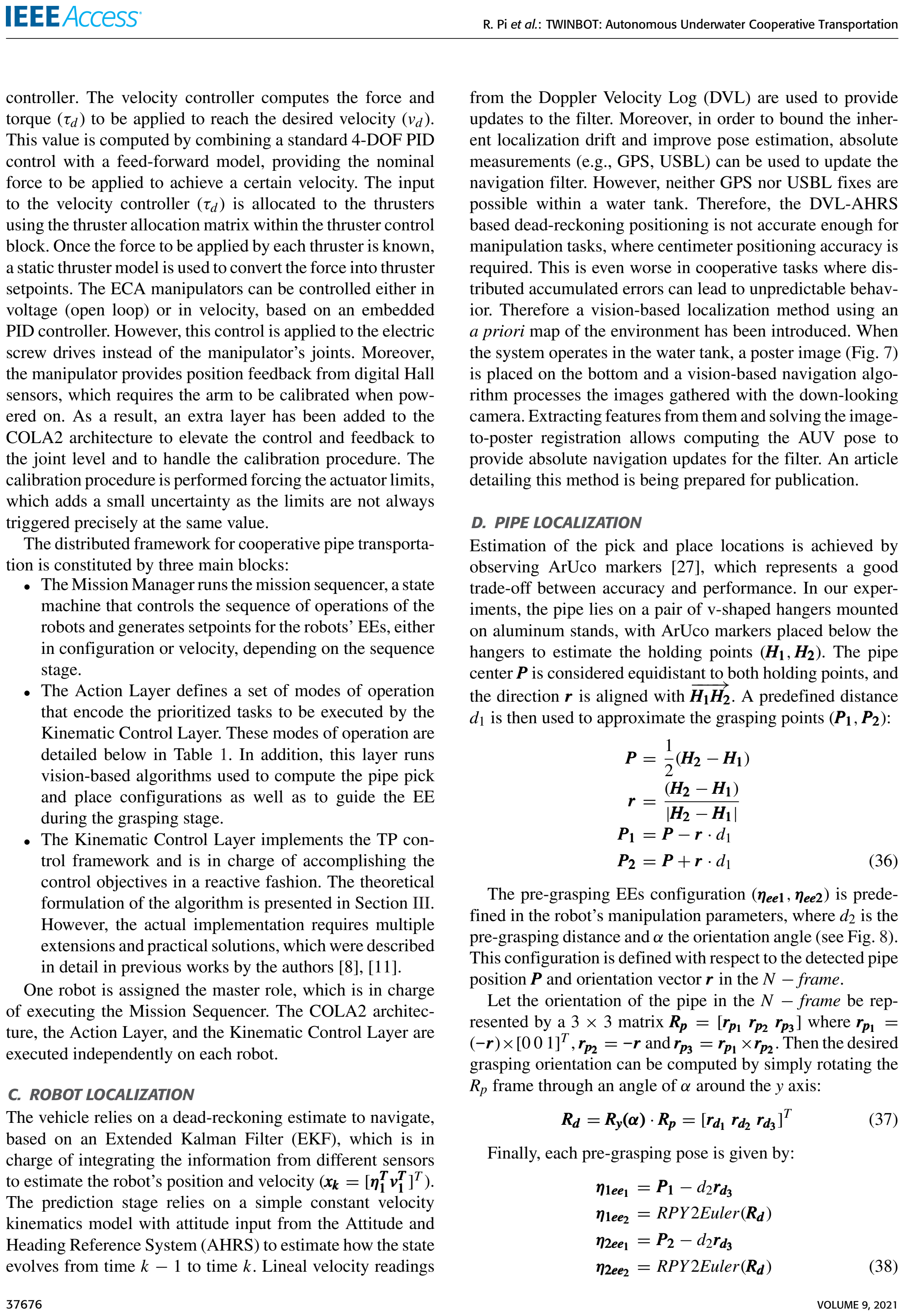
A. PLATFORMS

The team of robots consists of two GIRONA500 lightweight, modular I-AUVs designed and developed at the University of Girona [26], which can be reconfigured for different tasks by changing their payload and thruster configuration. Each vehicle was used in a 4 DOF configuration (yaw, surge, sway and heave), being passively stable in pitch and roll by design. The manipulators employed are an ECA 5E Micro and an ECA 5E Mini, both with four rotational joints actuated by electric screw drives, which makes them powerful but slow. Nevertheless, the drives exhibit high friction forces resulting in a velocity dead-band zone. The kinematic structure of both UVMSs is presented in Fig. 1. In order to correctly grasp the pipe, a custom end-effector has been designed and built (see Fig. 6). The shape of the fingers helps to drive the pipe to the end-effector center.

Additionally, three cameras are being used for each robot: 1) An analog camera attached to the body of the GIRONA500, which is forward-looking, and is used to localize the pipe based on the detection of ARUCO markers [27]; 2) An analog camera embedded in the end-effector, used to provide visual feedback to improve the grasping accuracy, and 3) A high-performance megapixel down-looking industrial camera, attached to the lower side of the I-AUV, used for getting seafloor image updates for the navigation filter.

B. SOFTWARE ARCHITECTURE

GIRONA500 uses the component-oriented layer-based software architecture for autonomy (COLA2) [28], integrating perception, navigation, guidance, and control. The navigation layer implements an extended Kalman filter (EKF) for sensor fusion to estimate the robot's position and velocity. The AUV control is based on a nested Task Priority/velocity PID



controller. The velocity controller computes the force and torque (τ_d) to be applied to reach the desired velocity (ν_d). This value is computed by combining a standard 4-DOF PID control with a feed-forward model, providing the nominal force to be applied to achieve a certain velocity. The input to the velocity controller (τ_d) is allocated to the thrusters using the thruster allocation matrix within the thruster control block. Once the force to be applied by each thruster is known, a static thruster model is used to convert the force into thruster setpoints. The ECA manipulators can be controlled either in voltage (open loop) or in velocity, based on an embedded PID controller. However, this control is applied to the electric screw drives instead of the manipulator's joints. Moreover, the manipulator provides position feedback from digital Hall sensors, which requires the arm to be calibrated when powered on. As a result, an extra layer has been added to the COLA2 architecture to elevate the control and feedback to the joint level and to handle the calibration procedure. The calibration procedure is performed forcing the actuator limits, which adds a small uncertainty as the limits are not always triggered precisely at the same value.

The distributed framework for cooperative pipe transportation is constituted by three main blocks:

- The Mission Manager runs the mission sequencer, a state machine that controls the sequence of operations of the robots and generates setpoints for the robots' EEs, either in configuration or velocity, depending on the sequence stage.
- The Action Layer defines a set of modes of operation that encode the prioritized tasks to be executed by the Kinematic Control Layer. These modes of operation are detailed below in Table 1. In addition, this layer runs vision-based algorithms used to compute the pipe pick and place configurations as well as to guide the EE during the grasping stage.
- The Kinematic Control Layer implements the TP control framework and is in charge of accomplishing the control objectives in a reactive fashion. The theoretical formulation of the algorithm is presented in Section III. However, the actual implementation requires multiple extensions and practical solutions, which were described in detail in previous works by the authors [8], [11].

One robot is assigned the master role, which is in charge of executing the Mission Sequencer. The COLA2 architecture, the Action Layer, and the Kinematic Control Layer are executed independently on each robot.

C. ROBOT LOCALIZATION

The vehicle relies on a dead-reckoning estimate to navigate, based on an Extended Kalman Filter (EKF), which is in charge of integrating the information from different sensors to estimate the robot's position and velocity ($\boldsymbol{x_k} = [\boldsymbol{\eta_1^T} \boldsymbol{v_1^T}]^T$). The prediction stage relies on a simple constant velocity kinematics model with attitude input from the Attitude and Heading Reference System (AHRS) to estimate how the state evolves from time $k-1$ to time k. Lineal velocity readings from the Doppler Velocity Log (DVL) are used to provide updates to the filter. Moreover, in order to bound the inherent localization drift and improve pose estimation, absolute measurements (e.g., GPS, USBL) can be used to update the navigation filter. However, neither GPS nor USBL fixes are possible within a water tank. Therefore, the DVL-AHRS based dead-reckoning positioning is not accurate enough for manipulation tasks, where centimeter positioning accuracy is required. This is even worse in cooperative tasks where distributed accumulated errors can lead to unpredictable behavior. Therefore a vision-based localization method using an *a priori* map of the environment has been introduced. When the system operates in the water tank, a poster image (Fig. 7) is placed on the bottom and a vision-based navigation algorithm processes the images gathered with the down-looking camera. Extracting features from them and solving the image-to-poster registration allows computing the AUV pose to provide absolute navigation updates for the filter. An article detailing this method is being prepared for publication.

D. PIPE LOCALIZATION

Estimation of the pick and place locations is achieved by observing ArUco markers [27], which represents a good trade-off between accuracy and performance. In our experiments, the pipe lies on a pair of v-shaped hangers mounted on aluminum stands, with ArUco markers placed below the hangers to estimate the holding points ($\boldsymbol{H_1}, \boldsymbol{H_2}$). The pipe center $\boldsymbol{P}$ is considered equidistant to both holding points, and the direction $\boldsymbol{r}$ is aligned with $\overrightarrow{\boldsymbol{H_1 H_2}}$. A predefined distance d_1 is then used to approximate the grasping points ($\boldsymbol{P_1}, \boldsymbol{P_2}$):

$$\begin{aligned} \boldsymbol{P} &= \frac{1}{2}(\boldsymbol{H_2} - \boldsymbol{H_1}) \\ \boldsymbol{r} &= \frac{(\boldsymbol{H_2} - \boldsymbol{H_1})}{|\boldsymbol{H_2} - \boldsymbol{H_1}|} \\ \boldsymbol{P_1} &= \boldsymbol{P} - \boldsymbol{r} \cdot d_1 \\ \boldsymbol{P_2} &= \boldsymbol{P} + \boldsymbol{r} \cdot d_1 \end{aligned} \tag{36}$$

The pre-grasping EEs configuration ($\boldsymbol{\eta_{ee1}}, \boldsymbol{\eta_{ee2}}$) is predefined in the robot's manipulation parameters, where d_2 is the pre-grasping distance and α the orientation angle (see Fig. 8). This configuration is defined with respect to the detected pipe position $\boldsymbol{P}$ and orientation vector $\boldsymbol{r}$ in the $N - frame$.

Let the orientation of the pipe in the $N - frame$ be represented by a 3×3 matrix $\boldsymbol{R_p} = [\boldsymbol{r_{p_1}}\ \boldsymbol{r_{p_2}}\ \boldsymbol{r_{p_3}}]$ where $\boldsymbol{r_{p_1}} = (-\boldsymbol{r}) \times [0\,0\,1]^T, \boldsymbol{r_{p_2}} = -\boldsymbol{r}$ and $\boldsymbol{r_{p_3}} = \boldsymbol{r_{p_1}} \times \boldsymbol{r_{p_2}}$. Then the desired grasping orientation can be computed by simply rotating the R_p frame through an angle of α around the y axis:

$$\boldsymbol{R_d} = \boldsymbol{R_y(\alpha)} \cdot \boldsymbol{R_p} = [\boldsymbol{r_{d_1}}\ \boldsymbol{r_{d_2}}\ \boldsymbol{r_{d_3}}]^T \tag{37}$$

Finally, each pre-grasping pose is given by:

$$\begin{aligned} \boldsymbol{\eta_{1ee_1}} &= \boldsymbol{P_1} - d_2 \boldsymbol{r_{d_3}} \\ \boldsymbol{\eta_{1ee_2}} &= RPY2Euler(\boldsymbol{R_d}) \\ \boldsymbol{\eta_{2ee_1}} &= \boldsymbol{P_2} - d_2 \boldsymbol{r_{d_3}} \\ \boldsymbol{\eta_{2ee_2}} &= RPY2Euler(\boldsymbol{R_d}) \end{aligned} \tag{38}$$

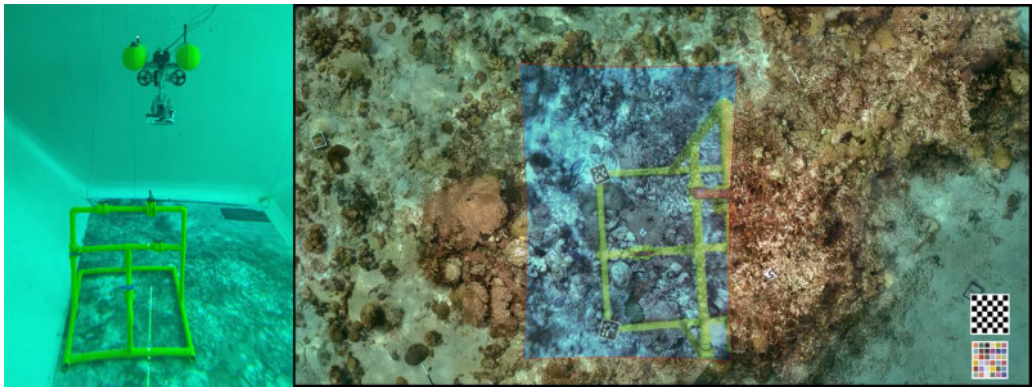


FIGURE 7. **Left: Water tank scenario with the poster placed on the bottom. Right: Example of a camera image registered to the known poster illustrating the achieved accuracy.**

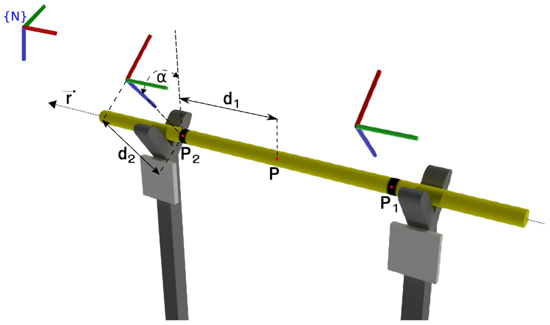


FIGURE 8. **Pre-grasping EE configuration setpoints. *P* and *r* are calculated based on the stand markers detection, whereas d_1, d_2, and α are user-defined.**

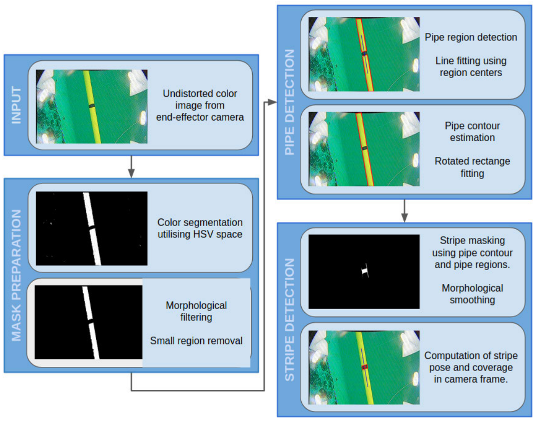

INPUT
Undistorted color image from end-effector camera
MASK PREPARATION
Color segmentation utilising HSV space
Morphological filtering
Small region removal
PIPE DETECTION
Pipe region detection
Line fitting using region centers
Pipe contour estimation
Rotated rectange fitting
STRIPE DETECTION
Stripe masking using pipe contour and pipe regions.
Morphological smoothing
Computation of stripe pose and coverage in camera frame.

FIGURE 9. **Stripe detection process.**

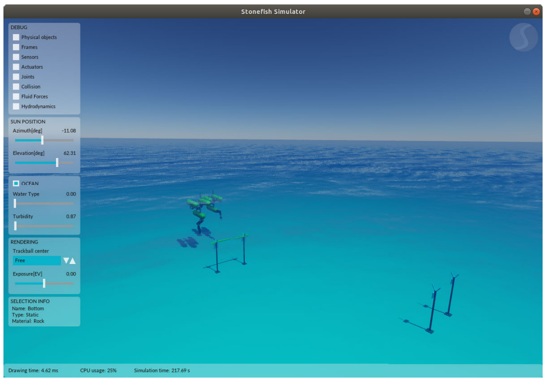
where the function $RPY2Euler(\cdot)$ computes the Euler angles of the input rotation matrix.

E. VISUAL SERVOING

Using visual feedback to control the end-effector movement is a common approach to achieve a robust execution of the grasping in real-world scenarios, mainly due to the difficulty of locating the end-effector with enough accuracy in the $N-frame$. Black stripes have been placed over the yellow-colored pipe to mark the grasping points ($\boldsymbol{P1}, \boldsymbol{P2}$). The algorithm steps, shown in Fig. 9, follow: First, the input image is converted to the HSV (hue-saturation-value) space to be ready for segmentation. HSV space is better for color segmentation as it separates color information from intensity or lighting. The segmented image is used to create a binary mask that can be applied to the original image, extracting only the colored pipe related pixels. This mask is then used to fit a line to the pipe body and estimate the entire pipe contour. Knowing the pipe contour and the colored pipe regions, the stripe region can be estimated. Finally, the stripe center is used to compute the end-effector alignment error, and the stripe area is used to estimate the distance to the pipe and set the EE desired forward velocity to approach the picking point. If the detection of the stripe fails, the previous EE velocity setpoint is kept, up to a maximum number of consecutive failed detections, after which the visual servoing would abort.

VI. SIMULATION

The core simulation tool used in this work was the open-source *Stonefish* C++ simulation library [29] combined with the Robot Operating System (ROS) interface package called *stonefish_ros*. This software is specifically designed for the simulation of marine robots. It delivers full support for rigid body dynamics of kinematic trees (like I-AUVs), geometry-based hydrodynamics, buoyancy and collision detection. It also simulates all underwater sensors and actuators to seamlessly replace the real system with a simulated robot. The full COLA2 software architectures of both robots run simultaneously with the simulator to perfectly mimic real experiments.



DEBUG
□ Physical objects
□ Frames
□ Sensors
□ Actuators
□ Joints
□ Collision
□ Fluid Forces
□ Hydrodynamics

SUN POSITION
Azimuth[deg] -11.08
Elevation[deg] 62.31

■ OCEAN
Water Type 0.00
Turbidity 0.87

RENDERING
Trackball center
Free
Exposure[EV] 0.00

SELECTION INFO
Name: Bottom
Type: Static
Material: Rock

Drawing time: 4.62 ms CPU usage: 25% Simulation time: 217.69 s

FIGURE 10. Simulated scenario using Stonefish.

The simulation setup for the cooperative pipe transportation is composed of the full model of two cooperating GIRONA500 I-AUV, each one equipped with its specific manipulator (described in Section V-A) and sensor suite, a flat sea bottom, two pairs of pipe stands and the pipe to be transported. The pipe stands are equipped with ARUCO markers [27] and the pipe appearance reflects the real one, enabling pipe detection and visual servoing using the virtual images (see Section V-D and Section V-E). Thanks to collision detection and the use of the real geometry of the robots' grippers and pipe, the grasping and transportation are precisely simulated, allowing for the identification of problems before water tank trials. Unique features of *Stonefish*, such as the simulation of a wrist-mounted force-torque sensor, allow for in-depth analysis of the system's performance and fast development of practical solutions.

A. SCENARIO

In the simulated scenario, two pairs of pipe stands ($\boldsymbol{S_{12}}, \boldsymbol{S_{34}}$) lie on the sea bottom at five meters depth, each one separated by 1.5 meters from its pair. The pose of the stands with respect to the $N - frame$ (x, y, θ) is given by: $\boldsymbol{S_{12}} = [0, 0.75, 0]^T$, $\boldsymbol{S_{34}} = [8, -3.25, 0]^T$. Since the pipe is kept parallel to the XY plane, pipe configurations are defined as $\boldsymbol{\vartheta} = [x, y, z, \varphi]^T$. The initial pipe configuration is $\boldsymbol{\vartheta} = [0, 0.75, 3.5, 0]^T$, laying on top of $\boldsymbol{S_{12}}$. Once the pipe is detected, each robot pre-grasping EE configuration ($\boldsymbol{\eta_{ee1}}, \boldsymbol{\eta_{ee2}}$) is computed and the robots are driven to it sequentially. Then, both robots simultaneously approach the grasping points ($\boldsymbol{P_1}, \boldsymbol{P_2}$) using visual guidance to finally grasp the pipe. After the pipe is fully grasped, the current pipe configuration $\boldsymbol{\vartheta_0}$ is defined, being assumed to be equidistant to both pipe holding points ($\boldsymbol{P_1}, \boldsymbol{P_2}$) and aligned with $\overrightarrow{\boldsymbol{P_1}\,\boldsymbol{P_2}}$. Next, the robot is guided along six waypoints defined with respect to $\boldsymbol{\vartheta_0}$: $\boldsymbol{\vartheta_1} = [0, 0, -0.5, 0]^T$, $\boldsymbol{\vartheta_2} = [4, 0, -0.5, 0]^T$, $\boldsymbol{\vartheta_3} = [4, 0, -0.5, \pi/2]^T$, $\boldsymbol{\vartheta_4} = [4, -4, -0.5, \pi/2]^T$, $\boldsymbol{\vartheta_5} = [4, -4, -0.5, 0]^T$ and $\boldsymbol{\vartheta_6} = [8, -4, -0.5, 0]^T$. The placement waypoint $\boldsymbol{\vartheta_7}$ is computed once the second pair of stands are detected.

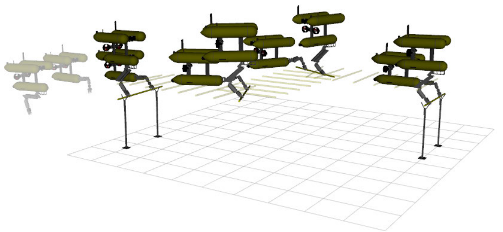


FIGURE 11. The UVMS team performing the pick, transport, and place operations on simulation. Five states out of the performed trajectory are shown.

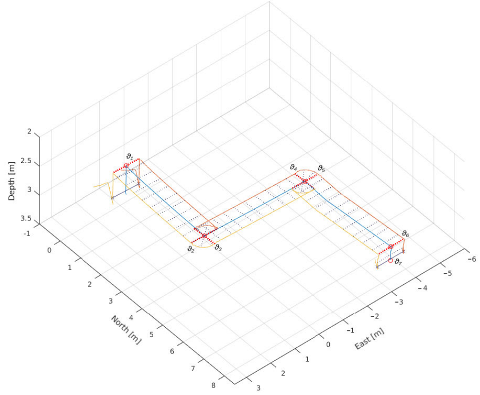


FIGURE 12. Pipe trajectory (blue) and robots' EE trajectories (yellow, orange) during cooperative transportation in the $N - frame$. Pipe orientation has been sampled over the trajectory (black dotted lines). Pipe waypoints are marked in red. Initial and final pipe configurations are shown as solid black lines.

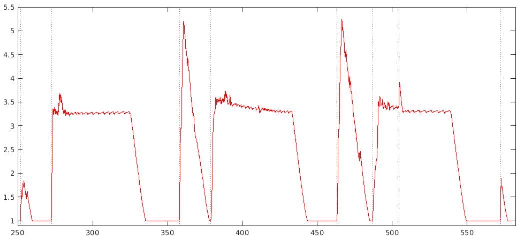


FIGURE 13. Evolution of scaling factor according to Alg. 1.

B. RESULTS

Fig. 12 presents the trajectory followed by the pipe and both EEs, demonstrating how the UVMS team successfully grasped the pipe from $\boldsymbol{S_{12}}$ stand, followed the predefined sequence of pipe waypoints, and placed it on the $\boldsymbol{S_{34}}$ stand. In addition, Fig. 11 shows the executed mission sampled over time to provide greater insight.

Fig. 14 reports the evolution of the EE's velocity in the $N - frame$ during pipe transportation, whereas Fig. 13 presents the evolution of the scaling factor according to Alg. 1. It can be

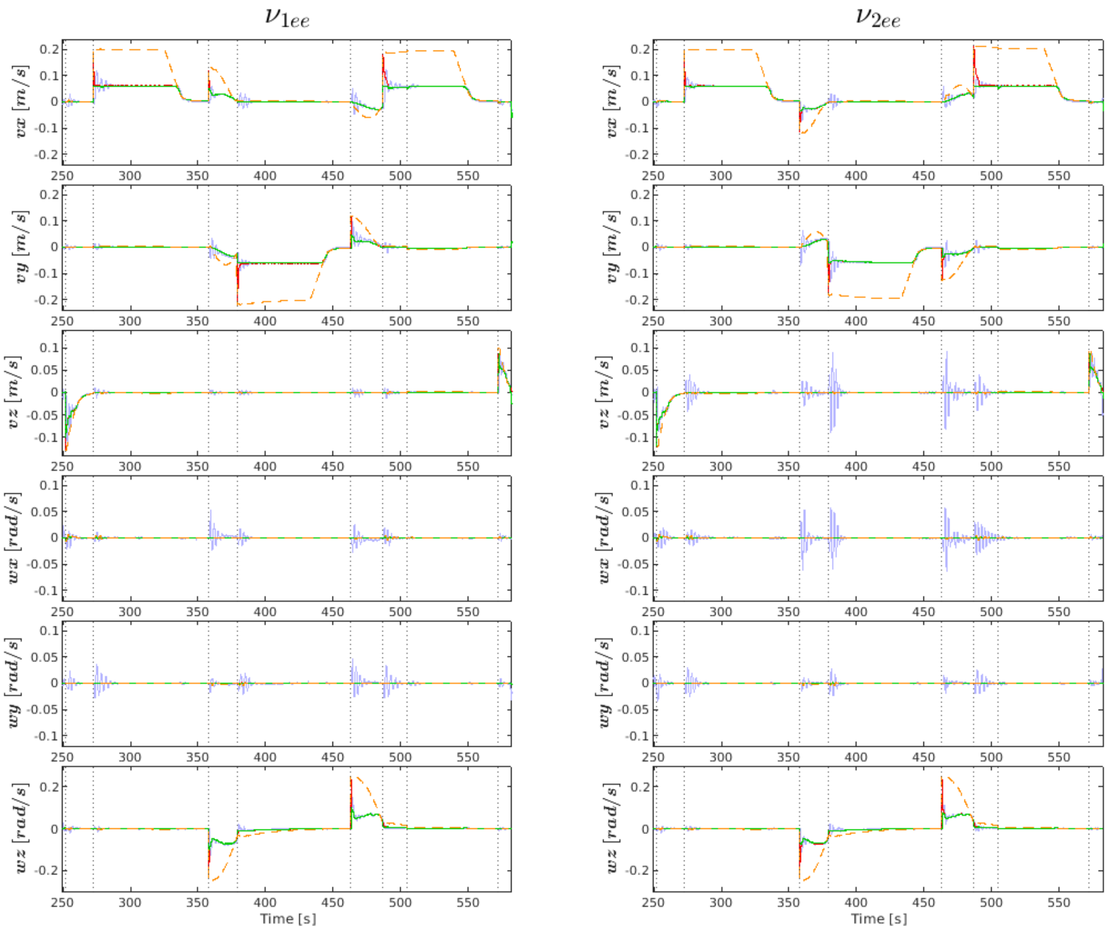


FIGURE 14. EE's velocity in the $N - frame$ during pipe transportation. The plot includes the unscaled setpoints (orange), the adapted setpoints (red), the internal controllers' velocity requests (green) and the internal controllers' feedback (blue). Setpoint changes are marked by vertical dotted lines.

seen that the velocity setpoints rapidly adapt to the robots' limitations.

A video demonstrating the whole mission using the Stonefish simulator can be found at the following url: https://youtu.be/iBnxSGs2t1U.

VII. EXPERIMENTAL VALIDATION

For the experimental validation, two GIRONA500 I-AUVs were used, one belonging to the University of Girona (the original prototype [26]) and another belonging to the University Jaume I and built by Iqua Robotics S.L (http://iquarobotics.com/). Each one was equipped with a 4 DOF ECA electric manipulator (see section V-A). The experiment was performed in the water tank of the University of Girona, Underwater Robotics Lab (CIRS). Thanks to the transparency of the Stonefish simulator, the same software architecture was used during the simulation and experimental trials. One additional component was used during the experiments - the vision-based navigation, reported in section V-C.

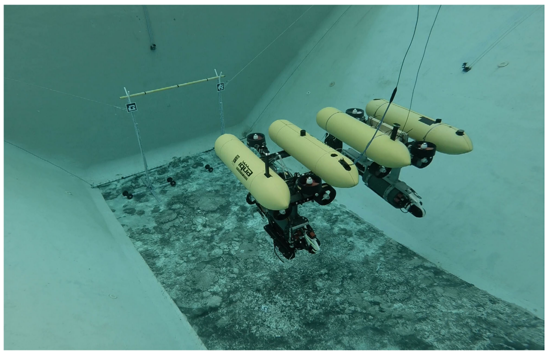
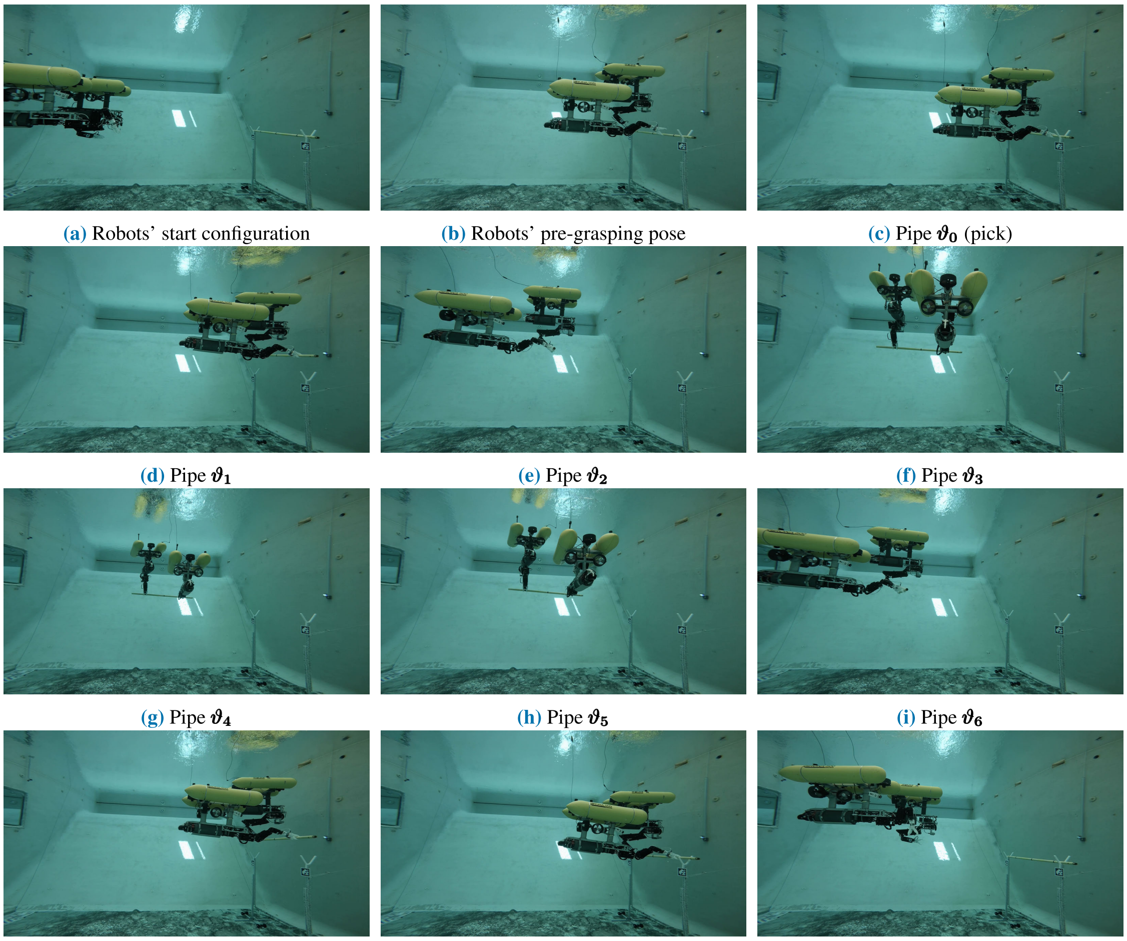


FIGURE 15. Water tank setup.



(a) Robots' start configuration (b) Robots' pre-grasping pose (c) Pipe $\boldsymbol{\vartheta}_0$ (pick)

(d) Pipe $\boldsymbol{\vartheta}_1$ (e) Pipe $\boldsymbol{\vartheta}_2$ (f) Pipe $\boldsymbol{\vartheta}_3$

(g) Pipe $\boldsymbol{\vartheta}_4$ (h) Pipe $\boldsymbol{\vartheta}_5$ (i) Pipe $\boldsymbol{\vartheta}_6$

(j) Pipe $\boldsymbol{\vartheta}_7$ (k) Pipe $\boldsymbol{\vartheta}_8$ (place) (l) Robots' retreat pose

FIGURE 16. Mission sequence in the water tank.

Although this component can also be simulated in Stonefish, it would require a lot of computational resources which can be avoided by simply lowering the simulated noise in the navigation sensors.

A. SCENARIO

The scenario, similar to the one used during the simulation, was then adapted to the dimensions of the water tank. The pipe was lying on a pair of stands separated by roughly 1.6 meters, having the grasping points marked with black stripes, 1.2 meters apart. The mission consisted of picking up the pipe, following a predefined L-shaped pipe trajectory, and placing it back to its original place. Similarly to the simulated scenario, the robots were sequentially commanded to a pre-grasping EE configuration after locating the pipe stands. Then, simultaneously, they approached the grasping points using visual servoing and grasped the pipe. Next, eight waypoints define the desired pipe trajectory, with respect to the initial pipe configuration $\boldsymbol{\vartheta}_0$: $\boldsymbol{\vartheta}_1 = [0, 0, -0.5, 0]^T$, $\boldsymbol{\vartheta}_2 = [-2.5, 0, -0.5, 0]^T$, $\boldsymbol{\vartheta}_3 = [-2.5, 0, -0.5, \pi/2]^T$, $\boldsymbol{\vartheta}_4 = [-2.5, 2.2, -0.5, \pi/2]^T$, $\boldsymbol{\vartheta}_5 = [-2.5, 0.6, -0.5, \pi/2]^T$, $\boldsymbol{\vartheta}_6 = [-2.5, 0, -0.5, 0]^T$, $\boldsymbol{\vartheta}_7 = [0, 0, -0.5, 0]^T$, and $\boldsymbol{\vartheta}_8 = \boldsymbol{\vartheta}_0$. The transition from waypoints $\boldsymbol{\vartheta}_2$ to $\boldsymbol{\vartheta}_3$ defines a 90° rotation from the center of the pipe, whereas the transition from $\boldsymbol{\vartheta}_5$ to $\boldsymbol{\vartheta}_6$ defines a $-90°$ rotation pivoting over the left robot's end-effector.

B. RESULTS

Fig. 16 presents snapshots of the performed mission in the water tank at consecutive time instants. Fig. 17 shows the trajectory for the computed pipe center and both EEs during the transportation phase. This plot has been separated into two consecutive moments for better insight.

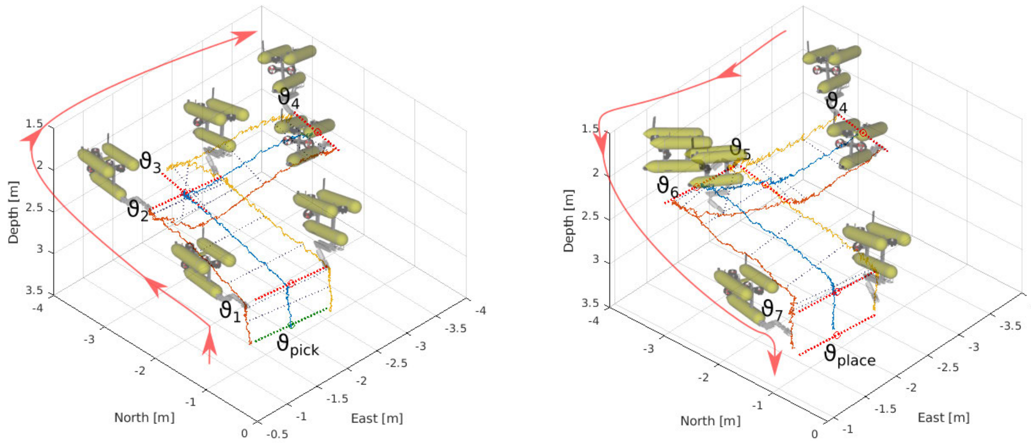


FIGURE 17. Back and forward pipe (blue) and robots' EE trajectories (yellow, orange) during the cooperative transportation phase. Pipe setpoints are marked in red.

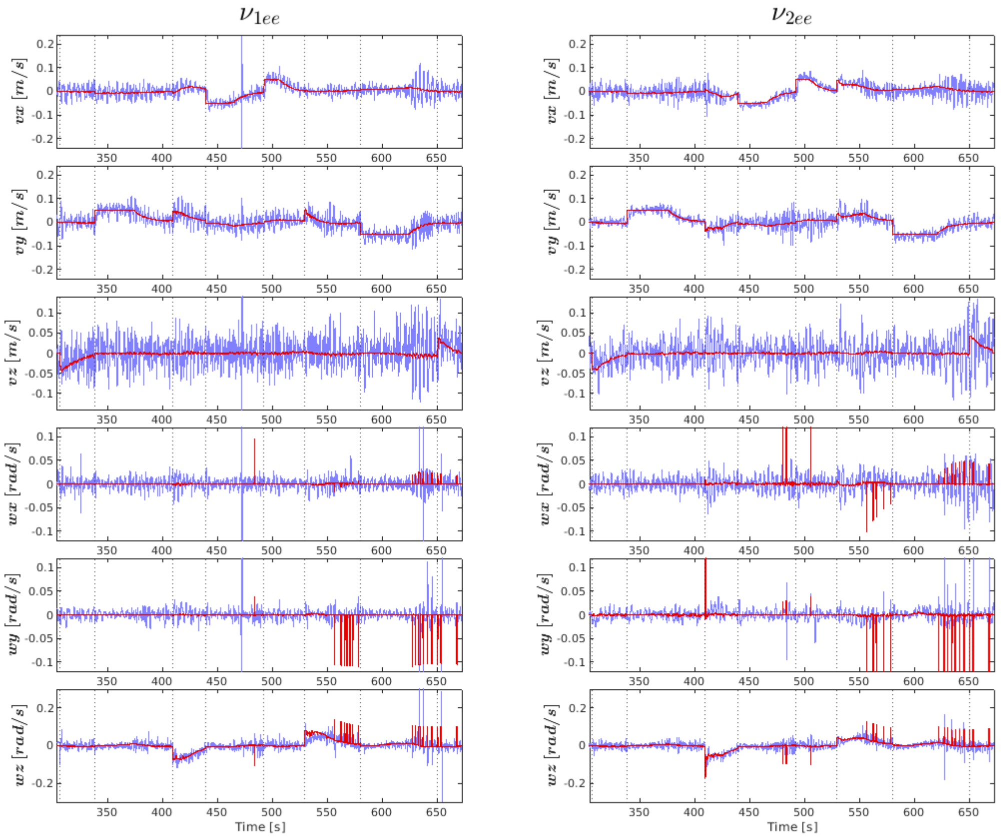


FIGURE 18. EEs velocity setpoints (red) and feedback (blue) in the $N-$*frame* during pipe transportation. Setpoint changes are marked by vertical dotted lines.

The robots can be seen to successfully perform a cooperative pipe transportation mission involving picking up the pipe, guiding it along the path through a sequence of waypoints, and placing it at its original site.

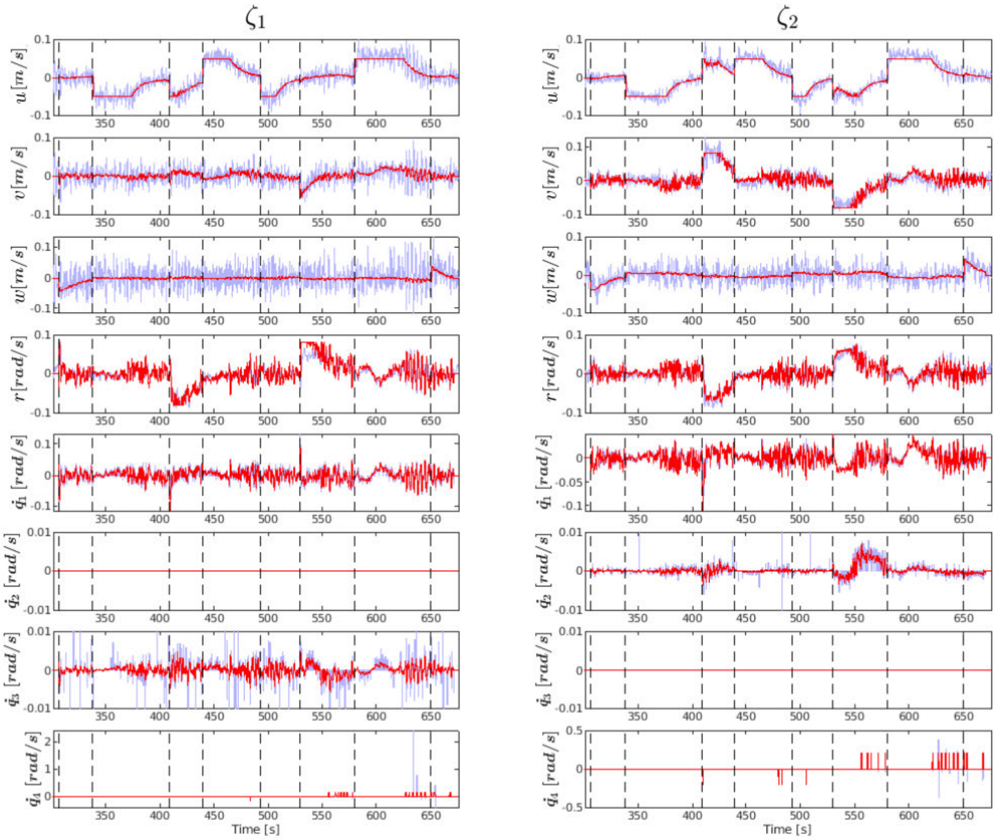


FIGURE 19. AUV and joint velocity setpoints (red) and feedback (blue) during pipe transportation. Setpoint changes are marked by vertical dotted lines.

Fig. 18 presents the control response (blue) of the end-effectors' requested velocities (red), showing how the robots were able to follow the EEs setpoints and Fig. 19 presents the actual UVMSs' quasi-velocities (vehicle and manipulator). It is worth noting that the manipulators' drives cannot achieve very-low speeds due to high internal friction forces. A custom TP extension is enabled to overcome this problem, which fixes the minimum achievable speed for those joints after the scaling, to avoid blocking the manipulator. Since the manipulator jaw suffers most from this problem, it is useful to note that it can be clearly seen that the generated spikes for $\dot{q}_4$ are aligned with the spikes visible in the EEs velocity requests.

In this experiment, due to the difficulty of operating acoustic modems in a small water tank, the communication between the robots was performed via an Ethernet cable, consistently limiting the bandwidth to simulate the real-world case. The cooperative transportation phase is the only one that requires continuous data transfer. On each iteration, the master sends an EE velocity request (6 DOF) to the slave, and this returns feedback of its EE position (6 DOF) and the achievable EE velocity (6 DOF) (see Sec. IV). For this experiment, standard ROS communications are used, and the cooperative control loop operates at 10 Hz. Without considering TCP/IP overhead, each message requires 28 bytes (6 float numbers × 4 bytes each float + 4 bytes for a timestamp), consuming a total of 6.72 kbps (3 messages × 28 bytes message × 10 Hz × 8 bits).

During the rest of the mission (approach, pick, place, and retreat), communication between the master and slave robots is only necessary for changing the mode of operation and its goal and send back an acknowledgment when it is successfully completed.

A video demonstrating the whole mission in the water tank can be found at the following url: https://youtu.be/epnU4v3Hz44.

VIII. CONCLUSION

This paper has presented a decentralized cooperative manipulation and transportation scheme for a team of UVMSs.

Simulated and water tank results have demonstrated a fully autonomous cooperative transportation mission, including detection, pickup, transport, and placement operations of a bulky rigid object. The method is based on a decentralized kinematic control of two I-AUVs, simultaneously controlling their end-effector velocities to achieve a common goal. The decentralized control provides modularity and robustness to the system and enables operation under low bandwidth communication. The desired task priority hierarchies are grouped into modes of operation to allow changing the system's entire behaviour with simple messages. The simulated and experimental results validate the proposed method.

REFERENCES

[1] H. H. Wang, S. M. Rock, and M. J. Lees, "Experiments in automatic retrieval of underwater objects with an AUV," in *Proc. Challenges Our Changing Global Environ. Conf. (OCEANS, MTS/IEEE)*, vol. 1, Oct. 1995, pp. 366–373.

[2] S. K. Choi, G. Y. Takashige, and J. Yuh, "Experimental study on an underwater robotic vehicle: ODIN," in *Proc. IEEE Symp. Auton. Underwater Vehicle Technol.*, Dec. 1994, pp. 79–84.

[3] V. Rigaud, E. Coste-Maniere, M. J. Aldon, P. Probert, M. Perrier, P. Rives, D. Simon, D. Lang, J. Kiener, A. Casal, J. Amar, P. Dauchez, and M. Chantler, "UNION: Underwater intelligent operation and navigation," *IEEE Robot. Autom. Mag.*, vol. 5, no. 1, pp. 25–35, Mar. 1998.

[4] D. M. Lane, J. B. C. Davies, G. Casalino, G. Bartolini, G. Cannata, G. Veruggio, M. Canals, C. Smith, D. J. O'Brien, M. Pickett, G. Robinson, D. Jones, E. Scott, A. Ferrara, D. Angelleti, M. Coccoli, R. Bono, P. Virgili, R. Pallas, and E. Gracia, "AMADEUS: Advanced manipulation for deep underwater sampling," *IEEE Robot. Autom. Mag.*, vol. 4, no. 4, pp. 34–45, Dec. 1997.

[5] J. Evans, P. Redmond, C. Plakas, K. Hamilton, and D. Lane, "Autonomous docking for intervention-AUVs using sonar and video-based real-time 3D pose estimation," in *Proc. OCEANS*, 2003, pp. 2201–2210.

[6] N. Palomeras, P. Ridao, D. Ribas, and G. Vallicrosa, "Autonomous I-AUV docking for fixed-base manipulation," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 12160–12165, Jan. 2014.

[7] A. Carrera, N. Palomeras, N. Hurtos, P. Kormushev, and M. Carreras, "Learning by demonstration applied to underwater intervention," in *Proc. 17th Int. Conf. Catalan Assoc. Artif. Intell. (CCIA)*, 2014, pp. 95–104.

[8] P. Cieslak, P. Ridao, and M. Giergiel, "Autonomous underwater panel operation by GIRONA500 UVMS: A practical approach to autonomous underwater manipulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2015, pp. 529–536.

[9] P. Cieślak, R. Simoni, P. R. Rodríguez, and D. Youakim, "Practical formulation of obstacle avoidance in the task-priority framework for use in robotic inspection and intervention scenarios," *Robot. Auton. Syst.*, vol. 124, Feb. 2020, Art. no. 103396. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889019306104

[10] D. Youakim, P. Cieslak, A. Dornbush, A. Palomer, P. Ridao, and M. Likhachev, "Multirepresentation, multiheuristic A* search-based motion planning for a free-floating underwater vehicle-manipulator system in unknown environment," *J. Field Robot.*, vol. 37, no. 6, pp. 925–950, Sep. 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21923

[11] P. Cieślak and P. Ridao, "Adaptive admittance control in task-priority framework for contact force control in autonomous underwater floating manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2018, pp. 6646–6651.

[12] G. Marani and J. Yuh, *Introduction to Autonomous Manipulation: Case Study with an Underwater Robot, SAUVIM*, vol. 102. Berlin, Germany: Springer, Apr. 2014.

[13] E. Simetti, F. Wanderlingh, S. Torelli, M. Bibuli, A. Odetti, G. Bruzzone, D. L. Rizzini, J. Aleotti, G. Palli, L. Moriello, and U. Scarcia, "Autonomous underwater intervention: Experimental results of the MARIS project," *IEEE J. Ocean. Eng.*, vol. 43, no. 3, pp. 620–639, Jul. 2018.

[14] M. Prats, D. Ribas, N. Palomeras, J. C. García, V. Nannen, S. Wirth, J. J. Fernández, J. P. Beltrán, R. Campos, P. Ridao, P. J. Sanz, G. Oliver, M. Carreras, N. Gracias, R. Marín, and A. Ortiz, "Reconfigurable AUV for intervention missions: A case study on underwater object recovery," *Intell. Service Robot.*, vol. 5, no. 1, pp. 19–31, Jan. 2012.

[15] P. J. Sanz, P. Ridao, G. Oliver, G. Casalino, Y. Petillot, C. Silvestre, C. Melchiorri, and A. Turetta, "Trident an European project targeted to increase the autonomy levels for underwater intervention missions," in *Proc. OCEANS-San Diego*, 2013, pp. 1–10.

[16] E. Tuci, M. H. M. Alkilabi, and O. Akanyeti, "Cooperative object transport in multi-robot systems: A review of the state-of-the-art," *Frontiers Robot. AI*, vol. 5, pp. 185–215, May 2018.

[17] G. Casalino *et al.*, "Underwater intervention robotics: An outline of the Italian national project MARIS," *Mar. Technol. Soc. J.*, vol. 50, no. 4, pp. 98–107, Jul. 2016.

[18] E. Simetti and G. Casalino, "Manipulation and transportation with cooperative underwater vehicle manipulator systems," *IEEE J. Ocean. Eng.*, vol. 42, no. 4, pp. 782–799, Oct. 2017.

[19] R. Conti, E. Meli, A. Ridolfi, and B. Allotta, "An innovative decentralized strategy for I-AUVs cooperative manipulation tasks," *Robot. Auton. Syst.*, vol. 72, pp. 261–276, Oct. 2015.

[20] S. Heshmati-Alamdari, C. P. Bechlioulis, G. C. Karras, and K. J. Kyriakopoulos, "Decentralized impedance control for cooperative manipulation of multiple underwater vehicle manipulator systems under lean communication," May 2019, *arXiv:1905.04531*. [Online]. Available: http://arxiv.org/abs/1905.04531

[21] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics* (Modelling, Planning and Control). London, U.K.: Springer, 2009.

[22] G. Antonelli, *Underwater Robots* (Springer Tracts in Advanced Robotics), vol. 96. Cham, Switzerland: Springer, Dec. 2013.

[23] E. Simetti and G. Casalino, "A novel practical technique to integrate inequality control objectives and task transitions in priority based control," *J. Intell. Robot. Syst.*, vol. 84, nos. 1–4, pp. 877–902, Dec. 2016.

[24] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results," *Frontiers Robot. AI*, vol. 3, p. 16, Apr. 2016. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2016.00016

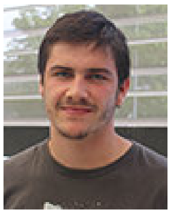
[25] B. Siciliano and J.-J.-E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Proc. 5th Int. Conf. Adv. Robot. Robots Unstruct. Environ.*, vol. 2, 1991, pp. 1211–1216.

[26] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 AUV: From survey to intervention," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 1, pp. 46–53, Feb. 2012.

[27] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image Vis. Comput.*, vol. 76, pp. 38–47, Aug. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885618300799

[28] N. Palomeras, A. El-Fakdi, M. Carreras, and P. Ridao, "COLA2: A control architecture for AUVs," *IEEE J. Ocean. Eng.*, vol. 37, no. 4, pp. 695–716, Oct. 2012.

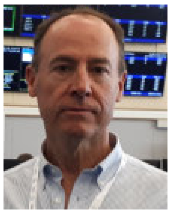
[29] P. Cieslak, "Stonefish: An advanced open-source simulation tool designed for marine robotics, with a ROS interface," in *Proc. OCEANS-Marseille*, Jun. 2019, pp. 1–6.



ROGER PI received the B.S. degree in computer engineering from the Universitat de Girona, in 2017, the M.Sc. degree in computer vision and robotics from the University of Burgundy, the M.Sc. degree in computer vision and robotics from the Universitat de Girona, and the M.Sc. degree in computer vision and robotics from Heriot-Watt University, in 2019. He is currently pursuing the Ph.D. degree with the Underwater Robotics Research Center (CIRS) under the supervision of Dr. P. Ridao. His research interest includes autonomous motion planning for underwater vehicle manipulator systems (UVMSs). He received the Best Master Student Award from Heriot-Watt University.

PATRYK CIEŚLAK received the Ph.D. degree from the Department of Robotics and Mechatronics, AGH University of Science and Technology, Kraków, Poland, in 2016. During his Ph.D. studies, he was involved in several projects focused around mechatronic design and control design in mobile robotics and manipulator systems. In his Ph.D. thesis, he tackled a difficult problem of designing a new balancing mono-wheel robot, from the mechanical structure, through model-based control, to the software and firmware architecture. He is the coauthor of a commercial rehabilitation robot called Prodrobot, which is a stationary lower limbs exoskeleton, used in the learning and improvement of natural gait patterns in children. Since 2017, he has been working with the Underwater Vision and Robotics Lab (CIRS), University of Girona, Spain. He started his work at CIRS as a Marie Skłodowska-Curie Postdoctoral Fellow, after being awarded a grant for a project aimed at developing control algorithms to enable compliant control for an I-AUV performing an autonomous non-destructive testing operation. During this project, he visited Heriot-Watt University, Edinburgh, U.K., where he worked for a period of two months, in the Ocean Systems Laboratory (OSL). After the successful completion of the grant-supported project, he continued his research in underwater manipulation, tackling tasks related to obstacle avoidance, motion planning and cooperative manipulation utilising two I-AUVs. He is continuing his research at CIRS, working on underwater intervention for inspection, maintenance and repair (IMR) operations in offshore wind farms. He is also the author of an advanced open-source robot simulator called Stonefish, designed for marine robotics. This simulator is currently used in all underwater related research at CIRS and in other institutions around the world, and can be found on Github.

PERE RIDAO (Member, IEEE) received the Ph.D. degree in industrial engineering from the University of Girona, Spain, in 2001. He is currently the Director of the Computer Vision and Robotics Research Institute (VICOROB), the Head of the Underwater Robotics Research Center (CIRS), and an Associate Professor with the Department of Computer Engineering, University of Girona. Since 1997, he has been participated in 24 research projects (15 European and nine National), he is the author of more than 100 publications, and he has supervised nine Ph.D. theses (four more are currently under direction) and 14 M.S. theses. He is the coauthor of four licenses and one Spanish/European patent, being co-founder of Iqua Robotics S.L. spin-off company. His research interests include designing and developing autonomous underwater vehicles for 3D mapping and intervention. He has served as the Chair for the IFAC's Technical Committee on Marine Systems.



PEDRO J. SANZ (Senior Member, IEEE) received the B.Sc. degree in physics from the University of Valencia (UV), the M.Sc. degree in engineering (CAD/CAM) from the Technical University of Valencia (UPV), and the Ph.D. degree in computer engineering from Universitat Jaume I (UJI), Spain. He has been appointed as a Visiting Scientist with different Universities, including TUM, Germany, from 2000 to 2016, Blaise Pascal, France, in 2002, and Bologna, Italy, in 2008. He is currently a Full Professor with the Computer Science and Engineering Department, UJI, and the Head of the Interactive and Robotic Systems Lab (IRS-Lab). He has been active since the nineties in research and development, within different projects on Advanced Robotics (e.g., Coordinator of European FP7-TRIDENT project from 2010 to 2013) and this continues (e.g., Coordinator of Spanish TWINBOT project since 2018). He is the author or coauthor of a broad range of research publications. He is an active member of different scientific societies, such as IEEE (RAS, SMC, and OES), EUCog, and euRobotics. His research interests include multi-sensory based grasping and dexterous manipulation, telerobotics, and human–robot interaction (HRI), all of them applied to real life scenarios, including assistive and underwater robotics. He was a Member of the Advisory Committee of the IEEE Systems Council from 2008 to 2012 and the Humanoids Competition Chair during the "2014 IEEE-RAS International Conference on Humanoid Robots," Madrid, in 2014. He has acted as the Chair of several Tutorials and Workshops within outstanding International Conferences on Robotics (IROS, IFAC, and ICMA). He has been the Coordinator of the Spanish Robotics Network (CEA-IFAC) from 2012 to 2016. He is also the Vice-Chair of the IEEE RAS Spanish Chapter. He has served as an Associate Editor for some prestigious journals [*IEEE Robotics & Automation Magazine (RAM)* and IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)].

• • •

4

Compliant manipulation with quasi-rigid docking for underwater structure inspection

In this chapter, we focus on contact-based underwater structure inspection, specifically addressing the challenge of performing CP inspection on floating wind platforms, which involves measuring the corrosion potential of the submerged part of the structure using a probe with a sharp tip to puncture through the coating of the structure. As this operation requires high precision to avoid damaging the structure, the I-AUV uses a magnetic EE to attach to the structure, providing stability and reducing environmental disturbances. This approach is experimentally validated in the water tank at CIRS. For a visual demonstration of the experiment, please refer to the attached video, also available at https://youtu.be/vdijAgp6z8w.

Title: Compliant Manipulation With Quasi-Rigid Docking for Underwater Structure Inspection
Authors: **Roger Pi**, Patryk Cieślak, Joan Esteba, Narcís Palomeras, and Pere Ridao
Journal: IEEE Access
Volume: 11, Pages: 128957–128969, Published: 2023
DOI: 10.1109/ACCESS.2023.3332486
Quality index: JCR2022 Engineering, Electrical & Electronic, Impact Factor: 3.9, Q2 (100/275)

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 October 2023, accepted 9 November 2023, date of publication 13 November 2023,
date of current version 21 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3332486

RESEARCH ARTICLE

Compliant Manipulation With Quasi-Rigid Docking for Underwater Structure Inspection

ROGER PI, PATRYK CIEŚLAK, JOAN ESTEBA, NARCÍS PALOMERAS, (Member, IEEE), AND PERE RIDAO, (Member, IEEE)
Computer Vision and Robotics Research Institute (VICOROB), Universitat de Girona, 17003 Girona, Spain

Corresponding author: Roger Pi (roger.pi@udg.edu)

This work was supported in part by Secretaria d'Universitats i Recerca del Departament d'Economia i Coneixement de la Generalitat de Catalunya under Grant 2019FI_B_00812, in part by the ATLANTIS Project under Grant H2020-ICT-2019-2-871571, in part by the COOPERAMOS Project under Grant PID2020-115332RB-C32, and in part by the OPTIHROV Project under Grant PDC2021-120791-C21.

ABSTRACT Offshore wind farms are a crucial source of renewable energy, but maintenance and repair can be challenging due to their remote locations and harsh environmental conditions. Professional divers or Remotely Operated Vehicles (ROVs) are commonly used to conduct maintenance operations, but they come with high daily operational costs. Autonomous Underwater Vehicles (AUVs) have the potential to improve the efficiency, safety, and costs of maintenance operations. This project evaluates the feasibility of using an AUV to conduct a cathodic protection (CP) survey, which involves measuring the corrosion potential of underwater structures to prevent deterioration. The AUV is equipped with a manipulator that has a CP probe with a sharp tip to puncture through the structure's coating and make contact with the steel underneath. To ensure high accuracy and reduce environmental perturbances, the AUV attaches to the structure while conducting the survey. The technology and methods used in this project are demonstrated in a water tank using a Girona1000 AUV. Task Priority kinematic control is combined with a custom force control strategy based on admittance control to enable tracking of the end-effector configuration and contact force during the probing operation. The mission flow control is implemented using behavior trees. The results show that the use of AUVs for CP surveys is feasible and has the potential to significantly improve the efficiency, safety, and costs of maintenance operations in offshore wind farms.

INDEX TERMS Autonomous underwater intervention, task priority control, force control, behavior trees.

ACRONYMS

AUV	Autonomous Underwater Vehicle.
BT	Behavior Tree.
CP	cathodic protection.
DVL	Doppler Velocity Log.
EE	End Effector.
FSM	Finite State Machine.
FT	force-torque.
I-AUV	Intervention Autonomous Underwater Vehicle.
INS	Inertial Navigation System.
NDT	Non Destructive Testing.
ROV	Remotely Operated Vehicle.
SVS	Sound Velocity Sensor.
TP	Task Priority.
USBL	Ultra Short Baseline.
UVMS	Underwater Vehicle Manipulator System.

The associate editor coordinating the review of this manuscript and approving it for publication was Guilin Yang.

I. INTRODUCTION

The inspection and maintenance of underwater structures, such as offshore wind farms and oil platforms, is a challenging task due to their remote locations and harsh environmental conditions. These tasks are often performed by professional

FIGURE 1. Photography of a wind farm at Viana do Castelo, Portugal, in the context of the project ATLANTIS (credit: EDP Energias de Portugal).

divers, or ROVs deployed from dynamic-positioning vessels, imposing high daily operational costs. Intervention Autonomous Underwater Vehicles (I-AUVs) have the potential to improve the efficiency and safety of underwater structure inspection by providing a more cost-effective and reliable alternative [1].

In this paper, we focus on the problem of performing compliant manipulation with quasi-rigid docking for underwater structure inspection using an I-AUV. Specifically, we address the challenge of conducting CP inspection on a wind-float structure (see Fig. 1). Wind-float structures are offshore platforms used to generate renewable energy from wind power, and they can vary in size and shape depending on the manufacturer and location.

To perform CP inspection, we use a probe capable of puncturing through the coating of the structure and making contact with the steel underneath to measure the corrosion potential of the underwater steel structure. However, performing precise manipulation of the probe while minimizing the risk of damaging the structure is a difficult task for an I-AUV in a dynamic environment. Attaching the I-AUV to the structure in a stable and controlled manner can guarantee accurate puncturing and minimize the risk of damaging the structure. To address these challenges, we propose a compliant manipulation strategy that combines quasi-rigid docking with force control, allowing the I-AUV to attach to the wind float and perform precise CP inspection without causing damage. We also present the technology and methods used to perform the whole mission, demonstrated in a mockup scenario simulating a wind-float structure.

II. RELATED WORK

Despite the growing interest and advances in underwater robotics [2], there is still a technological gap for applications that require interaction with the environment. This gap becomes particularly evident in the literature on underwater manipulation, where the majority of the work is predominantly confined to simulation environments [1]. Thus, there is a lack of extensive experimental results using real (autonomous) underwater robots for complex manipulation tasks.

Projects like SAUVIM [3], TRIDENT [4], and MARIS [5] have explored autonomous free-floating manipulation. They have developed control architectures and employed vision systems, such as stereo cameras and laser scanning, to detect and grasp objects on the seafloor. These projects have demonstrated successful grasping of mock-up objects while considering control objectives like maintaining the object in the field of view, avoiding occlusions caused by the manipulator, and respecting joint limits.

The PANDORA project [6] focused on free-floating valve-turning operations on a panel. It employed a task-priority approach and developed control frameworks to perform these operations. The PANDORA project also included tests using learning by demonstration [7] and later with motion planning capabilities using the ROS framework "MoveIt!" to generate reference trajectories for the Underwater Vehicle Manipulator System (UVMS) [8].

In [9], the authors used an in-house developed laser scanner [10] to build an occupancy grid for motion planning. That paper reports experimental results obtained on autonomous valve-turning operations in the presence of *a priori* unknown obstacles.

The EU DexROV project [11] focused on the remote control, via satellite communications, of a (semi-autonomous) UVMS umbilically attached to a support vessel from a distant onshore control center. The primary concept is that the operator interacts with a real-time simulation environment, and a cognitive engine analyzes the control requests to turn them into motion primitives that the UVMS executes autonomously in the real environment, thus not being affected by communication latencies. They presented experimental results on a mock-up of a real oil and gas panel deployed at 30 meters depth in the Mediterranean Sea.

The Droplet Project [12] explored underwater assembly of mortarless structures using a BlueRov vehicle equipped with a robotic hand. They designed pickup/drop platforms and low-weight 3D printed blocks for passive accommodation during the dropping of the block. The experiment was extremely simplified, but set the foundation for their next work [13], using cement blocks and custom-made interlocking cone inserts.

The TWINBOT project [14] demonstrated grasping and high-accuracy cooperative transportation of a bulky pipe, using two Girona500 I-AUV, each one equipped with a 4 DOF manipulator. The authors proposed a decentralized Task Priority (TP) kinematic control architecture, using a master-slave organization, suitable for the limited available wireless communication bandwidth. Experimental results demonstrated the pick, transport, and place operations in a water tank.

Another work in underwater cooperative transportation is presented in [15] using a nonlinear model predictive control approach, capable of sharing the load among the vehicles according to their payload capabilities. The work was demonstrated experimentally in a water tank using two small, custom-made UVMSs.

Few studies have performed experimental trials to explore how I-AUVs can manage the force exchanged with the environment for force regulation activities. A sliding mode impedance control is proposed in [16] and validated in a water tank on a flat panel. Although they provide simulated results for a mobile base and a 3 DOF manipulator, only a 6 DOF vehicle (the SPIR3.0 AUV) and a fixed force-torque (FT) sensor were considered in experiments. In [17], experimental results showcased an interaction controller that allowed the UVMS to glide its end effector across a flat panel's surface on a pool setup. The force mesurements, perpendicular to the panel, were not acquired by a FTs sensor in the manipulator, but by four load cells installed in the panel itself.

In [18], experimental trials were executed in a water tank simulating the inspection of a pipe using the Girona500 I-AUV [19] equipped with a 4 DOF manipulator and a FTs sensor. Similar to the work presented in this article, the approach was based on the combination of Task Priority control and an adaptive admittance controller. The robot was commanded to continuously touch the pipe along a predefined trajectory generated using cylindrical coordinates, considering a smaller pipe radius to ensure contact.

Another interesting work in force control is presented in [20]. This work explored how to maximize contact wrenches in a desired direction for I-AUVs by proposing novel methods for determining optimal configurations and actuator forces/torques. It addressed static wrench maximization, trajectory tracking with wrench optimization, and generating large wrench impulses using dynamic motions. The work was experimentally validated using a BlueRov vehicle and a 4 DOF Reach Alpha manipulator.

A. CONTRIBUTION

In contrast to our previous work [18], this paper presents significant advances by offering a comprehensive execution of a full Non Destructive Testing (NDT) mission utilizing force control and presenting results with a real I-AUV in a water tank environment, achieving a Technology Readiness Level (TRL) of 4. This TRL designation underscores the practical applicability and readiness of our methodology for controlled environments, marking a significant milestone in the development of autonomous underwater intervention.

This new approach couples the robot with the submerged structure using a second manipulator equipped with a magnetic End Effector (EE). The attachment provides stability, reducing unintended movements and improving the overall control of the inspection tools. This combination enables precise and controlled manipulation, resulting in more reliable inspection data and accurate assessment of the structure's condition. This stability minimizes the potential for damage to both the structure and the inspection equipment, enhancing the safety of the inspection operation. The magnetic attachment is not strong enough to neglect the force reaction during the contact force operation, which could potentially detach the magnet, especially due to a long lever arm between the probing point and the attachment location. Therefore, the vehicle's thrusters are used to compensate for the reaction forces at the attachment point while the probe contacts the structure, in order to hold the robot in place.

III. MECHANICAL DESIGN

A. GIRONA1000 I-AUV

The Girona1000 is an updated version of the Girona500 AUV [19]. The most significant changes are a maximum operational depth increased to 1000MSW and the integration of an Inertial Navigation System (INS). As the Girona500, it is also equipped with a Doppler Velocity Log (DVL), a pressure gauge, a Sound Velocity Sensor (SVS), and an Ultra Short Baseline (USBL) and a GPS for absolute position measurements underwater or on the surface. In this project, it has been equipped with two manipulators:

- An **ECA 5E Micro**: a 4 DOF manipulator actuated by electrical screw drives, which limits the joint's range and makes it slow, but strong. Its strength makes it suitable for the docking task, where it has to withstand forces generated by the AUV's inertia and the manipulation task. Its end-effector is equipped with a magnetic tool with the aim to attach to ferromagnetic structures. It consists of three neodymium magnets protected with a layer of epoxy, coupled to the ECA piston, which can slide the magnets in and out of the tool's housing (see Fig. 2).
- A **Reach Bravo7 Mk2**: A 6 DOF dexterous manipulator faster and more accurate, allowing precise manipulation. Its end-effector is equipped with a 6-axis force-torque FT sensor and a CP probe to perform measurements while keeping desired contact force.

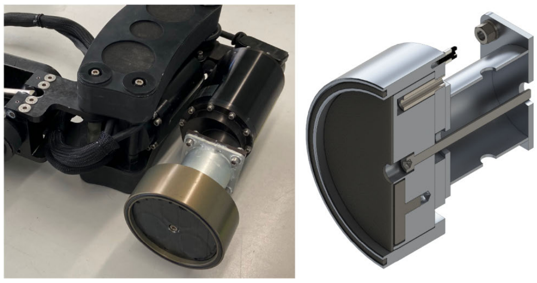


FIGURE 2. Magnetic end-effector design. On the left, the end-effector mounted on the ECA manipulator. On the right, a cross section of the CAD design.

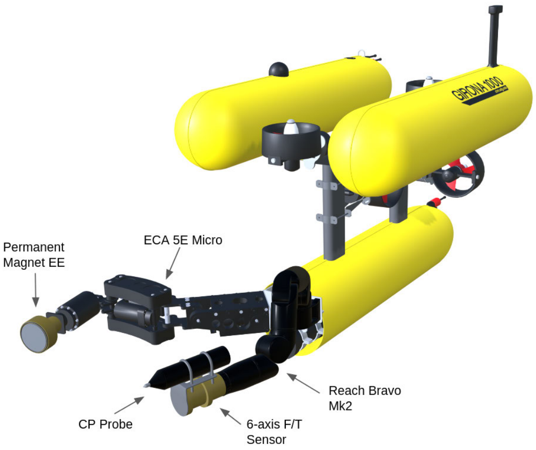


FIGURE 3. **Girona1000 equipped with the intervention payload.**

Due to the highly restricted range of motions possible to achieve with the ECA manipulator, its mounting configuration has been prepared to maximize the workspace when the ECA manipulator is attached to a vertical surface, like the floating wind turbine cyliders. However, this design decision is not optimal for folding the manipulator when not in use.

Figure 3 shows the Girona1000 AUV equipped with two manipulators for the proposed inspection task.

B. WINDFLOAT PILOT STRUCTURE

To test our proposed strategy, we constructed a pilot structure, a panel with a surface curvature similar to the one of the submerged cylinders of the WindFloat Atlantic, installed in Viana do Castello (recall Fig. 1), fitting inside the CIRS test tank (see Fig. 4). It was made of carbon steel for its ferromagnetic properties and protected from corrosion with a thin layer of epoxy paint. Additionally, it includes a true to life ladder and can be augmented with a set of magnetic ArUco markers [21], which are used to simplify the localization of the structure at early stages of development. Note that while the use of magnetic ArUco markers simplified the initial localization of the structure in our experiments, it is recognized that for broader applications and more intricate real-world scenarios, the development of enhanced structure recognition capabilities becomes increasingly important.

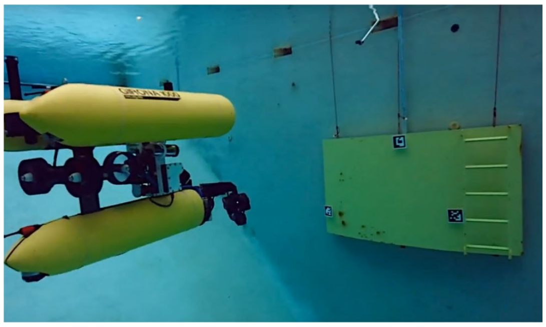


FIGURE 4. **Pilot structure emulating a section of a wind farm, at the CIRS test tank.**

IV. CONTROL ARQUITECTURE

The control architecture of the robot is schematically presented in Fig. 5. It is implemented using the Robot Operating System (ROS1) middleware, running on Ubuntu Linux.

The high-level mission is programmed using behavior trees [22] and interacts with the whole system sending commands to the different agents. The control layer is based on Task Priority [23], which can satisfy the goals of multiple control tasks by exploiting the system's redundancy. In TP, tasks are assigned priorities and controlled hierarchically to ensure that higher-priority tasks are achieved first, and lower-priority tasks are achieved only if they do not interfere with the higher-priority tasks (see Section V). The mission controller can reorder and switch on and off tasks, send them setpoints, and monitor their execution. One important task is the contact force task, which, based on readings from the force-torque sensor, controls the contact force of the probe w.r.t. the structure during the probing operation. The output of the control layer is sent to the manipulators through velocity

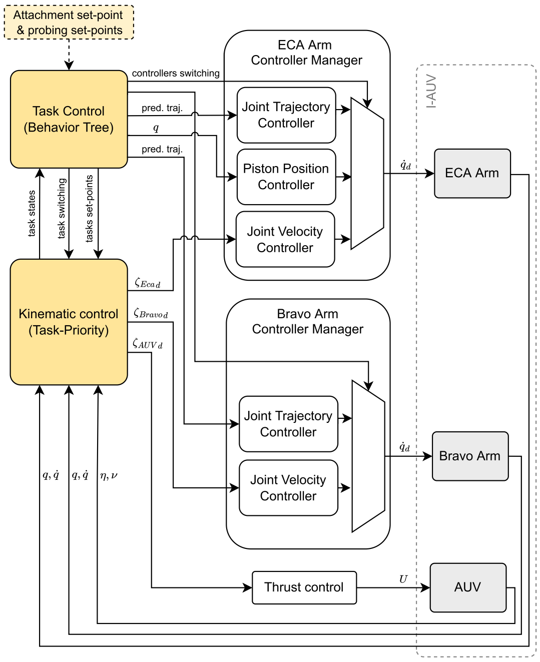


FIGURE 5. **Scheme for the control system of the Girona1000 I-AUV. The symbols used in the drawing are explained in the text. Work presented in this paper concerns the yellow coloured blocks.**

controllers, while the vehicle is controlled either in velocity or thrust.

The control interface for both manipulators is based on the ros_control package [24], which provides a standard interface and implementation for low-level controllers. A controller manager manages hardware resources and allows switching controllers on and off at runtime while handling resource conflicts between controllers. Both manipulators have a joint group velocity controller and a joint group trajectory controller. The velocity controllers control each arm in joint velocities and are mainly used by the task priority kinematic controller (see Section V). The trajectory controllers allow executing joint-space trajectories, defined by a set of position and velocity setpoints, to be reached at specific time instants. A cubic spline interpolator generates smooth and continuous motions with continuity guarantees at the velocity level. The high-level mission controller uses joint trajectory controllers to follow safe, offline generated trajectories for folding and unfolding of the manipulators. Additionally, a single joint position controller controls the piston of the ECA arm, to expose or hide the magnetic end-effector.

V. TASK PRIORITY KINEMATIC CONTROL

In this section, we aim to highlight the essential elements of TP that are pertinent to our research, while directing readers to our previous work [6], [14] for a more detailed description and formulation.

TP is based on a hierarchical control framework, where each task is assigned a priority level, and the robot's motion is controlled hierarchically to ensure that higher priority tasks are achieved first, while lower priority tasks are only achieved if they do not interfere with the higher priority ones.

Tasks are designed to achieve goals, such as reaching an EE pose, keeping a joint position fixed or avoiding joint limits. The mathematical foundation of TP is based on the concept of null space projection. Given a set of tasks with priorities $\boldsymbol{\sigma_1}, \boldsymbol{\sigma_2}, \ldots, \boldsymbol{\sigma_n}$, where $\boldsymbol{\sigma_1}$ has the highest priority and $\boldsymbol{\sigma_n}$ has the lowest priority, the null space projection is used to compute the desired system velocities $\boldsymbol{\zeta} = [\boldsymbol{\nu^T} \ \boldsymbol{\dot{q}^T}]^T$ that satisfy the higher priority tasks while also leaving the degrees of freedom associated with the lower priority tasks free to move. Two categories of tasks can be identified: 1) Equality tasks, whose goal is to drive the task variable to a desired value (e.g., EE pose task), and 2) Set tasks (also called inequality tasks) [25], [26], whose goal is to keep the task variable within a specified range (e.g., manipulator joint limits task).

Equalities encompass regulation tasks and tracking tasks. Regulation tasks aim to reach specific system states or conditions, while tracking tasks involve following desired trajectories with time constraints.

Set tasks are scalar regulation-only tasks used to keep a task variable σ within a specific range. These tasks activate only when the σ overpasses its limits to push σ back to the admissible set. Then, once the task variable σ has been successfully guided back within the predefined range, the set task deactivates.

Typical set tasks reported in the literature include Joint Limits, EE obstacle avoidance, and Minimum Altitude / Depth, to name but a few. The most representative task of this type is probably the one devoted to ensuring that the joint variables remain within their boundaries.

TABLE 1. Task priority hierarchy used in the presented work.

Hierarchy	Description	Type
Task 1	Joint Limits	Set Task
Task 2	AUV Position	Regulation Task (Position)
Task 3	AUV heading	Regulation Task (Orientation)
Task 4	AUV velocity	Tracking Task (Pose3D)
Task 5	ECA EE Pose	Regulation Task (Pose 3D)
Task 6	Bravo EE Pose	Regulation Task (Pose 3D)
Task 7	Bravo EE Compliant Pose	Regulation Task (Pose 3D)

Table 1 depicts the tasks used in the presented work and their associated priority. The topmost task is devoted to safety, maintaining joint positions within their defined limits. Tasks 2 & 3 control the AUV position and heading and are generally used for navigation. Position and heading are separated since there might be situations where it is desired to keep the heading (e.g., ensuring visibility of the target) while performing tasks other than navigation. Task 4 controls the vehicle's velocity, and is only used shortly after the robot attempts to attach to the structure, pushing the robot backward to test if the attachment succeeded. This task is configured to have the regulation component disabled (i.e., zero gain); therefore, it only has the feedforward component.

Tasks 5, 6, and 7 control the EE 3D poses of the ECA and Bravo manipulators. Task 5 is activated to attach to the structure, while tasks 6 and 7 coordinate to perform the probing operations. In particular, task 6 is devoted to moving the Bravo EE to a pre and post-operation pose, while task 7, whose input is regulated by an adaptive admittance controller, performs the touch operation.

VI. CONTACT FORCE CONTROL

A. ADAPTIVE ADMITTANCE CONTROL

The admittance control scheme [27] used to control the pose of the Bravo EE and the contact force exerted during the touch operation is based on the concept of a compliant frame $\bar{\eta}_{ee}$ attached to the stiff desired EE frame $\eta_{ee,d}$ by a virtual linear mass-spring-damper system, with dynamics described by:

$$\begin{aligned} M\ddot{\tilde{\eta}}_{ee} + K_D\dot{\tilde{\eta}}_{ee} + K_S\tilde{\eta}_{ee} &= h_{ee} \\ M\bar{\eta}_{ee} &= \eta_{ee,d} + \tilde{\eta}_{ee} \end{aligned} \tag{1}$$

where M, K_D, K_S are impedance parameters (constant, diagonal matrices), $\tilde{\eta}_{ee}$ is the pose error between the stiff and the compliant EE frame, and $h_{ee} = [f_{ee}^T \ \tau_{ee}^T]$ is the EE contact wrench, measured by the FT sensor installed in the manipulator's wrist.

Thus, the desired pose deformed by the admittance control layer is input to Task 7 during the touch operation.

Since the original algorithm lacks direct control over the contact wrench value, the stiffness constant K_S is replaced with a variable stiffnes, given by:

$$K_S(t) = K_P \int (h_{ee,d}(t) - h_{ee}(t))\, dt \tag{2}$$

where $h_{ee,d}$ is the desired EE wrench and K_P is a diagonal gain matrix. All impedance parameters, together with the initial K_S and K_P are tuned experimentally.

B. FEEDFORWARD FORCE EXTENSION

It is worth noting, that when the robot is exerting a force on the inspected surface, using the probing manipulator, a reaction force and torque are generated. These reactions should be compensated by the attraction force of the permanent magnet end-effector, used for the attachment. However, the magnet was not strong enough to hold the robot in place, mainly due to the long lever arm between the probing and the attachment locations, which results in a significant torque. Therefore the robot may detach in an uncontrolled way during probing. To counteract this problem we introduced an extension to the control system to compensate the reaction wrench using the thrusters of the I-AUV.

We can model the described situation by assuming that the whole robot can be treated as a single body, rigidly attached at one point to the inspected surface, and a force perpendicular to the inspected surface is exerted at the tip of the probe. Assuming that our goal is to reach a certain contact force, at the tip of the probe, we can consider this an equilibrium state and solve for a static balance of forces and torques in the model, presented in Fig. 6. This balance can be formulated according to the following system of equations:

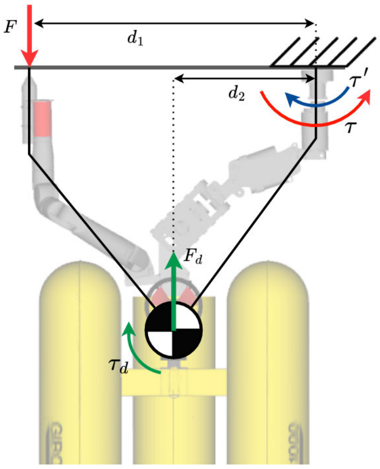


FIGURE 6. Force/Torque compensation during inspection.

$$\begin{aligned} F_d - F &= 0 \\ \tau_d - \tau + \tau' &= 0, \end{aligned} \tag{3}$$

where F is the force measured by the FT sensor (reaction to the probing force) and τ is the torque that this force generates with respect to the attachment point. To counteract the reactions we have to generate force F_d and torque τ_d, in the body frame, using the robot's thrusters. It is important to notice that the force generated by the thrusters additionally produces a torque with respect to the attachment point, designated by τ'. Then, the desired wrench can be calculated as follows:

$$\begin{aligned} F_d &= F \\ \tau_d - F d_1 + F_d d_2 &= 0 \\ \tau_d &= F(d_1 - d_2), \end{aligned} \tag{4}$$

where d_1 and d_2 are distances defined in Fig. 6. Due to the fact that the wrench control using the thrusters is not precise, we introduce a factor that defines the percentage of the required wrench that is compensated in this manner. We assume that the rest of the reaction forces and torques are compensated by the permanent magnet gripper and we tune the system so that the detachment does not occur.

VII. BEHAVIOR TREES

Behavior Trees (BT) allow for efficient switching between actions and modes of operation by providing a flexible and modular framework for modeling complex decision-making processes [28].

A BT is a directed rooted tree in which nodes are classified as leaves, composites, and decorators as described below. The edges of the tree represent the flow of control between the nodes. The terminology of parent and child is generally used to refer to connected nodes. A BT is executed starting from the root, which sends ticks at a certain frequency to its child. When a node receives a tick, it performs a specific action and returns a status to its parent node, which may be *Success*, *Failure*, or *Running*. Hereafter it is described how the most common types of nodes handle the tick and process the return statuses.

Composite nodes have one or more children and are used for organizing and coordinating the child nodes. The most common composite nodes are the Sequence and the Fallback. A Sequence node executes all its children in order as long as they return *Success*. It only returns *Success* if all its child nodes return *Success*. If one child returns *Failure* or *Running*, the node immediately stops and returns the same status to its parent. A Fallback node executes all its children in order as long as they return *Failure*. It returns *Failure* to its parent if all its child nodes return *Failure*. If one child returns *Success* or *Running*, the node immediately stops and returns the same status to its parent.

Decorator nodes can only have one child, and they modify or add functionality to it. For example, decorator nodes can be used to invert the status of their child node, set a time limit for execution, or perform other modifications to its behavior. Some examples include the inverter, repeater, and conditional nodes.

The nodes that perform the actual actions or evaluate conditions are called leaves. Leaves are often differentiated by actions and conditions. The main difference between them is that Condition nodes can only return *Success* or *Failure* within a single tick, whereas Action nodes can span multiple ticks, returning a *Running* status until they reach a terminal state. Generally, condition nodes represent simple checks (e.g., "is the door open?"), while action nodes represent complex actions (e.g., "open the door").

In addition to these types of nodes, BTs can also make use of a data structure called the Blackboard. The Blackboard is a shared memory space that allows nodes in the tree to access and modify information relevant to the overall behavior of the agent. It is similar to a global variable that can be read and written by different nodes. Nodes can read from and write to the Blackboard as needed, allowing them to make decisions based on the state of the world or coordinate their actions with other nodes in the tree.

In this paper, when representing graphically BTs, composite nodes are colored in red, decorator nodes are colored in yellow, action nodes are colored in blue, and sub-trees are colored in gray. Each node in the tree has clearly defined inputs and outputs, including any Blackboard variables that are used. When a Blackboard variable is used as input or output, it is expressed using the notation {*variable_name*}.

A. COMPOSITE NODES WITH MEMORY

In a standard Sequence or Fallback node, if a child node returns *Running* status and the tick ends, the next tick will start the evaluation of the child nodes from the beginning. Although this property stands for the principle of reactiveness, it may not be desirable in some situations and can result in unnecessary re-evaluation of nodes, slowing down the execution of the tree. For this reason, modern implementations include versions of composite nodes with memory. These nodes keep track of which child nodes were already evaluated during the previous tick and continue the evaluation from the last executed node, preventing unnecessary ticking in circumstances where reactiveness is not desired. Algorithms 1 and 2 show the pseudocode of the standard sequence and the sequence with memory respectively. An internal variable is maintained in the sequence with memory to point at the child that previously reported *Running*. The memory is reset if all children succeed or if a single child fails. In this work, we will refer to composite nodes without memory as Reactive, following the convention adopted in "BehaviorTree.CPP" [22], which is the library chosen by the authors to implement BTs. For instance, a Sequence with memory will be referred to as a **Sequence**, and a Sequence without memory will be referred to as **ReactiveSequence**.

Algorithm 1 Sequence (W/O Memory)

```
// child[] is an array of children nodes
for i = 1 to N (number of children) do
    childStatus = child[i].Tick()
    if childStatus == RUNNING then
        return RUNNING
    else if childStatus == FAILURE then
        return FAILURE
// All children succeeded
return SUCCESS
```

Algorithm 2 Sequence (W/ Memory)

```
// child[] is an array of children nodes
// idx is a private variable
while idx < N (number of children) do
    childStatus = child[idx].Tick()
    if childStatus == SUCCESS then
        idx ← idx + 1                    // increment idx
    else if childStatus == RUNNING then
        return  RUNNING                  // keep same idx
    else if childStatus == FAILURE then
        idx ← 0                          // reset idx
        return FAILURE
// All children succeeded
idx ← 0
return SUCCESS
```

It should be noted that the decision to use composite nodes with memory depends on the specific requirements and desired behavior of the application. While reactiveness is an important principle, it may not always be necessary or appropriate. Ultimately, it is up to the programmer to decide which parts of the BT should be reactive and which should have memory based on the specific needs and constraints of the system.

B. ADVANTAGES OF USING BEHAVIOR TREES

BTs have become a popular alternative to Finite State Machines (FSMs) for designing intelligent agent behaviors. One key advantage is their modularity, which allows for creating complex behaviors by combining simpler ones. BTs provide a more flexible way of handling complex behaviors with conditions, loops, and priorities. Additionally, BTs are more expressive than FSMs, as they can represent complex behaviors with multiple states and actions in a more compact and organized way. BTs also make it easy to modify or add new behaviors without affecting the rest of the tree. Overall, BTs provide a powerful and modular approach to designing complex behaviors for intelligent agents, making them a popular alternative to FSMs.

VIII. MISSION BEHAVIORS

A. ACTIONS AND CONDITIONS

1) GOTO POSE ACTION

When the **GoTo Pose** action is executed, it receives the desired position and orientation (through the Blackboard), as well as the name of the task in the TP hierarchy. The action then sends the resulting setpoint to the corresponding task through the ROS framework, which updates the control commands sent to the robot. The action also takes in a tolerance and timeout parameters. The tolerance specifies the maximum acceptable error between the current pose and the desired pose. If the current pose of the task target link is within the desired tolerance, the action returns a *success* status. If the controller is unable to achieve the desired pose within the given tolerance before the timeout expires, the action returns a *failure* status.

Additionally, a goal offset parameter might be provided to the **GoTo Pose** action, which specifies an offset to the desired pose. The goal offset is added to the desired pose before sending the setpoint to the task. This is used to perform an approach, touch, and retreat sequence given a single setpoint.

2) GOTO COMPLIANT POSE ACTION

The **GoTo Compliant Pose** action is similar to the **GoTo Pose** action, but it additionally takes the desired force and its tolerance as input parameters through the Blackboard. The action then sends the pose and force setpoints to the task, and monitors that the current pose and force applied are within the specified tolerances. If the task is able to achieve the desired pose and force within the specified tolerances before the timeout expires, the action returns a *success* status.

3) SWITCH TASKS ACTION

The **Switch Tasks** action is responsible for enabling and disabling tasks in the TP hierarchy (see Table 1). The action takes as input a set of tasks to be enabled and a set of tasks to be disabled and updates the corresponding tasks accordingly. Note that, once a task is achieved it remains activepopular alternativ (e.g., keeping an EE pose), so it is important to coordinate the switching of tasks during the mission. However, the use of the **Switch Tasks** action is not always explicitly shown in the figures to avoid cluttering the diagrams, since generally there is only one task active at the time (besides joint limits).

4) SWITCH CONTROLLERS ACTION

The control drivers of the manipulators are based on the ROS control framework. The **Switch Controllers** action is responsible for switching between different ROS controllers that control the manipulator's joints. The action takes as inputs the controllers to turn on and off and uses the ‘‘switch_controllers’’ service provided by the controller manager to switch the controllers.

This action is generally used to switch between the Joint Velocity Controllers and the Joint Trajectory Controllers. The former is used by the TP Kinematic Control layer to drive the manipulators, while the latter is used to follow pre-defined trajectories for folding and unfolding the manipulators. However, the use of the **Switch Controllers** action is not always explicitly shown in the figures to avoid cluttering the diagrams.

5) EXTEND/RETRACT MAGNET ACTIONS

The **Extend/Retract Magnet** actions are responsible for extending or retracting the piston of the ECA manipulator to slide the magnet in or out of the EE frame. Each action sends a predefined position setpoint to the piston position controller using the ros_control framework.

6) FOLLOW MANIPULATOR TRAJECTORY ACTION

The **Follow Manipulator Trajectory** action is responsible for sending a predefined trajectory to the Joint Group trajectory controller of the requested arm using the ros_control framework. This action is used to execute complex motion sequences that are planned in advance (e.g., fold / unfold operations). The action takes the manipulator name and the desired trajectory as inputs and sends it to the corresponding Joint Group trajectory controller, which then generates the corresponding control commands for the robot's joints. Once the trajectory has been executed, the action returns a *success* status. However, if the trajectory cannot be executed due to any reason (e.g., joint limits, controller not loaded), the action returns a *failure* status.

7) MAGNET CONTACT DETECTED CONDITION

The **Magnet Contact Detected** Condition is responsible for detecting whether the robot has made contact with the structure using the end-effector with a permanent magnet. Since there is no force-torque sensor on this manipulator, the current velocity of the robot is monitored to detect a sudden drop as an indicator of a collision with the structure.

8) LOCALIZE STRUCTURE ACTION

The **Localize Structure** action is responsible for detecting the position of the structure. The detection of the ArUco markers are used to fit a cylinder, which is assumed to be vertical (i.e., the cylinder's axis is aligned with the direction of gravity) and its radius known a priori. The problem can be formulated using non-linear least squares as follows:

$$\begin{aligned} &\underset{\boldsymbol{c}}{\operatorname{argmin}} \sum_{i=1}^{n} \|f_i(\boldsymbol{c})\|^2 \\ &f_i(\boldsymbol{c}) = r \; - \; \left\| \begin{bmatrix} c_x - a_{i,x} \\ c_y - a_{i,y} \end{bmatrix} \right\| \end{aligned} \tag{5}$$

where r is the known radius of the cylinder, $\boldsymbol{c}$ is the cylinder's center in the NED frame, and $\boldsymbol{a}_i$ is the location of the i-th ArUco marker in the NED frame. Note that since the z coordinate of the cylinder cannot be optimized, the $(\boldsymbol{c} - \boldsymbol{a}_i)$ error is projected onto the XY plane. Then, the depth of the structure is assumed to be referenced to the topmost ArUco marker.

B. TREES

The top-level behavior tree controls the overall mission and coordinates the execution of the subtrees. The mission consists of 6 phases: First, the robot locates the structure (1) and defines the probing and attachment points. Then it navigates close to the structure (2). Next, the robot must dock to the structure (3), probe the set of probing points (4), detach (5), and finally, surface (6). One could imagine the mission defined using only a sequence node, as shown in Fig. 7.

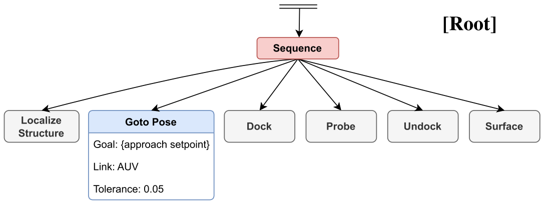


FIGURE 7. **Behavior Tree of the general mission.**

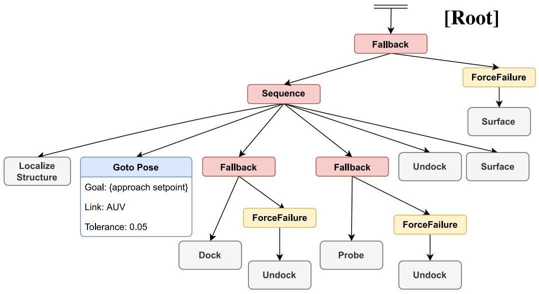


FIGURE 8. **Behavior Tree of the general mission with recovery structures.**

However, it is essential to contemplate recovery behaviors in case the robot fails to execute some task. Figure 8 depicts the top-level mission with recovery behaviors. Note that recovery actions are decorated with a **ForceFailure** that returns *Failure* regardless of the return status of the child.

1) DOCKING/UNDOCKING BEHAVIOR TREES

The docking procedure (see Fig. 9) aims to anchor the robot to the target structure using the ECA's magnetic tip, providing stability to the system before performing any intervention task. The task starts by unfolding the ECA arm and extending the magnet out. Then, the ECA EE is commanded to reach the docking setpoint, using a **Goto** action. While the robot moves toward the goal, it reactively checks if contact with the structure is detected. The setpoint *{attach_setpoint}*, is a global variable set inside the **Localize Structure** sub-tree. In order to ensure contact, the setpoint has a slight offset towards the center of the structure. Note that this prevents the **Goto** action from succeeding, but the **Reactive Fallback** will succeed as long as contact is detected. Also note that if contact is never detected, a **Timeout** decorator will stop the execution and report *FAILURE*. After detecting contact, the robot tests the attachment. Finally, it disables the AUV surge, sway, and yaw velocity controllers. Note that the surge is still controlled to avoid stress on the end-effector caused by the robot's buoyancy.

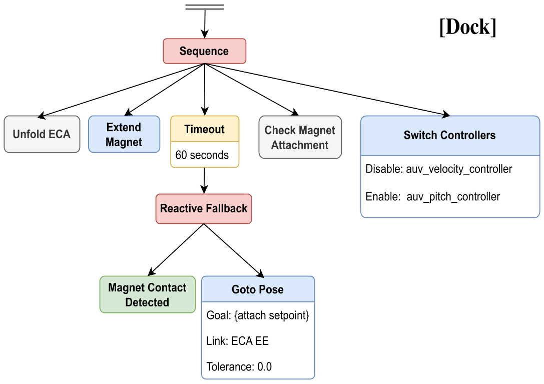


FIGURE 9. **Behavior Tree of the docking procedure.**

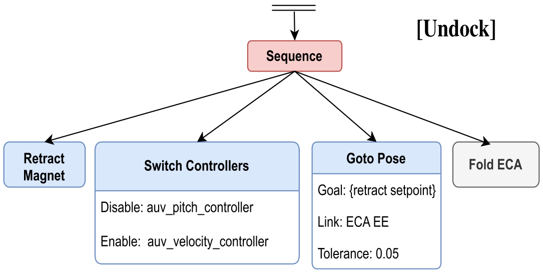


FIGURE 10. **Behavior Tree of the undocking procedure.**

The undocking procedure, depicted in Fig. 10, starts by retracting the magnet. Once detached, it starts the AUV velocity controllers and it commands the AUV to a retreat position. Finally, it folds the ECA manipulator.

2) FOLDING/UNFOLDING MANIPULATORS

The BTs responsible for the folding and unfolding of the manipulators combines the use of the **Switch Controllers** and **Follow Trajectory** actions. When the mission requires folding or unfolding of the manipulators, the **Switch Controllers** action is used to switch from the joint velocity controller to the joint trajectory controller. This is followed by the **Follow Trajectory** action, which sends a predefined trajectory to the joint group trajectory controller to perform the preprogrammed motion sequence. Once the motion is complete, the **Switch Controllers** action is used again to switch back to the joint velocity controller, allowing the TP Kinematic Control layer to drive the manipulators. By using a combination of these two actions, the behavior tree can efficiently execute folding and unfolding tasks without having to write custom code for each specific motion sequence (see Fig. 11).

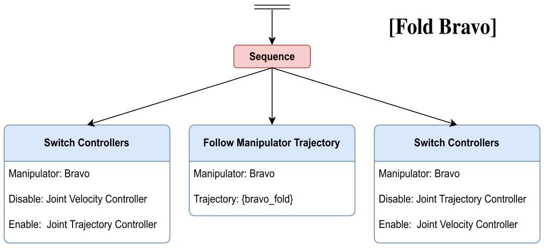

[Fold Bravo]
Sequence
Switch Controllers
Manipulator: Bravo
Disable: Joint Velocity Controller
Enable: Joint Trajectory Controller
Follow Manipulator Trajectory
Manipulator: Bravo
Trajectory: {bravo_fold}
Switch Controllers
Manipulator: Bravo
Disable: Joint Trajectory Controller
Enable: Joint Velocity Controller

FIGURE 11. Behavior Tree of the folding of the Bravo manipulator.

3) PROBING BEHAVIOR TREE

Given a set of points defined in the structure's surface, the probing procedure (see Fig. 12) aims to precisely touch each point using the CP probe mounted on the Bravo arm to perform measurements. Touching a point is structured in three phases: First, using a **Goto** action, the probe is commanded in a favorable pose close to the setpoint. Then, a **Goto Compliant** action is enabled, which uses the adaptive admittance control explained in Section VI to guarantee contact with the structure's surface during the measurement. The setpoints are stored in a stack. The node **Pop Setpoint** pops a setpoint off the stack and stores it in the *{probe_setpoint}* variable, and the node **IsSetpointsStackEmpty** is a condition that returns *Success* if the stack is empty and *Failure* if it is not. These nodes enable the capacity to iterate through the setpoints in a while-loop fashion, using a **Repeat until failure**. This node is wrapped with a **Fallback** to check if the *Failure* was due to an error during the probing (*Failure*) or due to an empty stack (*Success*).

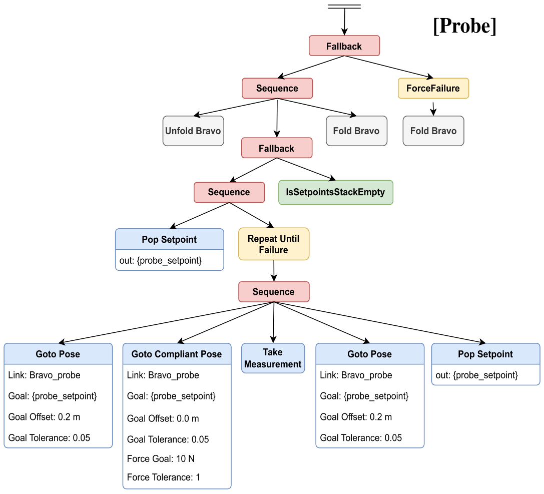


FIGURE 12. Behavior Tree of the probing procedure. The predefined probing setpoints are stored in a stack that pops a setpoint to the Blackboard on every iteration until the stack is empty.

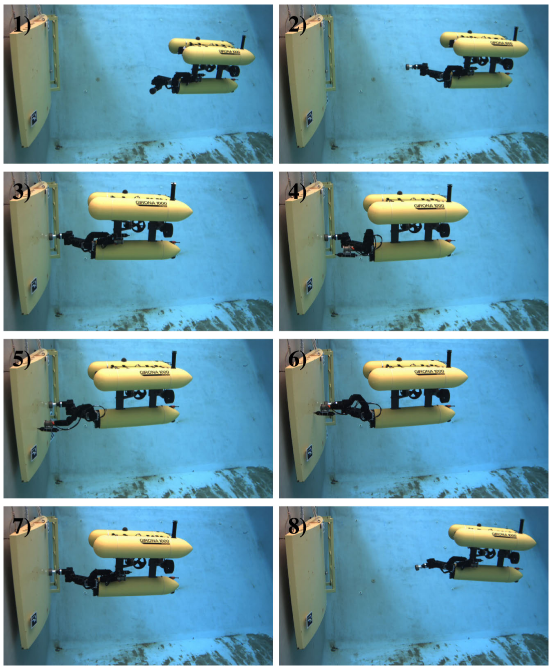


FIGURE 13. The Girona1000 I-AUV performing the CP survey operation sequence, at the CIRS water tank. In 1), the Girona1000 as detected the structure and approached a starting pose. In 2), the ECA manipulator unfolds and the piston extends to expose the magnetic EE. In 3), the I-AUV has successfuly attached to the structure. In 4), the Bravo manipulator unfolds to perform CP measurements (5,6). In 7), the Bravo manipulator folds back to, next, detach from the structure (8).

IX. EXPERIMENTAL VALIDATION

A. RESULTS

This section presents the results obtained during the experiment at the CIRS water tank, using the mockup structure presented in Section III-B. A video demonstrating the whole mission in the CIRS water tank can be found at the following URL[1]: https://youtu.be/vdijAgp6z8w.

[1] www.youtube.com/@cirsudg

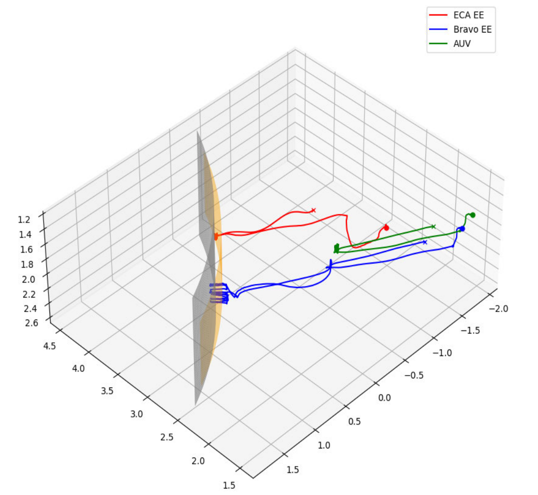


FIGURE 14. Trajectory of the I-AUV during the executed mission. Shown are the Girona1000 path (green), the ECA EE (red), the CP probe mounted on the Bravo (blue), the structure's diameter with which the setpoints where generated (orange), and the real structure's diameter (gray). The start of the trajectories is marked with a circle and the end is marked by a cross.

Figure 13 shows a sequence of eight snapshots from the real experiment, showcasing the I-AUV in action as it autonomously approaches, attaches to the structure, performs the CP survey, and detaches.

The trajectory of the I-AUV and the end-effector of both manipulators during the mission are shown in Fig. 14. The plotted trajectories displays the attaching, probing, and retreat, being part of the beginning and end removed to ensure clarity.

Figure 15 illustrates how the different BTs coordinate the switching of TP tasks and the manipulators' controllers.

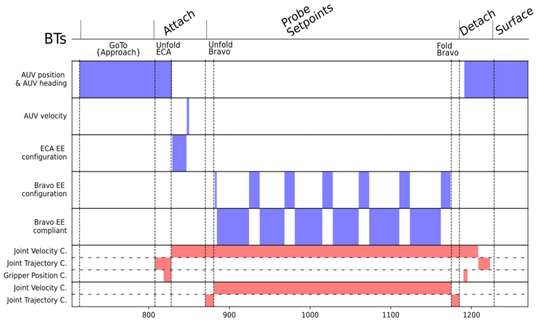


FIGURE 15. Diagram of the TP tasks and controllers activation by the different behaviors.

Figure 16 presents data from the moment of attaching to the structure. It shows how the impact of the magnetic gripper with the structure was detected by monitoring the derivative of the I-AUV's current velocity while it was approaching the structure. The sudden drop in velocity can only be explained by a collision, which indicates that the I-AUV has successfully touched the structure. Nonetheless, after the touch occurs, the I-AUV tests if the attachment was successful by attempting to move backward (see the activation of the AUV velocity task in Fig. 15).

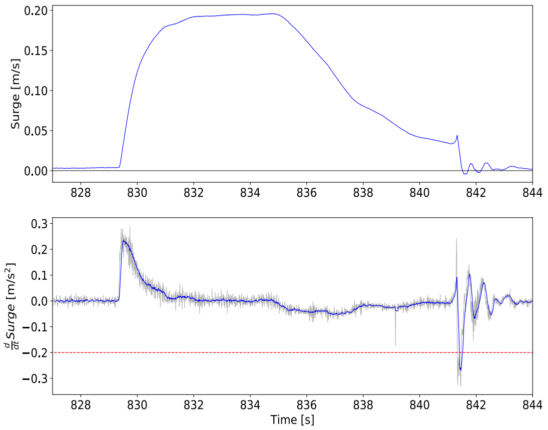


FIGURE 16. Detection of the magnet contact to the structure. Top: AUV surge velocity during the operation. Bottom: The filtered surge derivative (blue) and the trigger threshold (red).

For the inspection operation, the I-AUV was commanded to maintain contact between the tip of the CP probe and the surface of the structure at 6 different points. To ensure that the contact always occurs, no matter the errors in the localization of the structure, the setpoints were generated behind the surface. This was done by assuming that the structure's diameter was smaller than in reality. The compliant control algorithm ensures that a safe approach to these setpoints is possible. Figure 17 shows the measured and target contact forces between the probe and the structure, as well as the desired and actual locations of the probe tip. It can be appreciated how the setpoint location converges to the current location of the tip during the contact. It can be noticed that the force control oscillates significantly. This effect occurs due to the fact that the thrusters used in the vehicle are not possible to control in force but only using a synthetic setpoint, which is not directly related to any physical quantity. The relation between the thrust and setpoint was roughly identified using a static thrust experiment and encoded by polynomial functions. However, these functions do not account for the dynamical effects, occurring during navigation and when the thrusters change direction of rotation frequently. The second situation is common when trying to control the contact forces precisely. Moreover, the propeller design used in the thrusters is not optimized for positioning but rather speed.

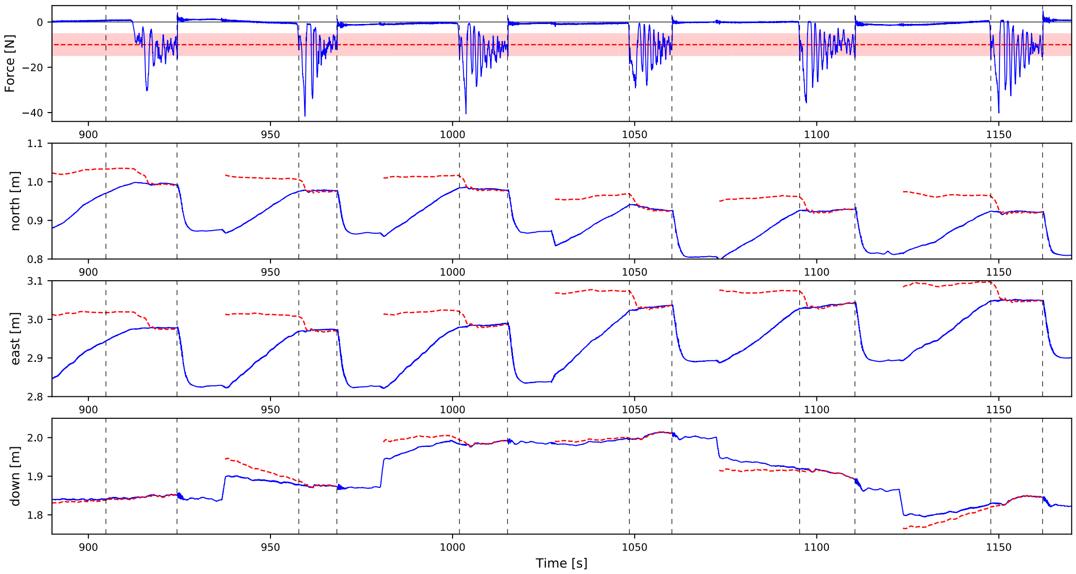


FIGURE 17. On top, tracking of the desired contact force along the z-axis of the EE The blue line is the measured force, the red dashed line is the desired force, and the red area is the region within the force threshold. The remaining plots denote the X, Y, and Z position of the EE (blue) and its target (dashed red), which is deformed by the adaptive force controller. The vertical dashed lines show the duration of contact.

X. CONCLUSION

The paper presents a complete system for an I-AUV to perform inspection of marine structures, which requires contact force control, as is the case of CP inspection for offshore wind turbines. The use of behavior trees allowed for the automation of the mission, simplifying the coordination of the different stages of the mission. The impact detection strategy, based on monitoring the approach velocity, provided a feasible solution when force measurement was not available. The attachment to the structure, by using a permanent magnet EE, improves the accuracy of the intervention operations requiring contact. The control ideas are based on a combination of admittance force control and the Task Priority control framework. The admittance control was modified to allow for direct control over the contact force value, by introducing a self-adjusting stiffness matrix. Moreover, the algorithm generates a feed-forward force to be applied by the vehicle's thrusters to counteract the reaction force at the contact point, preventing the detachment of the robot from the structure. The system was validated in the CIRS water tank using a mockup of a cylindrical pylon of an offshore wind turbine.

Overall, the system presented in this paper demonstrates the potential of using I-AUVs for efficient and accurate inspection of marine structures, which can have significant implications for the maintenance and safety of offshore infrastructures.

REFERENCES

[1] P. Ridao, M. Carreras, D. Ribas, P. J. Sanz, and G. Oliver, "Intervention AUVs: The next challenge," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 12146–12159, 2014.

[2] E. Zereik, M. Bibuli, N. Mišković, P. Ridao, and A. Pascoal, "Challenges and future trends in marine robotics," *Annu. Rev. Control*, vol. 46, pp. 350–368, 2018.

[3] G. Marani and J. Yuh, *Introduction to Autonomous Manipulation, Case Study With an Underwater Robot, SAUVIM*, vol. 102. Berlin, Germany: Springer, Apr. 2014.

[4] E. Simetti, G. Casalino, S. Torelli, A. Sperindé, and A. Turetta, "Floating underwater manipulation: Developed control methodology and experimental validation within the TRIDENT project," *J. Field Robot.*, vol. 31, no. 3, pp. 364–385, May 2014.

[5] E. Simetti, F. Wanderlingh, S. Torelli, M. Bibuli, A. Odetti, G. Bruzzone, D. L. Rizzini, J. Aleotti, G. Palli, L. Moriello, and U. Scarcia, "Autonomous underwater intervention: Experimental results of the MARIS project," *IEEE J. Ocean. Eng.*, vol. 43, no. 3, pp. 620–639, Jul. 2018.

[6] P. Cieslak, P. Ridao, and M. Giergiel, "Autonomous underwater panel operation by GIRONA500 UVMS: A practical approach to autonomous underwater manipulation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 529–536.

[7] A. Carrera, M. Carreras, P. Kormushev, N. Palomeras, and S. Nagappa, "Towards valve turning with an AUV using learning by demonstration," in *Proc. MTS/IEEE OCEANS*, Jun. 2013, pp. 1–7.

[8] D. Youakim, P. Ridao, N. Palomeras, F. Spadafora, D. Ribas, and M. Muzzupappa, "Autonomous underwater free-floating manipulation using MoveIt!" *IEEE Robot. Autom. Mag.*, vol. 24, no. 3, pp. 41–51, 2017.

[9] D. Youakim, P. Cieslak, A. Dornbush, A. Palomer, P. Ridao, and M. Likhachev, "Multirepresentation, multiheuristic A* search-based motion planning for a free-floating underwater vehicle-manipulator system in unknown environment," *J. Field Robot.*, vol. 37, no. 6, pp. 925–950, Sep. 2020.

[10] A. Palomer, P. Ridao, D. Youakim, D. Ribas, J. Forest, and Y. Petillot, "3D laser scanner for underwater manipulation," *Sensors*, vol. 18, no. 4, p. 1086, Apr. 2018.

[11] A. Birk et al., "Dexterous underwater manipulation from onshore locations: Streamlining efficiencies for remotely operated underwater vehicles," *IEEE Robot. Autom. Mag.*, vol. 25, no. 4, pp. 24–33, Dec. 2018.

[12] S. Lensgraf, A. Sniffen, Z. Zitzewitz, E. Honnold, J. Jain, W. Wang, A. Li, and D. Balkcom, "Droplet: Towards autonomous underwater assembly of modular structures," in *Proc. Robot., Sci. Syst.*, 2021.

[13] S. Lensgraf, D. Balkcom, and A. Q. Li, "Buoyancy enabled autonomous underwater construction with cement blocks," 2023, *arXiv:2305.05552*.

[14] R. Pi, P. Cieslak, P. Ridao, and P. J. Sanz, "TWINBOT: Autonomous underwater cooperative transportation," *IEEE Access*, vol. 9, pp. 37668–37684, 2021.

[15] S. Heshmati-Alamdari, G. C. Karras, and K. J. Kyriakopoulos, "A predictive control approach for cooperative transportation by multiple underwater vehicle manipulator systems," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 3, pp. 917–930, May 2022.

[16] P. Dai, W. Lu, K. Le, and D. Liu, "Sliding mode impedance control for contact intervention of an I-AUV: Simulation and experimental validation," *Ocean Eng.*, vol. 196, Jan. 2020, Art. no. 106855.

[17] S. Heshmati-Alamdari, C. P. Bechlioulis, G. C. Karras, A. Nikou, D. V. Dimarogonas, and K. J. Kyriakopoulos, "A robust interaction control approach for underwater vehicle manipulator systems," *Annu. Rev. Control*, vol. 46, pp. 315–325, 2018.

[18] P. Cieslak and P. Ridao, "Adaptive admittance control in task-priority framework for contact force control in autonomous underwater floating manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 6646–6651.

[19] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 AUV: From survey to intervention," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 1, pp. 46–53, Feb. 2012.

[20] W. J. Marais, S. B. Williams, and O. Pizarro, "Maximising wrenches for kinematically redundant systems with experiments on UVMS," 2022, *arXiv:2202.13535*.

[21] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image Vis. Comput.*, vol. 76, pp. 38–47, Aug. 2018.

[22] D. Faconti and M. Colledanchise. (2021). *Behaviortree.cpp*. [Online]. Available: https://www.github.com/BehaviorTree/BehaviorTree.CPP

[23] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 13, no. 3, pp. 398–410, Jun. 1997.

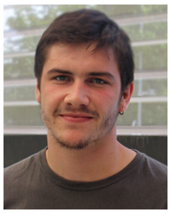
[24] S. Chitta, E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. R. Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtke, and E. F. Perdomo, "Ros_control: A generic and simple control framework for ROS," *J. Open Source Softw.*, vol. 2, no. 20, p. 456, Dec. 2017.

[25] E. Simetti and G. Casalino, "A novel practical technique to integrate inequality control objectives and task transitions in priority based control," *J. Intell. Robot. Syst.*, vol. 84, nos. 1–4, pp. 877–902, Dec. 2016.

[26] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results," *Frontiers Robot. AI*, vol. 3, p. 16, Apr. 2016.

[27] B. Siciliano, O. Khatib, and T. Kröger, "Force control," in *Springer Handbook of Robotics*, vol. 200. London, U.K.: Springer, 2017, pp. 363–405.

[28] M. Colledanchise and P. Ögren, *Behavior Trees in Robotics and AI*. Boca Raton, FL, USA: CRC Press, Jul. 2018.



ROGER PI received the B.S. degree in computer engineering from Universitat de Girona, Catalonia, in 2017, and the joint M.Sc. degree in computer vision and robotics from the University of Burgundy, Universitat de Girona, and Heriot-Watt University, in 2019. He is currently pursuing the Ph.D. degree with the Underwater Robotics Research Center (CIRS), under the supervision of Dr. Pere Ridao. His current research interest includes autonomous intervention for underwater vehicle manipulator systems (UVMSs). He received the Best Master Student Award from the University of Burgundy, Universitat de Girona, and Heriot-Watt University.

PATRYK CIEŚLAK received the Ph.D. degree from the Department of Robotics and Mechatronics, AGH University of Science and Technology, Kraków, Poland, in 2016. Since 2017, he has been with the Underwater Vision and Robotics Laboratory (CIRS), University of Girona. Initially as a Marie Skłodowska-Curie Postdoctoral Fellow, he developed control algorithms for autonomous non-destructive testing using an I-AUV. He is the co-author of a commercial rehabilitation robot called Prodrobot, which is a stationary lower limbs exoskeleton, used in the learning and improvement of natural gait patterns in children. He is the author of an advanced open-source robot simulator called Stonefish, designed for marine robotics. This simulator is currently used in all underwater-related research at CIRS and in other institutions around the world and can be found on (github.com/patrykcieslak). His current research interests include underwater manipulation, including obstacle avoidance, motion planning, and cooperative manipulation with two I-AUVs. He is continuing his research at CIRS, working on underwater intervention for inspection, maintenance, and repair (IMR) operations in offshore wind farms.

JOAN ESTEBA was born in Girona, in 1994. He received the bachelor's degree in industrial technologies engineering and the M.Sc. degree in industrial engineering from the University of Girona (UdG), in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the Computer Vision and Robotics Research Institute (VICOROB). He had been working for several years developing vision and robotic technologies in the automobile industry and leading different projects. He also had been leading some development projects in the food industry. His current research interest includes docking systems for autonomous underwater robotic systems. He is a Registered Professional Engineer in Catalonia.

NARCÍS PALOMERAS (Member, IEEE) was born in Vilafant, in 1981. He received the Ph.D. degree from the University of Girona, in 2011. He has participated in several research projects, all related to underwater robotics, both national and European (TRIDENT, PANDORA, MORPH, MERBOTS, LOONDOCK, TWINBOT, 3DAUV, and ATLANTIS...) and in different European competitions for AUVs, such as SAUC-E and ERL. Currently, he is the Coordinator of a Joint Degree Erasmus Mundus about intelligent field robotic systems. He is also a Postdoctoral Researcher with the Computer Vision and Robotics Research Institute (VICOROB), University of Girona. His current research interests include underwater robotics in topics, such as planning, exploration, intelligent control architectures, mission control, and localization.

PERE RIDAO (Member, IEEE) received the Ph.D. degree in industrial engineering from the University of Girona, in 2001. He was the Co-Founder of Iqua Robotics S.L., a spin-off company. He is currently the Head of the Underwater Robotics Research Center (CIRS) and a Full Professor with the Department of Computer Engineering, University of Girona. Since 1997, he has participated in 24 research projects (15 European and nine national). He is the author of more than 100 publications. He has supervised nine Ph.D. thesis (four more are currently under direction) and 14 M.S. Thesis. He is the coauthor of four licenses and one Spanish/European patent. His current research interests include designing and developing autonomous underwater vehicles for 3D mapping and intervention. He has served as the Chair for the IFAC's Technical Committee on Marine Systems.

5 Results and Discussion

In this chapter we discuss and extend the results presented in this thesis. First, in Section 5.1, we briefly summarize the results reported in each chapter. Then, we present unpublished experimental results in Section 5.2, which extends the work presented in Chapter 4 in sea trials at Viana do Castelo (Portugal).

5.1 Summary of the completed work

Although there is an increasing demand for underwater autonomous manipulation, there are still very few experimental demonstrations in this area of research. In this thesis, we have focused on advancing the state of the art in the field, experimentally validating two complete underwater intervention missions autonomously: a cooperative pipe transportation by two I-AUVs and a cathodic protection (CP) survey by a dual-arm I-AUV.

In Chapter 2, we presented a calibration procedure to estimate the intrinsic and extrinsic sensor parameters within the robotic system. The estimation procedure is based on nonlinear least-squares, modeled graphically using factor graphs. One key element of the solution is the robust formulation of the minimization equations leveraging Lie theory, with a special focus on the computation of Jacobians. The calibration procedure was experimentally validated on the Girona500 I-AUV.

In Chapter 3, we presented a decentralized cooperative transportation scheme for a team of I-AUVs, performing the complete sequence of pick, transport, and place operations. The approach focuses on a decentralized kinematic control of two I-AUVs, simultaneously controlling their EEs velocities to achieve a common goal, with minimal data transfer between robots. The proposed approach was validated through both simulated and water tank experiments. To the best of our knowledge, at the time of the experiment, only one group reported experimental results on this field on a smaller scale with custom-made robots.

In Chapter 4, we proposed a methodology for contact-based operation on semi-submergible structures. This methodology was applied in the context of CP inspection of a wind turbine structure. The Girona1000 I-AUV was equipped with two manipulators: one devoted to attach to the structure using a magnetic EE, and the other equipped with a CP probe to puncture the structure to take measurements. As the magnet can be detached from the structure during the probing operation due to the generated torque, the robot thrusters are actuated during the operation to compensate for the contact reaction forces. The proposed methodology was experimentally validated in a water tank using a mock-up structure.

5.2 Experiments

Apart from the results already presented in previous chapters, the work presented in Chapter 4 was also tested at sea at the ATLANTIS Test Centre.

The ATLANTIS Test Centre has been established in Viana do Castelo (Portugal) in the context of the ATLANTIS project [28] with the aim of providing a realistic environment to promote the use of robots in offshore IMR operations. The coastal testbed of the ATLANTIS Test Center is equipped with a floating structure that simulates the structure of an offshore floating wind turbine. The installed floating structure is a decommissioned Catenary Anchor Leg Mooring buoy, that provided support to the loading and discharging of liquid product cargo to/from tankers. The structure was modified, cleaned, and painted prior to deployment, and received the name of DURIUS (see Fig. 5.1). The pilot structure has a diameter of 16 meters and a height of 6 meters, and is anchored at the exit of the Lima river using three mooring chains (see Fig. 5.2).

Conducting inspection and maintenance activities on structures deployed at sea presents additional challenges, such as the accumulation of marine organisms, including algae, barnacles, and mussels. The accelerated growth of these organisms, collectively known as biofouling, not only compromises the structural integrity of offshore installations but also poses a considerable impediment to the efficacy of robotic intervention applications. Figure 5.3 shows the

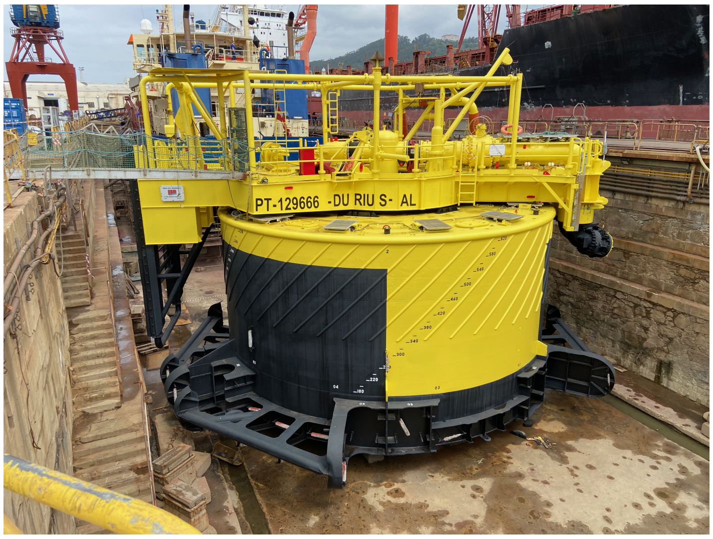
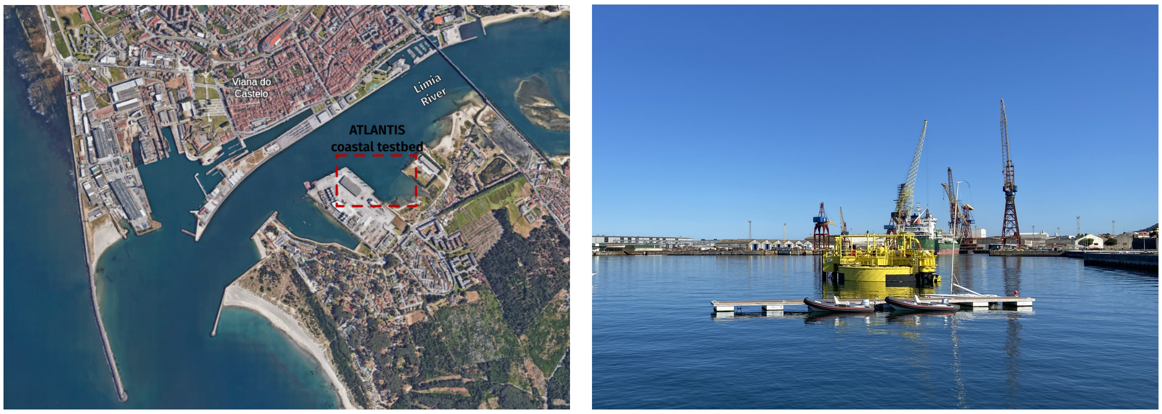


Figure 5.1: The DURIUS structure modified and painted before deployment



(a) The location of the coastal ATLANTIS Test Centre, in Viana do Castelo

(b) DURIUS deployed on the coastal ATLANTIS Test Centre

Figure 5.2: The ATLANTIS Test Centre

accumulated marine growth on the surface of the DURIUS structure that we encountered, after 3 months since its deployment at sea. In the context of the the CP testing presented in Chapter 4, the marine growth affects the proper puncturing of the CP probe and the efficacy of the magnet to couple with the structure. Hence, in the context of the project ATLANTIS, we also proposed the use of an electric brush to perform cleaning operations. We employed a comercial Nemo electric brush[1], submergible up to 50 meters depth, which we adapted to be powered and controlled through the Girona1000 I-AUV. Both the CP probe and the electric brush tools were designed to be interchangable on the Bravo EE. Figure 5.4 shows a picture of the electric brush mounted on the Bravo EE, coupled with the force-torque (FT) sensor, during the trials in Viana do Castelo.

Note that, as marine growth also affects the performance of the magnetic coupling, the cleaning operation needs to initiate with the robot floating. The methodology was proposed as follows (see Fig. 5.5):

[1] https://nemopowertools.com

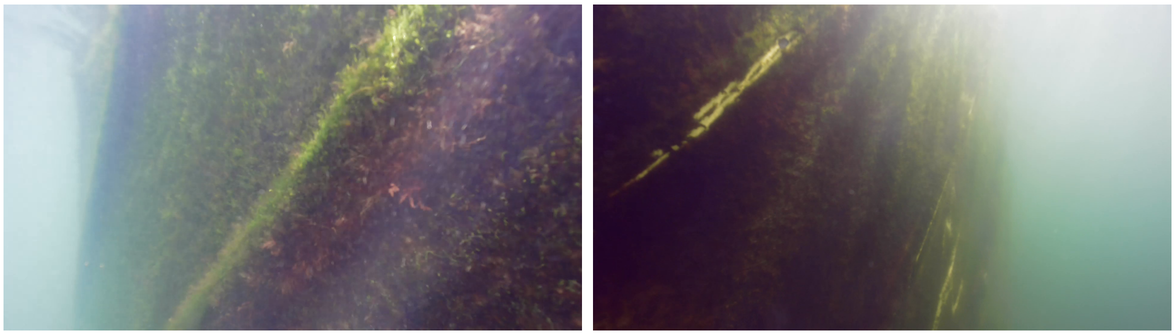


Figure 5.3: Two pictures of the marine growth on the surface of the DURIUS structure, accquired with the robot during the first day of trials.

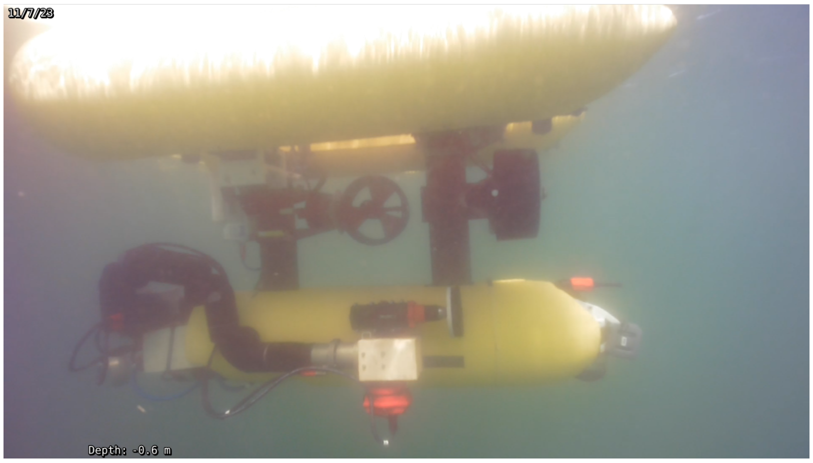


Figure 5.4: The Girona1000 I-AUV equipped with the electric brush tool.

1. The robot executes a preliminary cleaning while floating, creating a localized and debris-free surface to facilitate the attachment of the magnetic EE.

2. Subsequently, the robot engages with the structure using the magnetic EE within the cleaned area, ensuring a close contact with the structure.

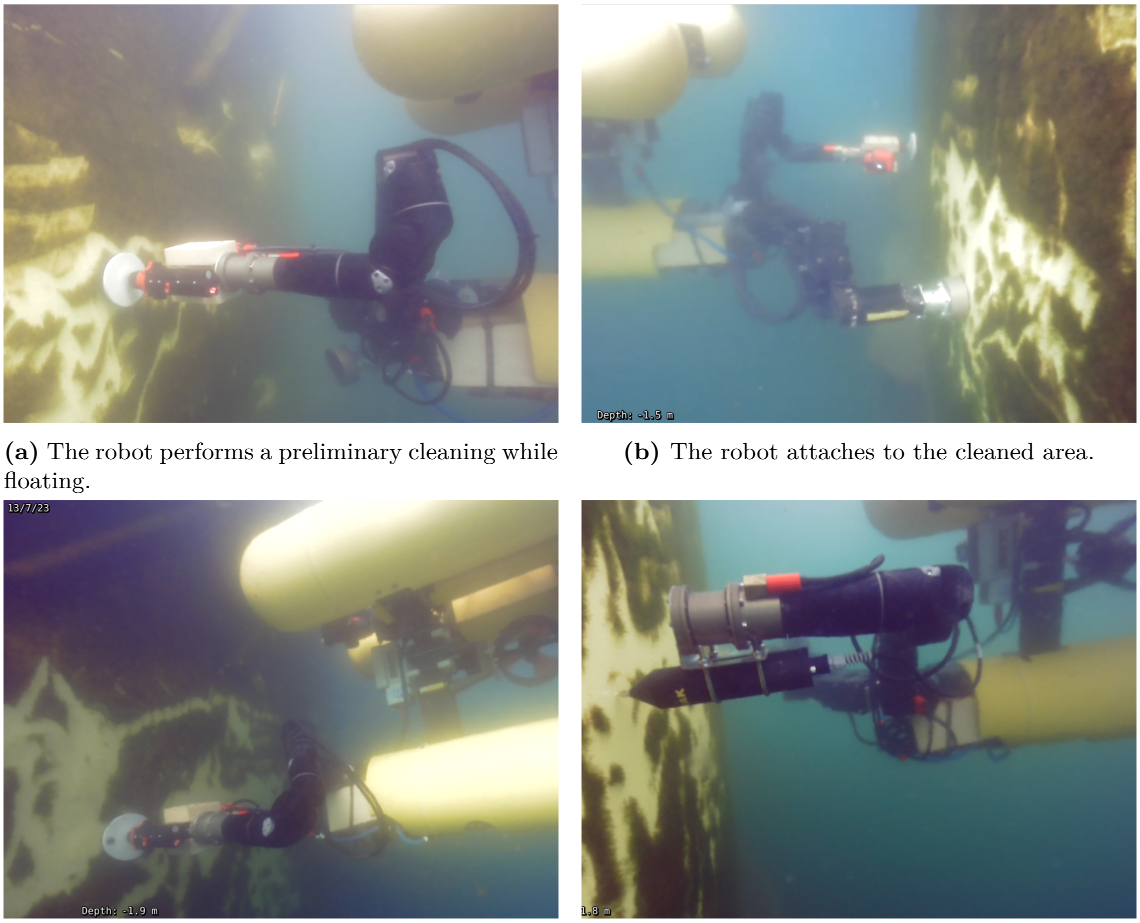
3. The cleaning operation is then carried out by the robot while being coupled to the structure through the magnetic EE.

4. The brush tool is exchanged with the CP probe. Currently, this procedural step involves manual intervention; therefore, the robot must be recovered.

5. Finally, the robot re-engages with the structure to perform CP testing on the biofouling-free area.

The impact of the brush is more apparent in Fig. 5.6. This image, captured by the robot's camera, highlights the contrast between conditions before and after cleaning.

The proposed force-control operation for CP testing presented in Chapter 4 was extended to execute a path-following task with the brush tool instead of only touch operations. For floating base cleaning, the trajectory is defined in the NED frame, and the 4 actuated DoF of the robot base are considered for the kinematic control. On the other hand, for cleaning with the ECA EE coupled with the structure, the trajectory is referenced to the point of contact



(a) The robot performs a preliminary cleaning while floating.

(b) The robot attaches to the cleaned area.

(c) The robot performs cleaning while coupled with the structure.

(d) After tool exchange, the robot performes CP testing on the biofouling-free area.

Figure 5.5: The sequence of intervention operations on the DURIUS structure with the Girona1000 I-AUV.

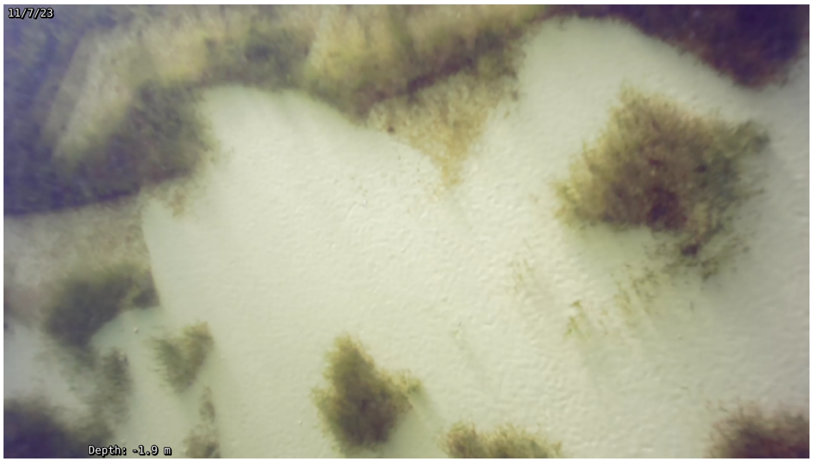


Figure 5.6: The effect of the brush on the Durius structure exposes its surface, enabling an appropriate inspection.

of the magnet with the structure, and the kinematic controller does not actuate the robot base. However, note that in both cases, the thrusters of the Girona1000 I-AUV must be used to compensate for the reaction forces during contact of the brush with the structure. For floating base force control, in the steady state, the contact force is always generated by the vehicle's thrusters. Even if the manipulator motion is the source of the initial force impulse, it is eventually necessary to counteract it with the vehicle's thrust. Similarly, the vehicle's thrusters are actuated when cleaning using magnetic coupling, to counteract the significant torque generated at the magnet EE due to the long lever arm between the brush and the magnet, which can potentially detach the robot in an uncontrolled way during operation, as explained in Chapter 4 (Section VI-B).

Note that the floating base cleaning suffers the uncontrolled motion of the structure as well as navigation drift, with the latter being significantly amplified in the ATLANTIS Test Centre. As the Durius structure is situated at the exit of a river (recall Fig. 5.2), the robot operates within a region where sea and fresh water coexist. These conditions introduce challenges that adversely affect the performance of acoustic sensors, notably the Doppler velocity log (DVL), as its signal traverses through different layers of water with varying densities. To assess the impact of these conditions, a survey mission was conducted at various depths, measuring sound velocity with the Girona1000 I-AUV's sound velocity sensor (SVS). The outcomes of this survey are illustrated in Fig. 5.7, depicting the robot's trajectory and depth alongside the SVS. Note how the sound velocity is clearly correlated with the depth of the vehicle.

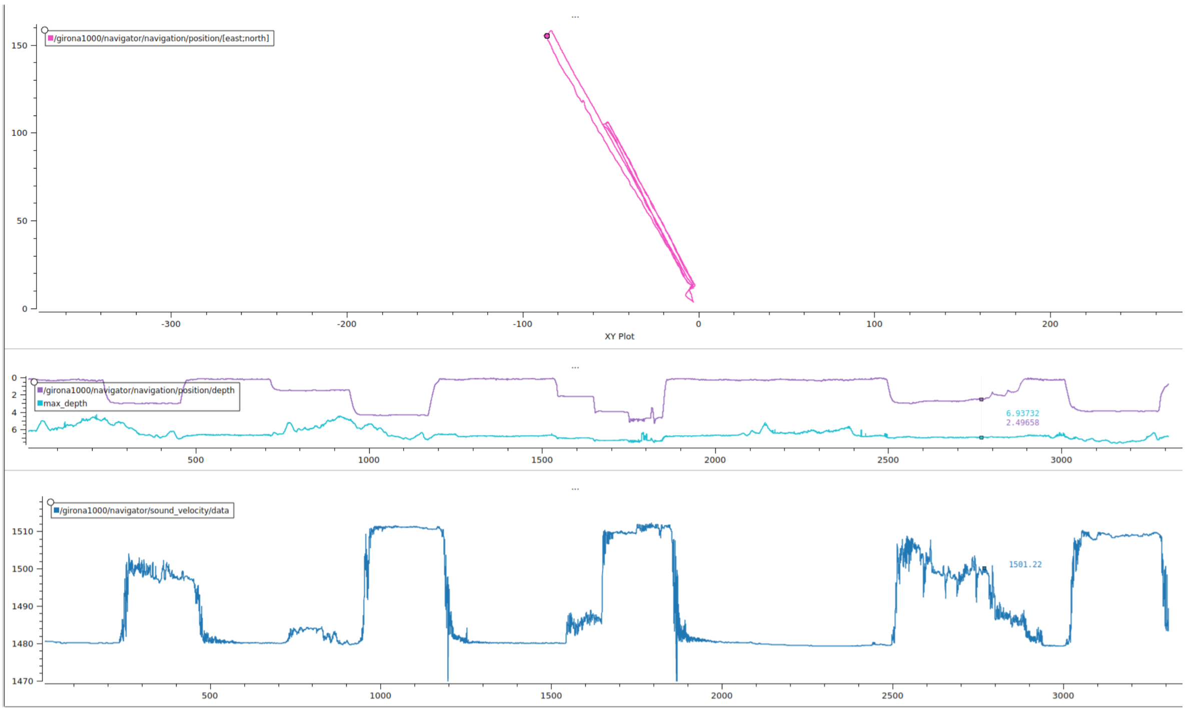


Figure 5.7: SVS survey close to the Durius structure at different depths. The top plot illustrates the xy position of the robot. The middle plot shows the depth measured by the robot (purple) and the sum of depth and altitude measurements (cyan). the bottom plot presents the corresponding SVS measurements, offering insights into the variations in sound velocity at different depths.

As a consequence of the noisy velocity measurements, not only did the navigation filter of the robot drift, but the controller was also affected during the floating cleaning operation. Figure 5.8 illustrates an attempt to perform a grid-like motion with the brush on the structure

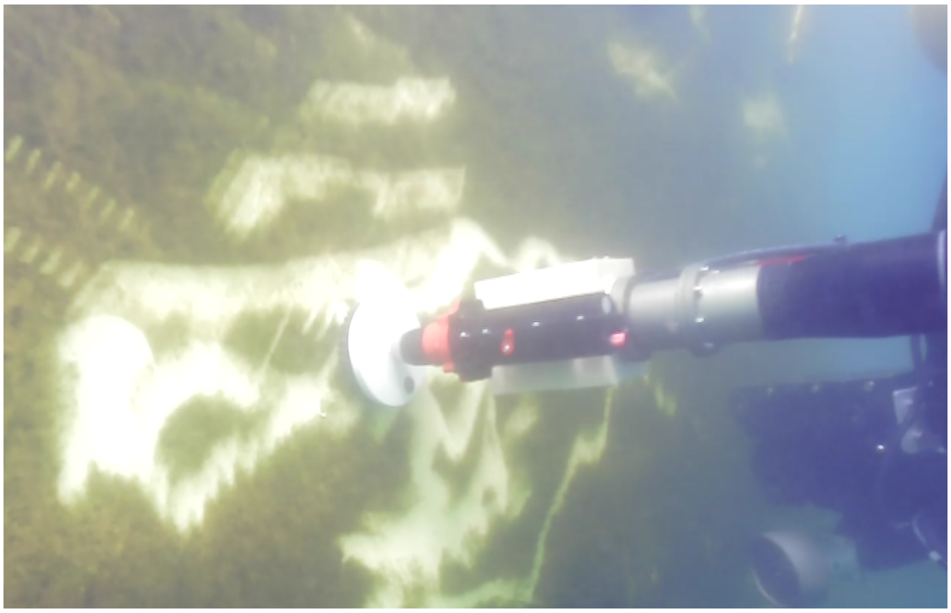


Figure 5.8: The noisy velocity measurements of the DVL due to the mixture of salty and fresh water affects the navigation and control of the floating cleaning operation.

while floating. Due to the impact of noisy velocity measurements, the brush's path is visibly drifting, necessitating the repetition of the operation several times to ensure adequate coverage of the area for proper attachment of the magnet. his situation highlights another benefit of using an attachment point with the structure for intervention. Beyond minimizing the effects of external disturbances and the motion of the structure, it establishes a reference point for subsequent intervention tasks, such as cleaning and CP testing in our case.

However, it is important to acknowledge that the magnet employed was not strong enough to attach to the structure consistently. In multiple trials, issues arose with the magnet detaching during the unfolding of the Bravo manipulator, due to its drag and inertia, or at the moment of contact of the brush or the CP tool with the structure. Consequently, extensive tuning of the control system responsible for the coupled motion of the vehicle and the manipulators was necessary to accomplish the reported tasks.

The custom-made magnetic EE is based on three neodymium magnets coated with epoxy for corrosion protection. Despite the considerable strength of the magnets, predicting their attachment force to the structure was challenging. First, the surface shell needs to be thick enough to hold the magnet's flux without becoming magnetically saturated. Second, the thickness of the coating applied on the surface significantly reduces the attachment's strength, as the magnetic force is inversely proportional to the distance squared. Although the thickness of the shell is not apparently an issue here, the thickness of the coating potentially is, due to the age of the structure and multiple maintenance cycles, greatly reducing the magnet's force on the structure. One attempt was made to replace the magnetic EE with another set of magnets available at hand (see Fig. 5.9), but this solution did not achieve any significant impact.

One idea to address these challenges in the future is to explore the use of magnetic handles typically used by divers and ROVs, as shown in Fig. 5.10. These magnetic handles are designed to provide a strong and secure attachment to metallic structures and have been proven effective in different underwater scenarios. Moreover, these commercial solutions often feature standardized designs and a variety of magnetic strengths that can be chosen based

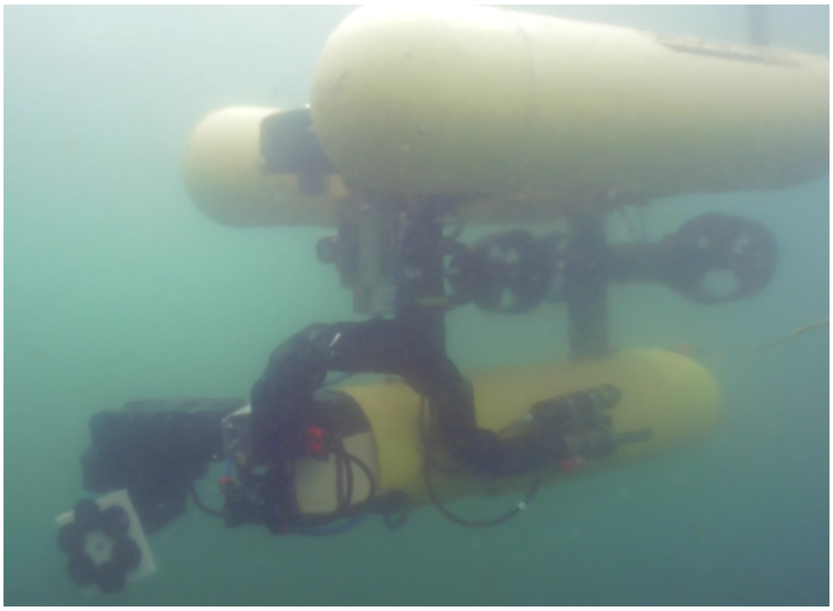


Figure 5.9: An alternative magnetic EE was explored to overcome the attachment issues.

on the scenario. By using these handles, the involved manipulator could be equipped with a standard gripper to hold and place the magnetic handle. Moreover, with this configuration, the robot could potentially use elements already present in the structure, such as welded handles or ladders, resulting in a more flexible solution.

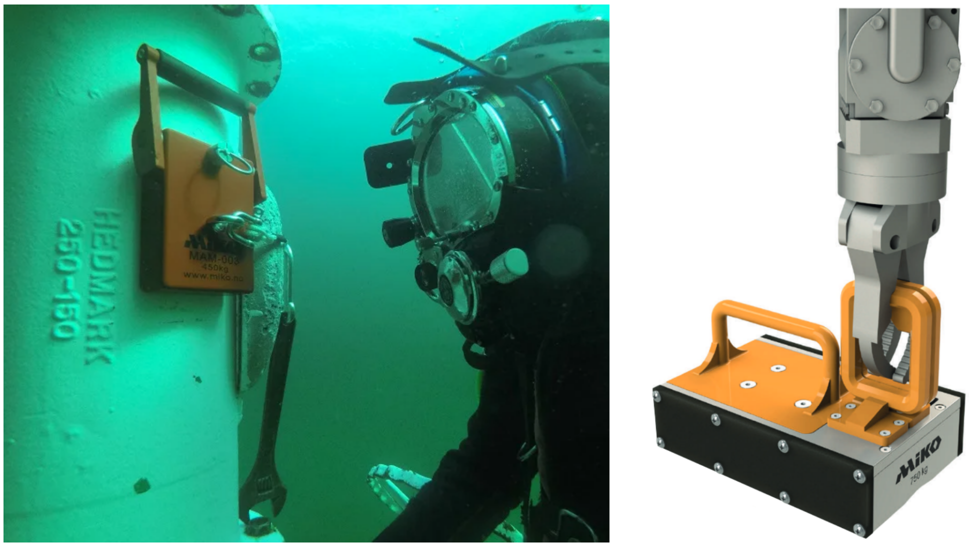


Figure 5.10: Two underwater magnets for divers and ROVs commercialized by Miko Marine[2]

Finally, the trials carried out in Viana do Castelo did not involve the use of markers for structure localization. The decision was influenced by three main factors. First, the rapid accumulation of marine growth rendered standard markers impractical for sea operations. Second, the turbidity of the water limited the visibility of the structure to less than 1 meter. Additionally, avoiding the use of markers, which require installation, is more convenient for real

[2] https://mikomarine.com

IMR tasks. To address these challenges, a profiler sonar was employed for structure detection, and subsequently, a circle-fitting algorithm was applied to the sonar data to determine the structure's location. The mixture of salty and fresh water also affected the performance of the sonar (see Fig. 5.11). The optimization process also estimated the radius of the cylindrical structure. Given our knowledge of the actual radius, this information served as a validation metric, ensuring the coherence and reliability of the detection results.

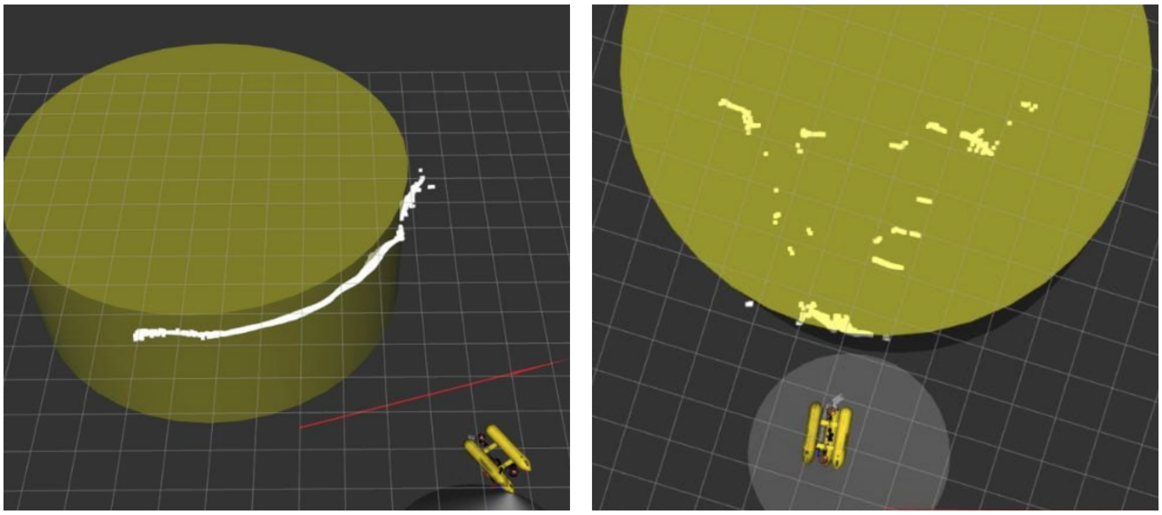


Figure 5.11: The profiler sonar utilized for detecting the Durius structure. The left image depicts a clear measurement resembling a circle, while the right picture illustrates a noisy measurement that was discarded.

6

Conclusions and Future Work

This chapter closes this thesis in two last sections. First, the main conclusions of this work are summarized in Section 6.1. Finally, new research lines for future work are proposed in Section 6.2.

6.1 Contributions of this thesis

This thesis has contributed to advancing the state of the art of I-AUVs, performing experimental demonstrations in different intervention scenarios. The objectives detailed in Chapter 1 have been successfully covered through the following contributions:

Develop a solution to accurately calibrate the I-AUV: The first contribution presented in this thesis was the formulation of a multi-manipulator multi-sensor calibration, which calibrates not only the extrinsic parameters of the sensors but also the intrinsic parameters of the manipulators, in this case, the joint offsets. In the article presented in Chapter 2, we explained the challenges and benefits of modeling the calibration problem mathematically rigorously, using Lie theory to perform a true on-manifold estimation. We put special attention to the development of Jacobians, which is essential for optimal estimators and a source of many difficulties when formulating new functions.

Cooperative autonomous pipe transportation: In this thesis, we also presented experimental trials of two I-AUVs cooperating to autonomously pick, transport, and place a large pipe. We propose a decentralized Task Priority control framework that requires very little data transfer, ideal for uncabled operations underwater.

Single-vehicle autonomous IMR of a floating wind turbine: Moreover, we have also investigated the use of an I-AUV to autonomously perform IMR operation on a semi-submergible floating structures. Specifically, we focused on cleaning operations, using a rotating brush, and performing CP surveys. The Girona1000 I-AUV was equipped with two manipulators, one dedicated to attaching to the target structure with a magnetic EE, and a dexterous manipulator equipped with a FT sensor and an interface to attach either a CP probe or a rotating brush. The high-level mission was modeled using Behavior Trees, with special emphasis on recovery behaviors in case the robot fails to execute some task. The low-level controller, based on Task Priority control, was extended with admittance control, as the aforementioned operations require to maintain contact with the structure. We tested these operations in the CIRS water tank and in more challenging and realistic conditions at sea in the ATLANTIS Test Center in Viana do Castelo.

Experimental validation of the work: The culmination of these efforts was the extensive experimental validation conducted in diverse scenarios, including controlled laboratory environments and maritime conditions, which distinguishes the contributions of this thesis.

6.2 Future work

In the pursuit of advancing autonomous underwater intervention capabilities, we identify various paths for future research and development. Consequently, we conclude this thesis by pointing out and discussing these potential research lines that can contribute to the continual enhancement of I-AUVs.

Advanced control strategies In this work, the coupled motion of the manipulators and vehicle is predominantly based on the Task Priority control framework for redundant robotic systems, featuring custom extensions to address the floating-base manipulation problem and enable force control. Future research could investigate more modern control solutions considering aspects such as dynamics, obstacle avoidance, or system uncertainties, enhancing the precision and reliability of the I-AUV controllers.

Motion Planning integration In addition, it is worth considering incorporating motion planning strategies for the manipulators and their combined motion with the vehicles during floating manipulation. Particularly, it may be interesting to explore methods for real-time motion planning, which can potentially be integrated with the control framework, enabling the ability to optimize planned trajectories online during their execution.

High-Level reasoning for task execution After the aforementioned topics, a natural step would be to introduce high-level reasoning methods to plan for the execution of tasks, taking into account complex objectives, task dependencies, and adaptive decision-making. In fact, with the advent of Large Language Models (LLMs), there is an unprecedented opportunity to leverage natural language processing and understanding capabilities. Integrating LLMs into the high-level reasoning of I-AUVs could empower them to interpret and respond to more intricate commands, handle dynamic scenarios, understand mission objectives, and adapt to unforeseen challenges through contextual analysis and decision-making. These advancements in high-level reasoning could mark a significant leap forward in the autonomy and adaptability of I-AUVs.

Enhanced perception capabilities Reducing dependence on artificial markers for intervention demonstrations is crucial for real-world applications. Future research efforts could focus on enhancing perception capabilities, aiming to detect and localize objects underwater without the need for external markers. Moreover, integrating advanced computer vision techniques with simultaneous localization and mapping (SLAM) could significantly improve the robustness and precision of I-AUVs, especially for operating close to structures where the performance of acoustic sensors such as the DVL decreases.

Tailored framework for autonomous intervention In order to keep advancing in the field, it is worth investing effort in building an underwater-tailored framework for intervention, integrating the aforementioned topics of control, planning, reasoning, perception, and navigation. Developing a modular framework for intervention would allow to systematically address specific challenges and incorporate these solutions within an evolving framework, preserving the research contributions. In many instances, valuable research efforts are conducted and demonstrated once but are not easily reproduced into subsequent projects, as the work is not structured in a scalable manner. A tailored, modular framework addresses this issue by providing a structured environment where each module is designed to be reusable and extensible.

Finally, it is important to promote testing the developed technologies at sea. While demonstrations in controlled conditions such as water tanks provide a solid foundation for the developed works, the ultimate validation of these innovations lies in their real-world performance. Sea testing offers unique challenges and insights that cannot be fully replicated in controlled environments, providing a rigorous testing ground to demonstrate the practical applicability of I-AUVs.

Bibliography

[1] **Roger Pi**, Patryk Cieślak, Pere Ridao, and Pedro J Sanz. “Twinbot: Autonomous underwater cooperative transportation”. In: *IEEE Access* 9 (2021), pages 37668–37684. DOI: 10.1109/ACCESS.2021.3063669 (cited on pages xi, 37).

[2] **Roger Pi**, Patryk Cieślak, Joan Esteba, Narcís Palomeras, and Pere Ridao. “Compliant Manipulation With Quasi-Rigid Docking for Underwater Structure Inspection”. In: *IEEE Access* 11 (2023), pages 128957–128969. DOI: 10.1109/ACCESS.2023.3332486 (cited on pages xi, 55).

[3] **Roger Pi**, Pau Vial, Narcís Palomeras, and Pere Ridao. “Robust calibration of multi-sensor multi-manipulator mobile robots using nonlinear on-manifold optimization”. Submitted to IEEE Transactions on Robotics (2024) (cited on pages xi, 15).

[4] Miguel Castillón, **Roger Pi**, Narcís Palomeras, and Pere Ridao. “Extrinsic visual–inertial calibration for motion distortion correction of underwater 3D scans”. In: *IEEE Access* 9 (2021), pages 93384–93398. DOI: 10.1109/ACCESS.2021.3092180 (cited on page xi).

[5] **Roger Pi**, Dina Youakim, Patrick Cieslak, and Pere Ridao. “Multi-representation multi-heuristic A* motion planning for a dual-arm underwater vehicle manipulation system”. In: *IFAC-PapersOnLine* 52.21 (2019), pages 205–210. DOI: 10.1016/j.ifacol.2019.12.308 (cited on page xi).

[6] **Roger Pi**, Joan Esteba, Patryk Cieslak, Narcís Palomeras, Pedro J Sanz, Raúl Marín, and Pere Ridao. “OPTIHROV: Optically Linked Hybrid Autonomous/Remotely Operated Vehicle, Beyond Teleoperation in a New Generation of Underwater Intervention Vehicles”. In: *OCEANS 2023-Limerick*. IEEE. 2023, pages 1–7. DOI: 10.1109/OCEANSLimerick52467.2023.10244690 (cited on page xii).

[7] H H Wang, S M Rock, and M J Lees. “Experiments in automatic retrieval of underwater objects with an AUV”. In: *OCEANS ’95. MTS/IEEE. Challenges of Our Changing Global Environment. Conference Proceedings* 1 (1995), pages 366–373 (cited on page 8).

[8] S K Choi, G Y Takashige, and J Yuh. “Experimental study on an underwater robotic vehicle: ODIN”. In: *Proceedings of IEEE Symposium on Autonomous Underwater Vehicle Technology*. Dec. 1994 (cited on page 8).

[9] V Rigaud, E Coste-Maniere, and M J Aldon. “UNION: underwater intelligent operation and navigation”. In: *IEEE Robotics & Automation Magazine* 5.1 (Mar. 1998), pages 25–35 (cited on page 8).

[10] D M Lane, JBC Davies, and G Casalino. "AMADEUS: Advanced MAnipulation for DEep Underwater Sampling". In: *IEEE Robotics & Automation Magazine* 4.4 (Dec. 1997), pages 34–45 (cited on page 8).

[11] Giacomo Marani, Song K. Choi, and Junku Yuh. "Underwater autonomous manipulation for intervention missions AUVs". In: *Ocean Engineering* 36.1 (2009). Autonomous Underwater Vehicles, pages 15–23. DOI: https://doi.org/10.1016/j.oceaneng.2008.08.007 (cited on page 8).

[12] G Marani and J Yuh. *Introduction to Autonomous Manipulation, Case Study with an Underwater Robot, SAUVIM*. Volume 102. Berlin: Springer, Apr. 2014 (cited on page 8).

[13] Mario Prats, David Ribas, Narcis Palomeras, JuanCarlos Garcia, Volker Nannen, Stephan Wirth, Jose Javier Fernandez, JoanP Beltran, Ricard Campos, Pere Ridao, PedroJ Sanz, Gabriel Oliver, Marc Carreras, Nuno Gracias, Raul Marin, and Alberto Ortiz. "Reconfigurable AUV for intervention missions: a case study on underwater object recovery". In: *Intelligent Service Robotics* 5.1 (2012), pages 19–31 (cited on page 8).

[14] P. J. Sanz, P. Ridao, G. Oliver, G. Casalino, Y. Petillot, C. Silvestre, C. Melchiorri, and A. Turetta. "TRIDENT An European project targeted to increase the autonomy levels for underwater intervention missions". In: *2013 OCEANS - San Diego*. 2013, pages 1–10. DOI: 10.23919/OCEANS.2013.6741370 (cited on page 8).

[15] J C Evans, P Redmond, C Plakas, K Hamilton, and D Lane. "Autonomous docking for Intervention-AUVs using sonar and video-based real-time 3D pose estimation". In: *Proceedings of OCEANS 2003*. 2003 (cited on page 8).

[16] N Palomeras, P Ridao, D Ribas, and G Vallicrosa. "Autonomous I-AUV Docking for Fixed-base Manipulation". In: *IFAC Proceedings Volumes* 47.3 (Jan. 2014), pages 12160–12165 (cited on page 8).

[17] A Carrera, N Palomeras, N Hurtos, P Kormushev, and M Carreras. "Learning by demonstration applied to underwater intervention". In: *Proceedings of the Seventeenth International Conference of the Catalan Association of Artificial Intelligence (CCIA)*. 2014 (cited on pages 8, 11).

[18] P Cieslak, P Ridao, and M Giergiel. "Autonomous underwater panel operation by GIRONA500 UVMS: A practical approach to autonomous underwater manipulation". In: *Proceedings of IEEE International Conference on Robotics and Automation*. 2015 (cited on pages 8, 11).

[19] Patryk Cieślak, Roberto Simoni, Pere Ridao Rodríguez, and Dina Youakim. "Practical formulation of obstacle avoidance in the Task-Priority framework for use in robotic inspection and intervention scenarios". In: *Robotics and Autonomous Systems* 124 (2020), page 103396. DOI: https://doi.org/10.1016/j.robot.2019.103396 (cited on pages 8, 11).

[20] Dina Youakim, Patryk Cieslak, Andrew Dornbush, Albert Palomer, Pere Ridao, and Maxim Likhachev. "Multirepresentation, Multiheuristic A* search-based motion planning for a free-floating underwater vehicle-manipulator system in unknown environment". In: *Journal of Field Robotics* 37.6 (), pages 925–950. DOI: https://doi.org/10.1002/rob.21923 (cited on pages 9, 11).

[21] Paolo Di Lillo, Enrico Simetti, Francesco Wanderlingh, Giuseppe Casalino, and Gianluca Antonelli. “Underwater Intervention With Remote Supervision via Satellite Communication: Developed Control Architecture and Experimental Results Within the Dexrov Project”. In: *IEEE Transactions on Control Systems Technology* 29.1 (2021), pages 108–123. DOI: 10.1109/TCST.2020.2971440 (cited on page 9).

[22] Patryk Cieslak and Pere Ridao. “Adaptive Admittance Control in Task-Priority Framework for Contact Force Control in Autonomous Underwater Floating Manipulation”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2018 (cited on pages 9, 11).

[23] Marc Carreras, Juan David Hernandez, Eduard Vidal, Narcis Palomeras, David Ribas, and Pere Ridao. “Sparus II AUV - A Hovering Vehicle for Seabed Inspection”. In: *IEEE Journal of Oceanic Engineering* 43.2 (2018), pages 344–355. DOI: 10.1109/JOE.2018.2792278 (cited on page 11).

[24] David Ribas, Narcís Palomeras, Pere Ridao, Marc Carreras, and Angelos Mallios. “Girona 500 AUV: From survey to intervention”. In: *IEEE/ASME Transactions on Mechatronics* 17.1 (2012), pages 46–53. DOI: 10.1109/TMECH.2011.2174065 (cited on page 11).

[25] A Carrera, SR Ahmadzadeh, A Ajoudani, P Kormushev, M Carreras, and DG Caldwell. “Towards autonomous robotic valve turning”. In: *Cybernetics and Information Technologies* 12.3 (2012), pages 17–26 (cited on page 11).

[26] Arnau Carrera, Narcís Palomeras, Natàlia Hurtós, Petar Kormushev, and Marc Carreras. “Cognitive system for autonomous underwater intervention”. In: *Pattern Recognition Letters* 67 (2015), pages 91–99 (cited on page 11).

[27] Dina Youakim, Pere Ridao, Narcís Palomeras, Francesco Spadafora, David Ribas, and Maurizio Muzzupappa. “MoveIt!: Autonomous Underwater Free-Floating Manipulation”. In: *IEEE Robotics and Automation Magazine* 24.3 (2017), pages 41–51. DOI: 10.1109/MRA.2016.2636369 (cited on page 11).

[28] Andry Maykol Pinto, Joao V.Amorim Marques, Daniel Filipe Campos, Nuno Abreu, Anibal Matos, Martio Jussi, Robin Berglund, Jari Halme, Petri Tikka, Joao Formiga, Christian Verrecchia, Serena Langiano, Clara Santos, Nuno Sa, Jaap Jan Stoker, Fabrice Calderoni, Shashank Govindaraj, Alexandru But, Leslie Gale, David Ribas, Natalia Hurtos, Eduard Vidal, Pere Ridao, Patryk Chieslak, Narcis Palomeras, Stefano Barberis, and Luca Aceto. “ATLANTIS - The Atlantic Testing Platform for Maritime Robotics”. In: *Oceans Conference Record (IEEE)* 2021-Septe (2021), pages 1–5. DOI: 10.23919/OCEANS44145.2021.9706059 (cited on page 70).