Programa de Doctorado en Informática
Escuela de Doctorado de la Universitat Jaume I

PhD Thesis

Framework for Autonomous Underwater Robotic Grasping of Unknown Objects

Author: Antonio Peñalver Monfort

Supervisors: Pedro J. Sanz Valero

Raúl Marín Prades

Castellón de la Plana, October 2018

Sempre amunt.

Acknowledgements

It would not have been possible to complete this doctoral thesis without the help and support of many people around me. Among these lines, I would like to express my gratitude. May the reader forgive me the discourtesy of following my wish to write the acknowledgements in my mother tongue.

En primer lugar, me gustaría dar las gracias a mis directores Raúl Marín y Pedro Sanz por introducirme en el mundo de la investigación y, en concreto, en el desafiante campo de la robótica submarina. También quisiera agradecerles los consejos y apoyo durante estos años.

Gracias a la Generalitat Valenciana por financiar el desarrollo de esta tesis a través del programa VALI+D para investigadores en formación (referencia de la ayuda ACIF/2014/298).

También he de dar las gracias a todos los miembros del IRSLab, quienes aparte de unos increíbles compañeros de trabajo, han sido mi segunda familia durante todos estos años. Os podría escribir muchas más cosas pero como ya sabéis, este no es mi fuerte.

Gracias a mis padres por cuidarme y educarme durante toda mi vida. Pero, sobretodo, por apoyarme incondicionalmente y darme la libertad para decidir mi camino.

Por último, gracias a Lidon, por escucharme, por aguantarme, por hacerme feliz y por recordarme cada día que las cosas importantes de verdad se encuentran fuera del trabajo (y del fútbol).

Resumen

Los océanos han tenido una enorme influencia en la vida de los humanos. Incluso hoy en día, existen un gran número de aplicaciones que necesitan ser desarrolladas en este entorno. No obstante, su exploración, especialmente en las grandes profundidades, está aún lejos de completarse.

Las primeras exploraciones de los océanos se llevaron a cabo utilizando vehículos ocupados por humanos. Pero, con el fin de evitar riesgos humanos, se desarrolló el primer vehículo remotamente operado (ROV). El uso extensivo de los ROVs está muy limitado debido a los altos costes operacionales, la fatiga que sufren los operadores, así como por temas de seguridad. Durante los últimos años, se han hecho grandes esfuerzos en dotar a estos vehículos de un mayor grado de autonomía. Con ello, se aumentan las capacidades de los vehículos, así como se reduce su coste. Como resultado, se han diseñado los vehículos autónomos subacuáticos (AUVs). Unas pocas plataformas se encuentran disponibles hoy en día en el mercado, y, en su mayor medida, son usadas para realizar misiones de mapeado.

Existen un gran número de aplicaciones que van más allá del mapeado. Estas aplicaciones tienen en común la necesidad de interaccionar con el entorno. Hoy en día, estas tareas requieren el uso de ROVs. Pero, recientemente, algunos investigadores han comenzado a trabajar sobre la evolución natural de los AUV, los vehículos autónomos subacuáticos de intervención (I-AUV). Este nuevo concepto consiste en dotar a un AUV de un manipulador para automatizar las tareas de intervención.

En esta tesis, se ha estudiado el problema de agarrar un objeto desconocido en un entorno subacuático de forma autónoma utilizando un I-AUV. Este problema engloba la reconstrucción 3D de la escena, la planificación y la ejecución del agarre.

Antes de comenzar un agarre, se necesita obtener tanta información como sea posible del objeto de interés y sobre lo que lo está rodeando. Consecuentemente, se ha presentado una nueva aproximación para obtener una reconstrucción 3D precisa y completa de forma autónoma, sobre un objeto situado en un entorno subacuático. La aproximación consiste en colocar un emisor láser y una cámara en el antebrazo del brazo robótico. Mientras el

brazo se mueve, el láser escanea la escena y la cámara la graba. Entonces, utilizando un detector de puntos láser junto con un algoritmo de reconstrucción 3D, se calcula la una nube de puntos 3D de la escena, a partir de las imágenes grabadas.

Las redundancias en las cinemáticas de los I-AUVs se ha de controlar para permitir que el manipulador alcance una posición deseada sin exceder ningún límite y, si es posible, cumpliendo un conjunto de tareas con distintas prioridades. En la tesis, se ha introducido un algoritmo para controlar cinemáticas redundantes. Este incluye, en un único programa, el tratamiento de múltiples tareas, múltiples cadenas cinemáticas con diferentes prioridades en las articulaciones, así como restricciones estrictas.

Una vez que el I-AUV ha reconstruido la escena, se ha de planificar la manipulación. La alternativa presentada toma como entrada la nube de puntos de la escena, a partir de ella, se detectan los objetos situados en la escena y se estiman sus tamaños. De entre ellos, se selecciona el objeto que mejor cumpla una serie de condiciones. Entonces, un algoritmo de planificación de agarre, calcula la posición y orientación que cumpla, en mayor medida, un conjunto de tareas ordenadas por prioridad y que estrictamente cumpla una serie de restricciones. Para conseguirlo, se simulan los movimientos que el sistema realizaría mientras intenta alcanzar una posición válida.

Después de calcular la posición y orientación de agarre, esta debe ser alcanzar adecuadamente con el efector final del manipulador. Pero, no es una buena idea mover directamente el efector hacia esa posición, ya que los dedos de la garra podrían colisionar con el objeto mientras se alcanza la posición. La metodología propuesta, mueve, en primer lugar, el efector final a una posición que facilita los movimientos de aproximación posteriores. Entonces, este es guiado hacia la posición de agarre. Una vez allí, la pinza se cierra agarrando el objeto.

Finalmente, se han explicado dos metodologías para detectar problemas durante la intervención y corregirlos. El primero, controla la posición y orientación alcanzados por el efector final del brazo. Se basa en la detección de un marcador colocado en la pinza del manipulador, utilizando una cámara situada en la base del brazo. Cada vez que la cámara detecta el marcador, se calcula la posición del efector final y se corrigen los valores de las articulaciones del brazo.

La segunda metodología, detecta contactos entre la garra y el entorno, y corrige la trayectoria del efector final para evitarlos. Para ello, se ha instalado un sensor de fuerza-torque en la muñeca del manipulador. Durante la fase de aproximación, si se detecta una colisión, se estudia la posición y orientación actual del efector final, para detectar si la garra está en contacto con el objeto de interés, o con el suelo. Dependiendo del resultado, la trayectoria se modifica evitando este contacto y continuando con el agarre.

Abstract

During the course of the history, oceans have had a huge influence on humans' live and there exist a large number of applications that needs to be done in this environment. Nonetheless, the exploration of the full depths of the oceans is still far from being complete.

First explorations of the oceans were conducted through human occupied vehicles. But, with the main goal of avoiding human risks, the first remotely operated vehicle (ROV) was developed. Extensive use of ROVs is currently very limited because of very high operational costs, operator fatigue and safety issues. In recent years, efforts have been made to provide those vehicles with a greater degree of autonomy, expanding the vehicle capabilities as well as reducing costs. As a result, the so called Autonomous Underwater Vehicles (AUVs) were designed. Few autonomous platforms are nowadays in the market, but they are routinely used in survey mission.

A large number of applications exist which go beyond the survey capabilities. These applications have in common the necessity of interacting with the environment. Nowadays these tasks require the use of ROVs. But, very recently, some researchers have started to think about the natural evolution of the intervention AUV, the Intervention Autonomous Underwater Vehicle (I-AUV). This new concept consists in endowing an AUV with a manipulator to automate some intervention tasks.

In this thesis, the problem of autonomously grasping an unknown object in an underwater scenario has been studied. This problem encompassess the 3D reconstruction of the scene, the grasp planning and the grasp execution.

Before start a grasping, it is needed to obtain as much information as possible about the object of interest and what is surrounding it. Consequently a new approach to autonomously obtain an accurate and complete 3D reconstruction of an object in an underwater scenario has been presented. The approach consists in attaching a laser strip emitter and a camera at the forearm of a robotic arm. While the arm is moving, the laser scans and the camera records the scene. A laser peak detector and a 3D reconstruction algorithms are used to obtain a 3D point cloud from the images captured.

I-AUVs' kinematic redundancy must be properly controlled to allow the manipulator to reach a desired pose without exceeding any limit and, if it is possible, fulfilling a set of tasks with different priorities. A framework for controlling kinematic redundancy has been introduced. It integrates, in a unified program, the treatment of multiple tasks, multiple kinematic chains, different joint priorities and hard constraints.

Once the I-AUV has reconstructed the scene, the manipulation requires to be planned. The presented alternative takes as input the point cloud of the scene, detects the objects placed there and estimate their sizes. The object that better fits with a series of conditions is selected. Then, a grasp planning algorithm calculates the grasping pose that fulfil in a greater way a hierarchy of tasks and that strictly accomplishes a set of constraints. To do so, it simulates the movements that the system would perform while it is trying to reach a valid pose.

After calculating the grasping pose, this must be properly reached by the end-effector of the manipulator. But, it is not a good idea to move it directly to this pose, since the fingers of the gripper could collide with the object when they are reaching the final pose. The proposed methodology moves the end-effector to a pose that facilitate the posterior approximation movements. Then, it is guided to the grasping pose. Once there, the gripper is closed grasping the object.

Finally, two methodologies for detecting problems during an intervention and correct them have been explained. The first one, controls the position and orientation reached by the arm end-effector. This is based on the detection of a marker placed on the gripper using a camera placed on the base of the arm. Each time the camera detects the marker, the end-effector pose is calculated and the arm joint values corrected.

The second methodology, detects contacts between the gripper and the environment and correct the trajectory of the end-effector for avoiding them. This is based on the installation of a force-torque sensor at the wrist of the manipulator. During the approaching phase, if a collision is noticed, the current position and orientation of the end-effector is studied to detect if the gripper is in contact with the object or the floor. Depending on the result, the trajectory is changed to avoid this collision and continue with the grasping.

Contents

Chapter 1

Introduction

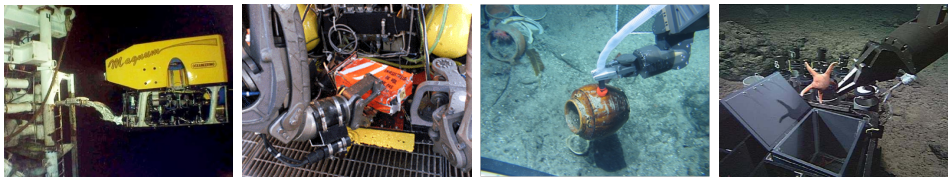
The oceans cover about 70% of the earth's surface and contain 95% of the living space on earth. With as many as 100 million species, marine biodiversity far outweighs that on land and new species are being discovered all the time.

Oceans also have a huge influence on humans' life. They produce 70% of the oxygen, absorb heat and re-distribute it around the world. They have represented a critical sources of food and an indispensable way of transport between nations.

Nonetheless, the exploration of the full depths of the oceans, with its abundant living and non-living resources, is still far from being complete. The first explorations of the oceans were conducted through human occupied vehicles, like the bathyspheres build by Charles William Beebe (1877-1952) and Auguste Piccard (1884-1962) or the well-known Alvin deep-sea submersible build by Allyn Vine (1941-1994). But the production of these vehicles reached its peak in the late 1960s, when several of the defense constructors such as General Dynamics, Rockwell and Westinghouse were building these systems.

Thanks to the advance of the technology and in order to avoid human risks, the idea of an unmanned underwater vehicle appeared soon and the first remotely operated vehicle (ROV) was developed. The first ROV known as POODLE, was build by Dimitri Rebikoff in 1953. The initial technical problems in ROV design were worked out in the early 1980s. Then, the ROVs were able to definitely substitute unmanned vehicles in a huge number of applications. However, extensive use of manned submersibles and ROVs are currently very limited because of very high operational costs, operator fatigue and safety issues [Yuh and West, 2001].

In recent years, efforts have been made to provide those vehicles with a greater degree of autonomy. The main goal is to remove the tether which connects the vehicle with the surface ship, expanding the vehicle capabilities as



(a) *(b)* *(c)* *(d)*

Figure 1.1: Example of ROV's interventions where the manipulator is used: (a) Offshore oil and gas industry operations; (b) Black-box recovery from a crashed airplane; (c) Intervention operations in archaeology; (d) Sampling for Biology.

well as reducing costs. This is achieved by attaching to the vehicles their own energy supplies and giving them a certain level of intelligence, that allows them to determine their actions based on the inputs from their own sensors and a pre-defined mission plan. As a result, the so called Autonomous Underwater Vehicles (AUVs) were designed. After years of research, few autonomous platforms are already available in the market, most of them able to perform side scan sonar and bathymetric multi-beam surveys. 3D optical maps are nowadays one of the major fronts of research.

However, a large number of applications exist which go beyond the survey capabilities. These applications have in common the necessity of interacting with the environment. The most common can be summed up in the following list:

- Oil and gas industry: inspection and repairing of submerged infrastructures.
- Search and recovery: localization and grasping objects on the seafloor.
- Deep water archaeology: recovery of benthic stations.
- Science: periodic maintenance of underwater permanent observatories, ocean survey and sampling of marine chemistry, geology and biology.

Nowadays, these tasks require the use of ROVs (see Fig. 1.1), which means an expensive oceanographic vessel connected to the vehicles and a ROV's pilot suffering cognitive fatigue whilst he is doing a really complicated job. For all these reasons, very recently some researchers have started to think about the natural evolution of the intervention ROV, the Intervention Autonomous Underwater Vehicle (I-AUV). This new concept consists in endowing an AUV with a manipulator to automate some intervention tasks (e.g. valve turning, object recovery, etc.).

The main differences between the behaviours of an I-AUV and an AUV start when the vehicle is close enough to the object which is going to be manipulated. It is in that moment when the manipulator goes into action. Due to the novelty of the I-AUV concept, this phase of the mission has not still been studied enough so it is still far from being a solved problem.

In this thesis, the problem of autonomously grasping an unknown object in an underwater scenario has been studied. This problem encompasses the 3D reconstruction of the scene, the grasp planning and the grasp execution.

1.1 Previous Research Projects

Concerning autonomous manipulation in underwater environments, the pioneering works appeared in the 90s (OTTER [Wang et al., 1995], ODIN [Choi et al., 1994]), but significant advances in this direction arrived during the last decade, especially when the first simple autonomous operations at sea were demonstrated. Most of the advances were obtained in coordinated research projects like the ones listed hereinafter:

- **UNION** 1996-99 [Rigaud et al., 1998]: The project focused mainly on the development of coordinated control and sensing strategies for combined manipulator and vehicle systems. UNION represents the first mechatronic assembly of a complete vehicle-manipulator system for automated manipulation.

- **AMADEUS** 1993-99 [Lane et al., 1997]: Amadeus had two phases: the first phase represents the first attempt to develop a dexterous gripper suitable for underwater applications. The 3-fingered gripper was hydraulically actuated and coordinately controlled by mimicking. In the second phase,the coordinated control of two underwater electromechanical arms was studied. The project demonstrated the coordinated motion of the two fixed based manipulators while manipulating a rigid object inside a water tank.

- **SWIMMER** 1999-01 [Evans et al., 2001]: It was a hybrid AUV/ROV intervention system, where an AUV shuttle transports an intervention ROV to the subsea. SWIMMER was able to autonomously transit to the seafloor and dock to a subsea cradle based docking station. Once the vehicle was docked, the transported ROV was deployed and the intervention was carried out in a conventional teleoperated way.

- **ALIVE** 2001-04 [Evans et al., 2003]: The ALIVE vehicle was equipped with two hydraulic grippers for docking in a subsea intervention panel

using an imaging sonar and a manipulator arm. It has been reported as the first AUV able to autonomously carry out a manipulation action consisting in opening/closing a valve in a subsea panel.

- **SAUVIM** 1997-09 [Marani et al., 2009]: SAUVIM focused on the free floating manipulation concept and demonstrated accurate navigation and station keeping being the first project to demonstrate autonomous recovery of an a priori known object. The object was endowed with artificial landmarks and the robot autonomously located it and hooked it with a recovery device while hovering.

- **TRIDENT** 2010-13 [Sanz et al., 2013]: TRIDENT project proposes a new methodology to provide multipurpose dexterous manipulation capabilities for intervention operations in unknown, unstructured and underwater environments. In the TRIDENT project, a multipurpose generic intervention is composed of two phases. In the first phase, the I-AUV is deployed for surveying a given region of interest on the seabed and build an image photo-mosaic. The target of interest is then identified on the mosaic and the manipulation action is specified by means of a suitable user interface. After that, during the second phase, the I-AUV navigates close to the identified target, localizes it and executes the intervention mission, doing all of that in an autonomous manner.

- **PANDORA** 2012-15 [Lane et al., 2012]: The aim of the PANDORA project is to extend the range of tasks that can be carried out autonomously and increase their complexity while reducing the need for operator assistances. Dynamic adaptation to the change of conditions is very important while addressing autonomy in the real world and not just in well-known situations. The key of PANDORA is the ability to recognize failures and respond to them, at all levels of abstraction.

In summary, to the best of authors' knowledge, grasping and manipulation remain open research problems, and this situation becomes drastically worst in underwater scenarios, due to the difficulties arose under the very hostile underwater conditions. Only a few commercial robots, mainly specialized for very specific and limited operations, and mostly used in the offshore industry, have been endowed with grasping and manipulation capabilities. Related with research projects, only some of them have demonstrated reasonable performance in sea trials.

1.2 Context

The research of this thesis has been conducted in the Interactive and Robotic Systems Lab (IRSLab) at the Jaume I University of Castellón. During last decade, the group has been mainly working in the autonomous underwater robotic manipulation topic, publishing papers such as [Prats et al., 2012e], [Sanz et al., 2013].

Moreover, the research group also studies other fields of work very related with the underwater manipulation. The group is developer and maintainer of one of the most used underwater simulator, the UWSim [Prats et al., 2012c], which has been an essential tool for testing all the software before it is used in the real systems. In order to facilitate the control of the I-AUVs, some works related to human robot interfaces have been developed [Garcia et al., 2010] [Garcia et al., 2015]. In the field of vision, the group is working in dehazing underwater images using deep learning [Pérez et al., 2017]. In [Fornas et al., 2016], the best grasping pose is extracted from a point cloud obtained through a stereo camera. Free floating control of an AUV is studied in [Fernández et al., 2015]. And a new field of research is the wireless underwater communication, some preliminary results are shown in [Centelles et al., 2015] [Rubino et al., 2017].

The work presented in this thesis falls completely in the main field of the group, the autonomous underwater robotic manipulation. But, in order to improve in this general topic, other more specific topics have also been studied. Some of them are 3D reconstruction, vision, grasp planning, robot kinematics, etc.

Prior to the beginning of the thesis and during the course of it, the IRSLab have been involved in several projects which main goal was to perform an underwater intervention using an I-AUV. Some of the improvements achieved during the thesis have helped to the proper development the projects. A brief description of them can be found below and an image of each experimental validation in the Fig. 1.2:

- **RAUVI**: Reconfigurable AUV for intervention missions (DPI2008-06548-C03) funded by the Spanish Ministry. The main goal of the project is to develop and improve the necessary technologies for autonomously performing an intervention mission in underwater environments. Detection and hooking of a black-box mock-up using and I-AUVI was successfully proved [1].

- FP7 EU **TRIDENT**: Marine robots and dexterous manipulation for enabling autonomous underwater multipurpose intervention missions(FP7-

[1] Video of the RAUVI final experiment https://www.youtube.com/watch?v=ha9_vMAnWyQ

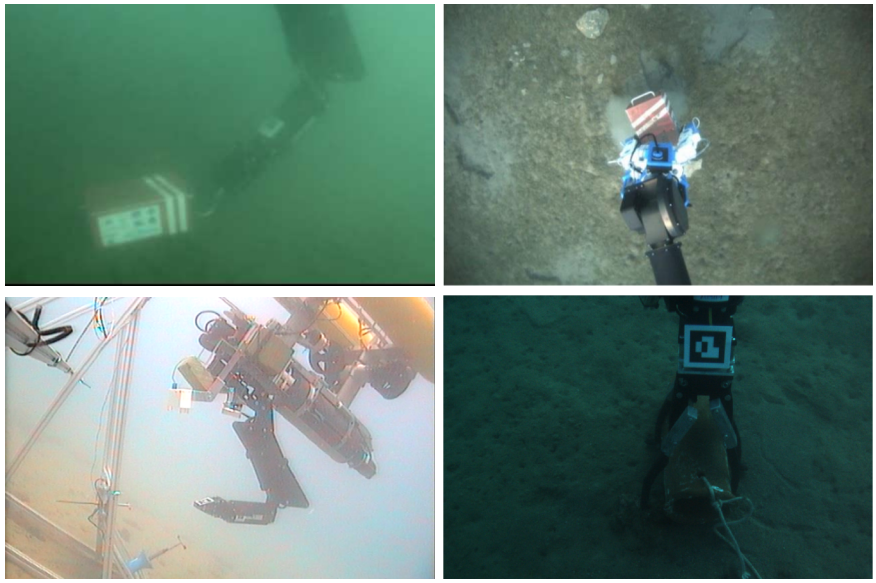


Figure 1.2: Sea trials on the RAUVI (top left), TRIDENT (top right), TRITON (bottom left) and MERBOTS (bottom right) projects.

ICT-2009-248497) funded by the European commission. This project proposes a new methodology to provide multipurpose dexterous manipulation capabilities for intervention operations in unknown, unstructured and underwater environments. Dexterous autonomous grasping of a black-box mock-up using free-floating manipulation was demonstrated [2].

- **TRITON**: Multisensory based underwater intervention through cooperative marine robots (DPI2011-27977-C03) funded by the Spanish Ministry. The project is focused on the development of technologies really close to the real needs of the final user, facilitating the potential technological transfer of its results. The test bed for this project was an experimental validation in an intervention panel where the vehicle autonomously docked and manipulated a valve and hot stab in pool and sea conditions [3].

- **MERBOTS**: Multifunctional cooperative marine robots for intervention domains (DPI2014-57746-C3) funded by the Spanish Ministry.

[2] Video of the TRIDENT final experiment https://www.youtube.com/watch?v=2qf7ukrUcCc

[3] Video of the TRITON final experiment https://www.youtube.com/watch?v=xA2SGLi5TYg

> This project aims to use multirobot cooperation and multimodal systems to perform archaelogical interventions in a safer and cheaper way. The case of study for this project is the autonomous cooperative localization, unearthing and grasping of an amphora located in the sea floor [4].

Additionally, the work showed in the thesis was benefited by two stays of three months. The first one was carried out in the Ocean System Laboratory (OSL) at the Heriot Watt University (Endiburgh, UK) under the supervision of the Professor Yvan Petillot. The second one was accomplished in the German Research Center for Artificial Intelligence (DFKI) at the University of Bremen (Bremen, Germany) under the supervision of the Doctor Peter Kampmann.

1.3 Aims and Scope

As mentioned in the introduction, the goal of this thesis is to study the problems related with the autonomous grasping of unknown objects in underwater scenarios. These problems have been separated in three groups plus an additional group in charge of improving the reliability of the intervention (see Fig. 1.3). For each group, solutions which suppose an advance on the state of the art have been proposed. Taking into account the context of the thesis it is of utmost importance that the proposed methodologies could be easily adapted to work in any kind of manipulators and be directly used in an I-AUV.

A brief description of the objectives proposed for each of the groups can be read below:

- **Scene reconstruction:** Before starting a grasping, it is needed to obtain as much information as possible about the object of interest. Consequently, an approach for reconstructing the scene where the object is located, in an autonomous way is the main contribution of this objective.

- **Grasp planning:** After the reconstruction of the scene, a plan for properly grasping the target must be programmed. The goal of this objective is to develop a methodology for detecting the object in the point cloud of the scene, approximate its real dimensions from its partial reconstruction and defining the best feasible pose for grasping it, taking into account the characteristics of the manipulator.

[4] Video of the MERBOTS final experiment https://www.youtube.com/watch?v=1xECxNb0-dQ

- **Grasp execution:** Once the pose for grasping the object is defined, the manipulator must autonomously reach this pose at the same time that different tasks are accomplished. To do so, a framework able to control the movements of the system is developed.
- **Improving reliability:** During the movements of the manipulator, it is of paramount importance to control that these are the correct ones and readdress them if needed. This objective focuses on the design of a toolbox able to monitor the manipulator state and assure an expected behaviour.

1.4 Outline

The different topics introduced in this thesis are presented in 7 chapters structured as follows.

In **Chapter 2** a methodology for reconstructing an underwater scene is presented, together with an algorithm for optimizing the process of reconstruction. This methodology have been compared with other 3D reconstruction techniques.

Chapter 3 describes a kinematic control framework for redundant robots, that integrates the treatment of multiple tasks, multiple kinematic chains, different joint priorities and hard constraints.

An algorithm that estimates the best feasible pose to grasp an unknown object is shown in the **Chapter 4**. Three experiments demonstrate the suitability of the algorithm.

A methodology for guiding the manipulator end-effector to a desired pose and then grasping the object is presented in **Chapter 5**. The results obtained using the methodology in a real system are also shown.

Chapter 6 describes two algorithms for detecting error in the trajectory followed by the arm and correct them. Both algorithms have been tested and the results are also presented.

Finally, conclusions and future work are detailed in **Chapter 7** summarizing the work developed in the thesis.

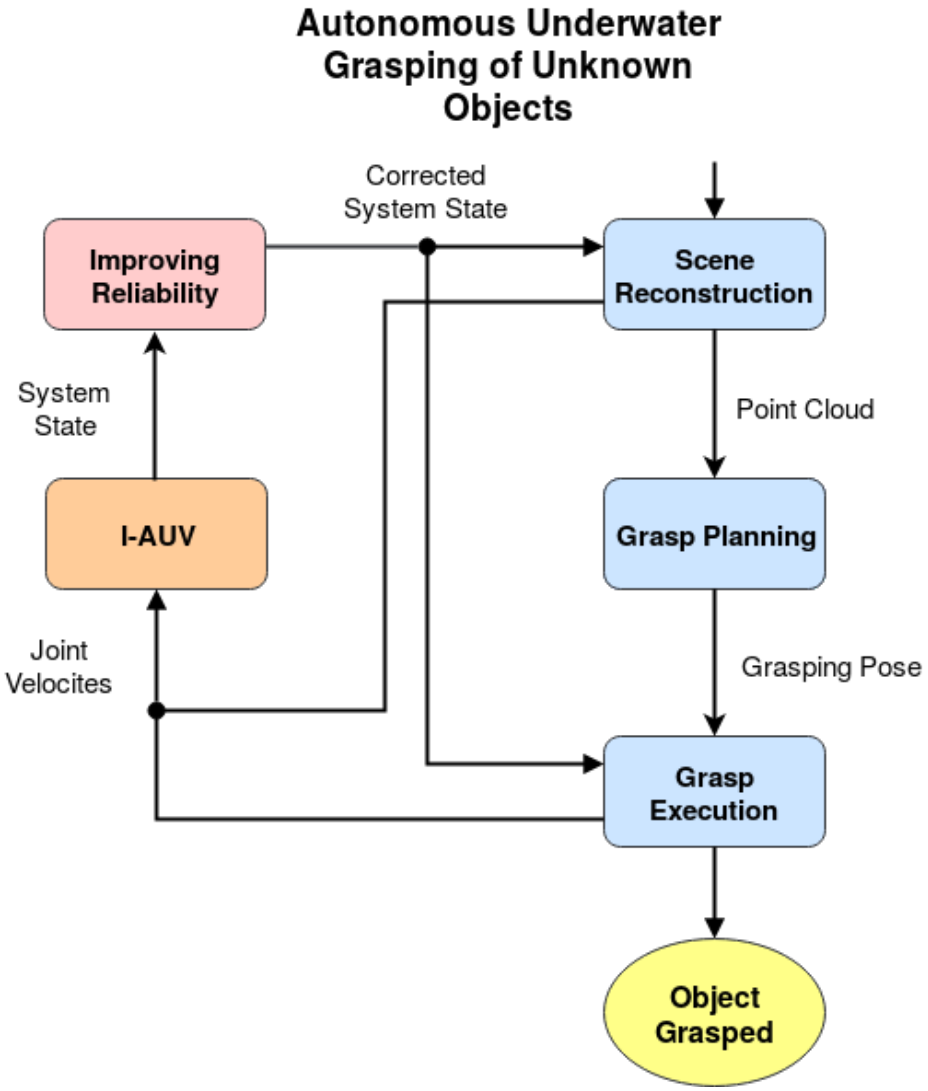


Figure 1.3: Flowchart of an autonomous underwater grasping intervention. This is divided in three phases (blue) which has as a result the object grasped (yellow). Two of the phases require the use of the I-AUV (orange) to be performed. At the same time, the I-AUV movements have to be monitored to detect errors and correct them (pink).

Chapter 2

Scene Reconstruction

This chapter presents an approach for autonomously reconstructing an underwater scene in real time. The approach uses a robotic manipulator with a laser projector and a monocular camera attached at the forearm. Through the movement of the arm, the scene is scanned by the laser. At the same time the camera is taking pictures of the scene projected by the laser. The laser peaks that appear in the images are detected and their 3D position is estimated, obtaining, in this way, a point cloud of the scene (see Fig. 2.1).The approach has been tested in a simulator and using a real system. The results have been compared with other 3D reconstruction techniques in underwater scenarios.

Along with this approach, an algorithm for optimizing the process of reconstruction has also been presented. This algorithm takes advantage of the limited zone of interest for the manipulation procedure, to reduce the time spent in detecting the laser peaks on the images.

The proposed approach have been published in international conferences. Concretely, in [Peñalver et al., 2015a] the approach was firstly presented and some simulated results demonstrated its effectiveness. In [Peñalver et al., 2017], experiments using a real system were shown.

2.1 Motivation

As was mentioned in the Introduction, the behaviour of an I-AUV and an AUV differs when the vehicle is close enough to an object of interest. In that moment and, previous to the manipulation, it is necessary to find out some characteristics of this object, specially whether it is unknown. Some of those characteristics are position, size, shape, etc. If the acquisition of the information about the object is not accurate enough, the probability of

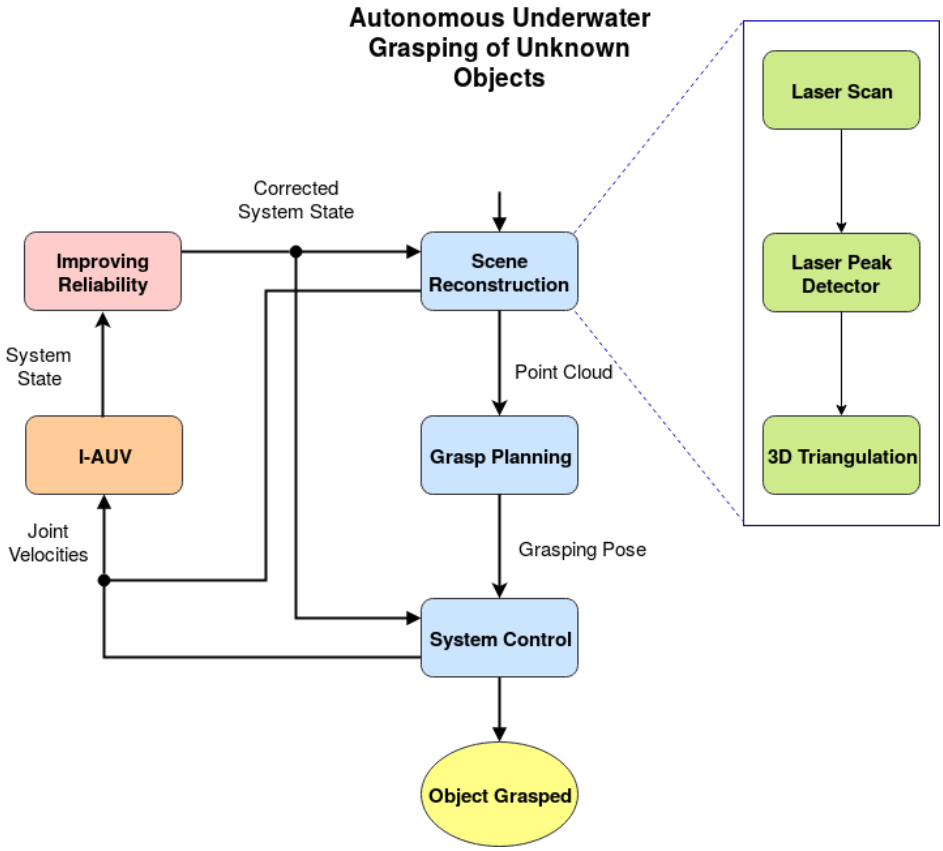


Figure 2.1: Flowchart of an autonomous underwater grasping intervention. The first of the phases of the intervention consists on reconstructing the scene. This phase is composed by three subphases (green). First the scene is scanned using a laser, then the laser peaks are detected and finally these detections are triangulated to obtain a 3D point cloud.

obtaining a suitable grasping plan decreases dramatically and with it, the likelihood of performing the grasping successfully.

For that reason, a good reconstruction of the target is a crucial part of the autonomous manipulation. Furthermore, this reconstruction must be accurate and as complete as possible.

2.2 State of the Art

A large number of underwater applications require high resolution and accurate 3D reconstruction for underwater objects. Archaeological and biological applications, intervention tasks or industrial facilities inspections are just some examples. In these applications, sensors have to work at short distances to obtain 3D information of an object accurately.

Different methods exist for obtaining 3D information in these situations. They can be generally classified according to the sensor used.

2.2.1 Sonar

Sonar, an acronym for sound navigation and ranging, is a technique that uses sound propagation to navigate, communicate with or detect objects on or under the surface of water. There are two general methods of using sonar, passive and active.

Passive sonar are used primarily to detect acoustic signals in an underwater environment. Passive sonar does not emit its own signal, its purpose is to detect the acoustic signals emanating from external sources. This kind of sonar cannot be used for obtaining 3D reconstructions.

Active sonar transducers emit an acoustic signal or pulse of sound. If an object is in the path of the sound pulse, a portion of the sound is reflected back to the sonar transducer. Then, by determining the time between the emission of the sound pulse and its reception, the transducer can determine the range and orientation of the object.

Sonar-based approaches are the most extended because acoustic devices do not suffer from turbidity. They have been extensively used for underwater 3D mapping. Some examples are, the generation of a 3D map of the American Samoa [Lundblad et al., 2006], the Olympic Coast National Marine Sanctuary [Intelmann, 2006] or the Geisha Guyots [Vogt and Smoot, 1984].

However, acoustic approaches cannot obtain as high resolution and accuracy as optical approaches. Thus, the optical ones are more suitable for short-distances operations [Massot-Campos and Oliver-Codina, 2015]. Only in a few works a sonar device has been used for reconstructing small objects.

One of the first work was [Rosenblum and Kamgar-Parsi, 1992], where using low-resolution data provided by a 3D imaging sonar of that time, small objects were reconstructed obtaining accurate size and positional information, but not as good results were achieved regarding the shape.

A sequence of sonar images are used to estimate the three dimensional aspects of underwater objects in [Zerr and Stage, 1996]. The data are recorded by a sector scanning sonar, carried by an underwater vehicle which follows a circular track around the object.

Using the objects shadows in a sonic image, in [Aykin and Negahdaripour, 2013], the 3D reconstruction of an hemi-cylinder (concave and convex side) and an hemi-sphere rock is obtained.

In [Guo, 2013], an approach for reconstructing a scene using a single beam sonar is tested in a pool by obtaining the model of a 60x40x30 cm cube.

In [Negahdaripour et al., 2009], they studied the 3D reconstruction of underwater objects by opti-acoustic stereo imaging, it means by combining imaging sonars with conventional cameras.

2.2.2 Structure from Motion

This method consists of taking images of an object or scene using a monocular camera. Then, some features are extracted from the camera shots. These features are matched between consecutive frames with the goal of calculating the 3D relative camera trajectory. Knowing the pose of the camera when each photo was taken, by means of triangulation methods, the 3D reconstruction of the object or scene can be calculated.

These kind of methods only need a still camera or a video recorder for working, which makes them a much cheaper approach than acoustic ones. Other advantages with respect to sonar are, as was said before, the higher accuracy they provide, and even in some cases, the possibility of obtaining color information.

Otherwise, this method is not useful in turbid waters, on untextured floors or in the darkness. Even increasing the light with artificial methods could only worsen the situation since the light is strongly backscattered by the suspension particles in the water [Wang et al., 2000]. Another drawback is that images need to be postprocessed to obtain the 3D reconstruction.

There are a large variety of works where structure from motion is used to reconstruct scenes with small objects. And the accuracy obtained in most of them could be enough for performing grasping interventions. Some of these works are named hereinafter.

In [McKinnon et al., 2011], a high resolution camera, 2272x1704 pixels, together with the SURF features extractor method have been used to re-

construct a piece of coral, obtaining an accuracy of 0.7 mm at distances between 1 and 1.5 meters.

In [Cocito et al., 2003], they describe a method for accurately measure the volume, surface area and other morphometric measurements of biological objects. The method has been tested using images captured by divers which always contain a cube to recover scaled 3D data. The accuracy achieved is around 1 cm.

A framework for structure recovery using video sequences was described in [Nicosevici et al., 2009]. SIFT features extractor was used with an average error of 11 mm.

The documentation of an archaeological site where experimental cleaning operations were conducted is shown in [Bruno et al., 2013]. A commercial software, Photoscan by Agisoft, was used to perform a multi-view 3D reconstruction with an average accuracy of 4.5 mm.

A new refractive structure from motion algorithm that takes into account the refraction of glass ports in water was presented in [Jordt-Sedlazeck and Koch, 2013]. By considering the refraction coefficient between the air-glass-water interface, the results can be improved.

2.2.3 Stereo Vision

This method follows the same working principle as structure from motion, but in this case, the camera has only two frames which take pictures at the same time. Thus, the features are only matched between left and right frames. Once the stereo rig is properly calibrated, the relative position of one frame with respect to the other is known, and therefore, the 3D position of the features can be triangulated.

Stereo vision is as cheap as the Structure from Motion method, and thanks to the fact that the relative distance between the two images is fixed and known, the accuracy obtained is even higher. Another advantage is that the reconstruction can be done in real time. It is not needed to move the camera for getting a reconstruction of a scene.

Same as in the Structure from Motion methods, this approach is not the most appropriated in turbid waters, untextured scenarios or dark places. Another drawback could be that the reconstruction is only done from one point of view, leaving larger parts of the target unreconstructed.

Nowadays, this method is the most used for reconstructing small objects in scenarios with good texture and well illuminated. Some examples are briefly detailed below.

A system which consist of an inexpensive underwater stereo camera, a turn table and a personal computer have been built in [Kumar and Kumar, 2011] to reconstruct 3D models of underwater objects. This system is robust

under illumination changes thanks to the use of SIFT features.

Commercial GoPro cameras are used to set a 35 mm baseline stereo rig in [Schmidt and Rzhanov, 2012]. With this system, they perform micro bathymetry using SIFT features and they achieve a resolution of 3 mm in their reconstructions.

A method for underwater localization and mapping for detailed inspection tasks was explained in [Servos et al., 2013]. This work demonstrates that stereo SLAM results for underwater applications can be improved by accounting for refraction in the stereo matching and SLAM algorithms.

In [Brandou et al., 2007], the stereo system IRIS is hung from the tip of the arm of the Victor6000 ROV. The system uses SIFT combined with RANSAC to discard outlliers. In order to test the system, small-scale natural underwater objects have been reconstructed.

A low-cost stereo system is used to make accurate in situ measurements of fishes in [Costa et al., 2006]. A neural network is built to correct the measurements obtaining errors lower than 1 cm.

2.2.4 Structured Light

This system is based on a camera and a color projector. The projector casts a known pattern on to the scene, normally a set of light planes. Then, the way that this pattern is deformed when striking surfaces is used by the vision system to triangulate the depth and surface of the objects in the scene.

This method is also a cheap alternative. Furthermore, it can work on untextured grounds on short distances. It gets good accuracy even in the darkness. And the projector can emit in the wavelengths that are less absorbed by the water and thus decrease the scattering coefficient.

The main drawbacks come from the election of the projected pattern. If this pattern is simple, it is needed to move the projector in order to project the whole scene, producing a time consuming. On the other hand, if the pattern is complex, the identification of different pattern regions on the images becomes ambiguous so that the correspondence problem is not directly solved.

Despite this, the structured light is the method that better fits with the problems that appear in underwater scenarios. Some works where this approach has been used, are hereinafter explained.

A laser line has been mounted onto a servomotor that can rotate 45 degrees in [Hildebrandt et al., 2008]. Then a 640x480 pixels of resolution camera records the scene whilst the laser scans the scene. Calibration is made in this article with a novel rig consisting of a standard checkboard next to a gray surface on one side. This is because they explain that laser is better detected on gray surface.

In [Kondo et al., 2004], a sheet laser beam has been attached to the Tri-Dog I-AUV. They scan the floor by moving the vehicle, achieving a resolution in their 3D reconstructions of 40 mm at three meter from the floor. They also use the system for governing the robot. They track the images in real time to keep a safe distance from the seabed.

The authors of [Caccia, 2006] mounted a camera and a vertical laser stripe in a translation stage. 3D information is interpolated from a data table previously acquired from calibration. Even though a laser peak is detected in the image, its depth value is calculated from the four closest points in the calibration data.

In [Massot-Campos et al., 2015], the authors designed a laser-based structured light system which uses as a pattern 25 parallel lines. In one camera shot, this solution is capable of recovering sparse 3D information, whilst with two or three shots, denser information can be obtained. The system is targeted at underwater autonomous manipulation stages where a high density point cloud of a small area is needed.

A fixed camera has been mounted to the AUV Girona 500 frame and a laser stripe on an underwater manipulator carried by the vehicle in [Prats et al., 2012b]. The laser scans the scene by moving the arm, and at the same time the camera records the scene. The images recorded are used to obtain a point cloud. This is used to determine the target grasping points. The sea bottom is tracked to estimate the robot motion during the scanning process, so small misalignments between the data can be compensated.

2.3 Multi-View Laser Reconstruction

The first of the goals of this thesis is to develop a methodology to autonomously reconstruct an underwater scene where an object of interest is located. The object is supposed to be close enough to the I-AUV to be manipulated without moving the vehicle. Moreover, the reconstruction should be as accurate and complete as possible.

The choice of a proper device to perform the reconstruction is vital for achieving this goal. The vehicle is close to the target, thus the level of scattering and absorption could be high. Moreover, if the object is placed at a depth not reachable by sun light, the scene could be completely dark.

If the worse cases are taken into account, the structured light device would be the best option because it performs well with poor visibility if the target is close to the system.

Concerning the structured light devices, it is also important the pattern that is projected on the scene. Complex patterns have the advantage of allowing a non dense reconstruction in just one shot, but the scattering effect increases because of the wider water illuminated. Another drawback

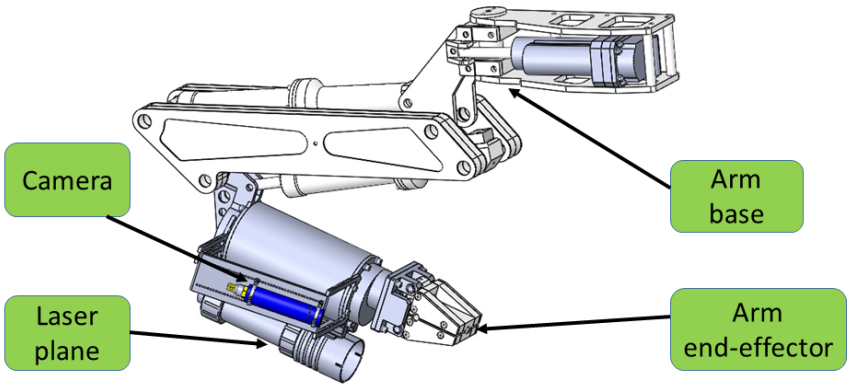


Figure 2.2: Eye-in-Hand configuration to autonomously reconstruct underwater scenes.

could be the complexity of the process of matching the illuminated pixels with the projected pattern, in order to obtain its 3D positions. On the other hand, with a simple straight line, it is needed to scan the whole scene. This process is more time consuming but the obtained point cloud is much more dense.

The last important topic to think about, is the color of the projected light. The color spectra between blue and green is the optimum to avoid absorption and backscattering.

For our approach, a green laser that projects a straight line has been used. This laser is placed on the forearm of the robotic arm. So, moving the arm, the laser light scans the scene like in [Prats et al., 2012a]. The novelty of this approach is in the position of the camera. It has been placed on the forearm of the robotic arm (eye-in-hand configuration, see Fig. 2.2). Using this new approach both the camera and the laser projector are moved with the arm during the scan of the scene. Up to now, the cameras were placed in a fixed position while the reconstruction, having all the time the same point of view.

With this approach, the object could be reconstructed from different point of views, obtaining a more complete 3D model. Another advantage is that using the arm, the camera can be closer to the object than in the usual approaches, obtaining more precision in the reconstruction and avoiding even more the problems of scattering and abortion.

Hereinafter, the process for obtaining a 3D reconstruction of a scene using this methodology is detailed.

2.3.1 Calibration

A good calibration of the whole system is a crucial step for an accurate reconstruction. A little error in the calibration of any of the parts of the system, could lead to a big error in the position of each point of the 3D point cloud that represents the scene. In the proposed calibration methodology, which consists on three stages, an external stereo camera is needed.

1. In the first stage, the intrinsic and extrinsic parameters of the stereo and the eye-in-hand camera (${}^{S}\mathbf{M}_{eye}$) are obtained using a checkerboard pattern [Svoboda et al., 2005].

2. In the second stage, an ARmarker is placed in a known position of the gripper, thus its position with respect to the end-effector of the gripper can be easily calculated (${}^{M}\mathbf{M}_{E}$). Knowing that, the marker is detected by the stereo camera using the ARToolkit library [Kato and Billinghurst, 1999] and its pose with respect to the stereo camera is estimated (${}^{S}\mathbf{M}_{M}$). Using then the three already calculated transformations, the transformation between the end-effector and the eye-in-hand camera can be obtained as follows: ${}^{E}\mathbf{M}_{eye} = ({}^{S}\mathbf{M}_{M} * {}^{M}\mathbf{M}_{E})^{-1} * {}^{S}\mathbf{M}_{eye}$.

3. For the third stage, the laser needs to be switched on. Then, a picture is taken using the stereo camera. The two images obtained are processed, in order to leave only the pixels where the laser is seen. Next, the 3D position of the parts of the scene projected by the laser is obtained by triangulation. This process needs to be done adding and removing objects in the scene. Doing that, we obtain 3D points that belong to the laser plane. All these 3D points are added to the same point cloud and then, the RANSAC algorithm is used to determine the planar parameters of this laser plane [Inglis et al., 2012]. These parameters are referenced to the stereo camera. But using the transformation between the end-effector and the stereo camera (${}^{S}\mathbf{M}_{E} = {}^{S}\mathbf{M}_{M} * {}^{M}\mathbf{M}_{E}$), it is possible to reference this plane to the end-effector.

To sum up, after the three stages, the intrinsic parameters of the eye-in-hand camera has been calculated and the transformations between both camera and laser, with respect to the end-effector, have been obtained (${}^{E}\mathbf{M}_{L}$, ${}^{E}\mathbf{M}_{eye}$).

2.3.2 3D Reconstruction

Once the system is fully calibrated and ready to start the reconstruction, the robotic arm is moved at a constant velocity following a predefined trajectory.

At the same time, the eye-in-hand camera captures images of the scene projected by the laser.

In order to extract the 3D information of the scene using those images, it is necessary to find out the pixels which are illuminated by the laser. Then, the 3D position of each pixel is triangulated taking into account the transformations between the different parts of the system in the moment the images were captured.

Laser Peak Detection

The process of detecting which pixels of an image are illuminated by a laser is commonly known as *Laser Peak Detection*.

In our methodology, the image is converted from RGB to HSV color model. Then, the pixels which are out of a predefined threshold of huge, saturation and value, are discarded (this threshold has been obtained experimentally). Next, four operations are applied over the thresholded image. These operations are: erosion, dilation followed by dilation and erosion. With these operations, the pixels detected as laser which are far from other pixels detected are discarded, i.e. the outliers are removed.

Due to the fact that the laser pattern is a straight line and that the eye-in-hand camera is placed parallel to that line, there can just be a point illuminated by the centroid of the laser at each column of the image. Thus, for each column, the pixel, which is into the threshold, with the highest intensity is selected. Then, the center of masses algorithm [Forest et al., 2004] is applied to this pixel together with the five pixels above and below it. Using it, the pixel illuminated by the centroid of the laser is obtained with a subpixel accuracy (see Fig. 2.3).

3D Triangulation

Once the pixels illuminated by the centroid of the laser are detected, their 3D position must be calculated. For that, a methodology similar to the proposed in [Prats et al., 2012a] is used.

After a properly calibration of the system the images given by the camera are undistorted. So, a pixel with row and column coordinates r and c, defines a line in projective coordinates given by the column vector:

$$\mathbf{l} = (l_x, l_y, 1) = \left(\frac{c - c_0}{p_x}, \frac{r - r_0}{p_y}, 1 \right) \tag{2.1}$$

Being c_0, r_0, p_x and p_y, the camera intrinsic parameters, i.e. the principal point in pixels and the focal length to pixel size ratio. If a pixel in the image belongs, in addition, to the projected laser plane, the intersection of the camera ray with the laser plane gives the 3D coordinates of the point. The

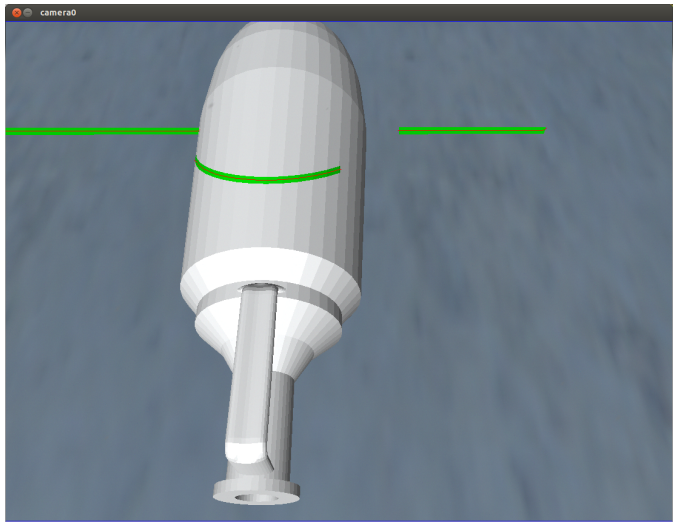


Figure 2.3: Image recorded by the eye-in-hand camera during the reconstruction. The red points inside the green laser, are the pixels illuminated by the centroid of the laser.

line defined by the camera ray can be expressed with its parametric equation as:

$$\mathbf{P} = (X, Y, Z) = \lambda \mathbf{l} \tag{2.2}$$

If, in addition, a 3D point, $\mathbf{P}$, belongs to the laser plane, it holds that:

$$(\mathbf{P} - \mathbf{P_0})^T \mathbf{n} = 0; \tag{2.3}$$

where $\mathbf{n}$ is the plane normal given in camera coordinates, and $\mathbf{P_0}$ is a 3D point that belongs to the plane. Merging equations (2.2) and (2.3) leads to:

$$\lambda = \frac{\mathbf{P_0}^T \mathbf{n}}{\mathbf{l}^T \mathbf{n}} \tag{2.4}$$

and the final 3D coordinates of the point are given by $\mathbf{P} = \lambda \mathbf{l}$.

In order to compute these equations, it is necessary to know the laser plane equation with respect to the camera (given by the point $\mathbf{P_0}$ and a normal $\mathbf{n}$), which was obtained in the calibration process.

With that methodology, the 3D positions of the points are related to the camera (${}^{eye}\mathbf{M}_P$), but with the eye-in-hand approach, the camera is in continuous movement. In order to be able to join all the obtained 3D points in the same point cloud, those positions must be related to a fixed frame. The base of the arm could be a good frame to refer the points because it

is fixed during the reconstruction, and knowing the joint values of the arm in the moment the image was captured and using the Direct Kinematics of the arm, it is easy to refer the point to the base (${}^{B}\mathbf{M}_{P}$).

$$ {}^{B}\mathbf{M}_{E} = DK(q_0 q_1 ... q_n) \tag{2.5} $$

$$ {}^{B}\mathbf{M}_{P} = {}^{B}\mathbf{M}_{E}{}^{E}\mathbf{M}_{eye}{}^{eye}\mathbf{M}_{P} \tag{2.6} $$

2.4 Optimization

Due to the limited memory capacity on the usual computers used in the I-AUVs, in most cases it is not possible to record all the captured images and to process them afterwards. So, when an image is captured, it is processed immediately, and the next images are discarded until the process is complete. It means that the more time you spend extracting 3D information from one image, the less dense your final 3D point cloud will be.

Moreover, the aim of the reconstructions proposed is to discover the characteristics of an object that is desired to be manipulated, and it is supposed that the object is inside the arm workspace. It means that the part of the scene out of this workspace is not important. It is even desirable that the system does not reconstruct that part in order to avoid outliers and to keep the final point cloud as lightweight as possible without losing any detail of the target.

In order to reduce the time of processing the images and taking into account the aforementioned, a method to optimize the reconstruction has been developed. This method consists on calculating the pixels of the image which can be discarded because, in case they are illuminated by the laser, the part of the scene illuminated is out of the workspace of the arm (see Fig. 2.4). As a result, the images given to the laser peak detector are smaller and therefore their processing is faster.

Before the beginning of the intervention, the user must introduce two parameters: (1) The *max_radius* which is the radius of the smallest sphere that wraps all the arm workspace, having as a center the base of the arm, and (2) the *min_radius* which is the radius of the bigger sphere with the base of the arm as a center, which does not intersect with any part of the arm workspace.

The optimization takes as input the image that needs to be processed. Then for each column of the image three steps are applied.

1. The finality of the first step is to find the line which is the intersection between the laser plane and the plane which crosses the center of projection of the camera and the pixels of the column that is being processed. For this, two pixels of this column are selected and using

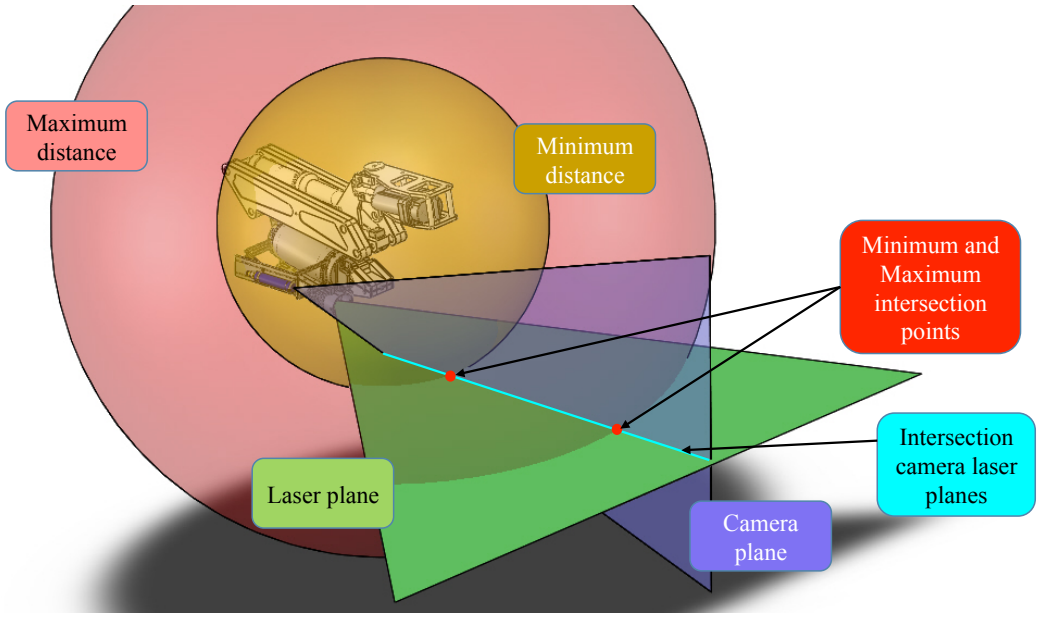


Figure 2.4: Time optimization method for extracting 3D information from an image.

the *3D Triangulation* algorithm explained in the last section, the 3D positions of the two points that would be illuminated by the laser and seen in these pixels are calculated. These two points ($\mathbf{P_1}$, $\mathbf{P_2}$) belong and define ($\mathbf{L_1} = \overrightarrow{\mathbf{P_1P_2}}$), which is the line we were looking for.

2. The second step consists in calculating the intersection point between $\mathbf{L_1}$ and the two spheres with center the base of the arm and with radius *max_radius* and *min_radius*. Being $\mathbf{A} = (A_x, A_y, A_z)$ and $\mathbf{B} = (B_x, B_y, B_z)$ two points that belong to a line, $\mathbf{C} = (C_x, C_y, C_z)$ the center of a sphere and r its radius, the intersection points between the line and the sphere can be calculated by:

$$a = (B_x - A_x)^2 + (B_y - A_y)^2 + (B_z - A_z)^2 \tag{2.7}$$

$$\begin{aligned} b = 2((B_x - A_x)(A_x - C_x)+ \\ (B_y - A_y)(A_y - C_y)+ \\ (B_z - A_z)(A_z - C_z)) \end{aligned} \tag{2.8}$$

$$c = (A_x - C_x)^2 + (A_y - C_y)^2 + (A_z - C_z)^2 - r^2 \tag{2.9}$$

$$\delta = b^2 - 4ac \tag{2.10}$$

$$d = \begin{cases} \nexists & \text{if } \delta < 0 \\ \frac{-b}{2a} & \text{if } \delta = 0 \\ \frac{-b \pm \sqrt{\delta}}{2a} & \text{if } \delta > 0 \end{cases} \tag{2.11}$$

$$\textbf{intersection\_point} = (\mathbf{B} - \mathbf{A})d + \mathbf{A} \tag{2.12}$$

So, using the points obtained in the last step (P_1 and P_2), the radius introduced by the user and being the center of the spheres the base of the arm, the intersections between L_1 and the two spheres can be calculated.

3. In the third step, it is calculated in which rows of the image the points obtained in the previous step would be seen. For each sphere, there may be two intersection points, in this case, the point that is placed behind the camera is discarded. In order to obtain the rows for the two points, the equations (2.1) and (2.2) are merged:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \lambda \frac{c - c_0}{p_x} \\ \lambda \frac{r - r_0}{p_y} \\ \lambda \end{pmatrix} \tag{2.13}$$

And then, the row is given by:

$$r = \frac{Y p_y}{Z} + r_0 \tag{2.14}$$

Once the two rows for each column are calculated, the pixels which are not between these two rows are discarded and the resultant image is sent to the *Laser Peak Detector* algorithm.

2.5 Experimental Results

In order to validate the functionalities of the algorithms previously described, some experiments have been performed. The multi-view laser reconstruction approach, have been firstly tested using a simulator. After demonstrate that the methodology performs successfully in simulation, a real system has been configured to reconstruct real objects using this method. In both cases, the obtained results have been compared with reconstructions performed using other approaches. Finally, the time improvements reached by the optimization algorithm have also been shown.

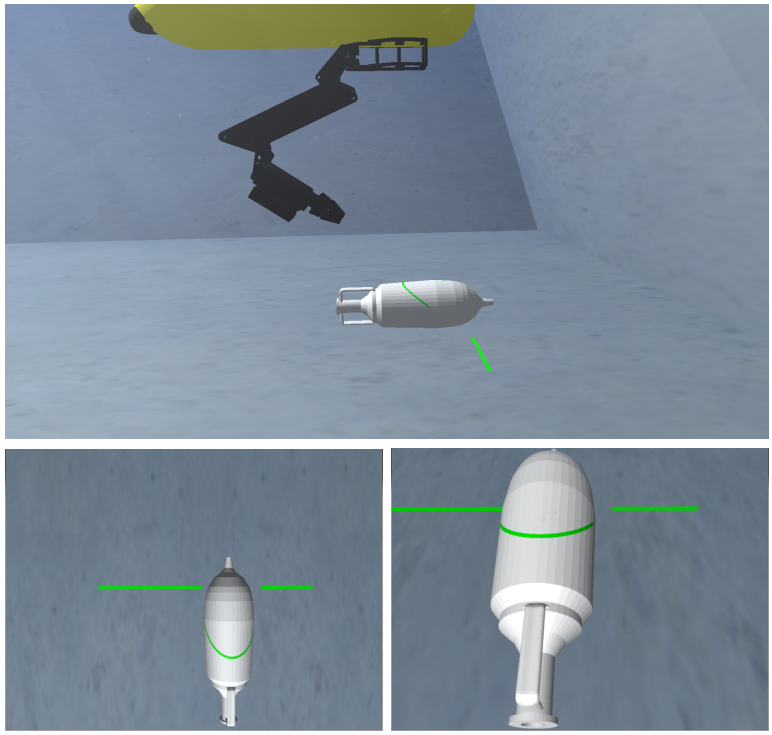


Figure 2.5: ARM 5E with a stripe laser projector during a scan (top); Image recorded from the fixed camera (left). Image recorded from the eye-in-hand camera (right)

2.5.1 Simulated Multi-View Laser Reconstruction

First of all, the proposed algorithm has been tested using the underwater simulator UWSim [Prats et al., 2012c]. UWSim is a software tool for visualization and simulation of underwater robotic missions. The software visualizes an underwater virtual scenario that can be configured using standard modelling software. Controllable underwater vehicles, surface vessels and robotic manipulators, as well as simulated sensors, can be added to the scene and accessed externally through network interfaces. This allows to easily integrate the visualization tool with existing control architectures.

In order to simulate our methodology, it is needed to load in the simulator a model of a robotic manipulator, a laser stripe projector and a camera. The manipulator is the light-weight ARM5 [Fernández et al., 2013]. The projector simulates a stripe pattern which consists of a green rectangle of 1024 pixels wide and 10 pixels high. And the camera has a resolution of 640x480 pixels. Both the camera and the laser have been attached to the

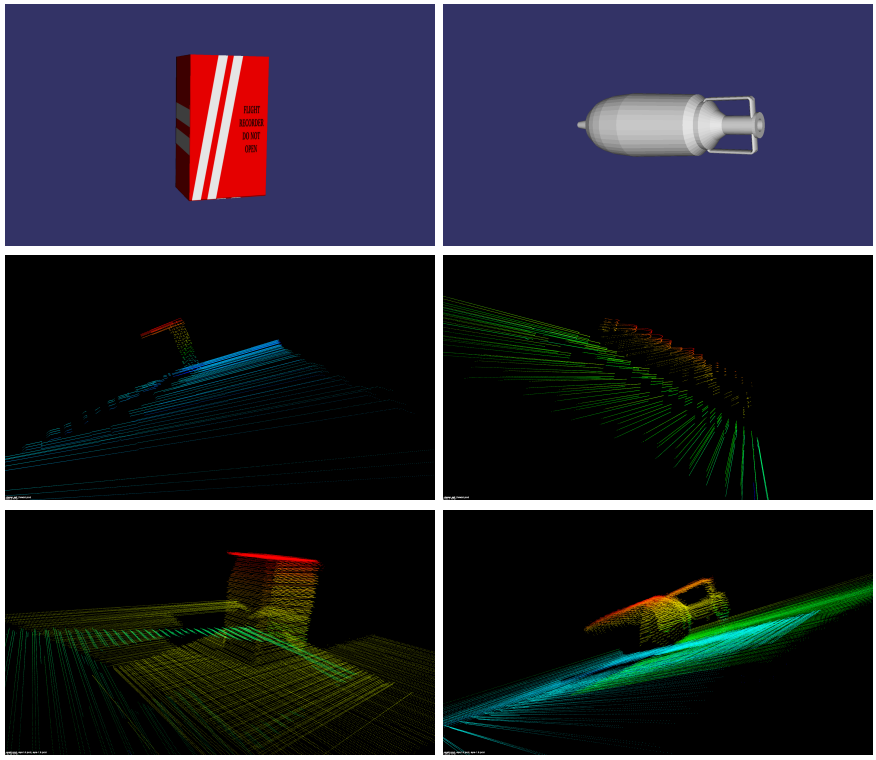


Figure 2.6: Reconstruction of two objects: black-box and amphora (left to right). Original model, fixed camera and multi-view approach (top to bottom)

forearm of the manipulator as is indicated in the Fig. 2.2.

An extra camera has been placed at the base of the robotic arm facing downwards. This is a 1024x768 pixels camera. This camera is used to simulate the usual laser reconstruction approaches, where the camera is fixed during the scan.

In the Fig. 2.5, a simulated scene together with a picture taken from the eye-in-hand and the fixed camera can be seen.

Two objects have been reconstructed using both approaches (see Fig.2.6).

The first object is a black-box mockup similar to the ones that can be found in the airplanes. Thanks to he possibility of reconstructing the box from different point of views, using the eye-in-hand camera, even four sides of the box can be reconstructed. On the other hand, using the fixed camera only two sides are reconstructed.

The second object is an amphora. This object makes possible to notice how the reduced distance between the camera and the object in the multi-

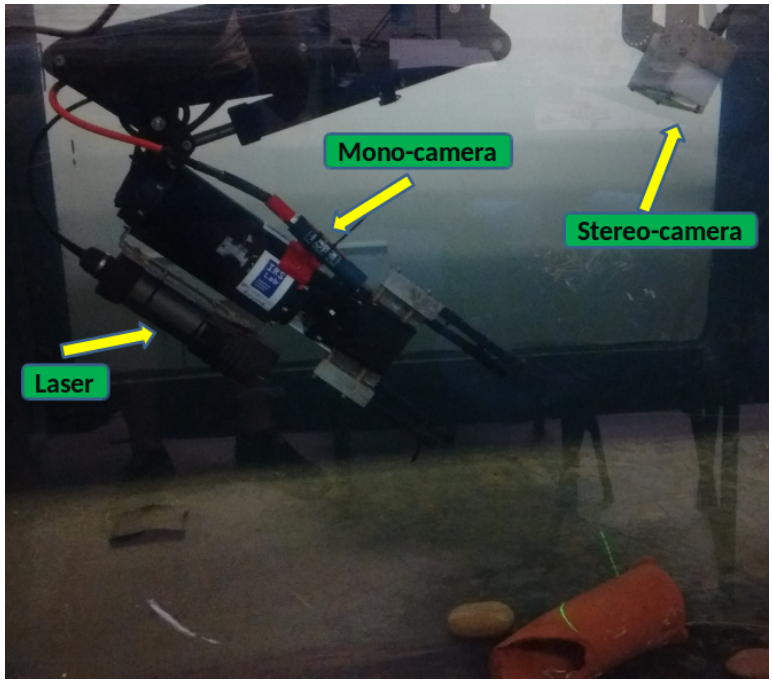


Figure 2.7: Real system configuration for testing the reconstruction approaches.

view approach, increases the accuracy of the final reconstruction.It is also worth noting how even the handle of the amphora has been properly reconstructed. For reconstructing this thin part accurately, it is required both, to get the camera close to the handle and to reconstruct it from different views.

2.5.2 Multi-View Laser Reconstruction in a Real System

Once the correct behaviour of the methodology has been checked in a simulator, it is time to test it using a real system. Like in the simulation, a manipulator (Light-weight ARM5E, previously cited), a laser projector (MKIII [Tritech,]) and a monocular camera (Bowtech DiveCam [Bowtech,]) is required, in addition, a fixed stereo camera (Videre Stereo Camera [Videre,]) is used for comparing different approaches (see Fig. 2.7). The monocular camera is a 640x480 pixels camera which has been attached together with the laser at the forearm of the manipulator. The stereo camera has also a resolution of 640x480 pixels and it has been placed at the base of the arm. The whole system has been introduced in a water tank (see Fig. 2.7).

In this case, three methods of reconstruction have been used. the first one is the multi-view laser reconstruction approach. In the second one, the

laser also scans the scene but the camera used for capturing the images is one of the frames of the stereo camera, which means that the camera is fixed during the scanning. And the last one is the stereo reconstruction approach, explained in the section 2.2.3.

The reconstruction of four objects using the three methods can be found in the Figs. 2.8 and 2.9.

The first object is an amphora with a broken part (see Fig.2.8). The multi-view approach makes possible to reconstruct even inside the hole of the amphora. So, the reconstruction gives more information about the real shape of the object. Moreover, the accuracy obtained with the eye-in-hand camera is higher than using the stereo camera.

The second object is a black-box mockup (see Fig.2.8). As in the simulated experiments, attaching the camera to the manipulator makes possible to reconstruct more sides of the black-box. This object also shows how the structured light approaches work better than other approaches in objects low textured.

The third object to reconstruct is a big amphora (see Fig.2.9). This object is well reconstructed using the three methodologies, but the multi-view one brings a higher accuracy and a more complete reconstruction.

Finally, a small amphora has been reconstructed (see Fig.2.9). This amphora is quite difficult to reconstruct, due to its size, unless the camera is placed really close to the object. This is easier to achieve when the camera is at the arm because it is always less dangerous to move the arm to the object than moving the whole vehicle.

2.5.3 Optimization Test

Finally, the improvements offered by the optimization algorithm has been demonstrated. For that, the time used for detecting the laser peaks without using the algorithm and using it have been measured.

The experiments have been performed in the UWSim simulator. They consists on scanning a scene using different camera resolutions and different reconstruction limits. Then, during the scans, the time inverted in detecting the laser peaks have been accumulated. This time also includes the time used by the optimization algorithm discarding the pixels which are out of the defined workspace, in case the limits are defined.

In order to compare the results, the accumulated times have been divided by the number of images reconstructed in each scan. Thus, the average time consumed for detecting the laser peaks in one image have been illustrated in the next graphic (see Fig. 2.10).

The figure demonstrate that the optimization algorithm reduce the time spent in detecting the laser peaks. As it was expected, the lower the range

between limits is, the more time saved detecting the laser. But, it is also remarkable that even when the limits are distant and the camera resolution is low, the use of the optimization algorithm never increases the time of detecting the laser peaks.

2.6 Conclusions

Previous to an autonomous underwater grasping of an unknown object, it is required to obtain as much information as possible about this object. To do so, reconstructing the scene where the object is located is the first step.

This chapter has presented a new approach to autonomously obtain an accurate and complete 3D reconstruction of an object in an underwater scenario. The approach consists in attaching a laser stripe emitter and a camera at the forearm of a robotic arm. While the arm is moving, the laser scans and the camera records the scene. The fact of positioning the camera at the arm, which is in movement during the scan, makes possible to record the scene from different points of views and from positions closer to the object.

A laser peak detector and a 3D reconstruction algorithms have been used to obtain a 3D point cloud from the images captured. In order to reduce the time of processing these images, an optimization algorithm has been proposed.

Different shaped objects were reconstructed in a very accurate and precise manner, and the results were compared with other reconstruction approaches. Then the improvements that the optimization algorithm produces has also been tested. From the obtained results, we can conclude that this approach improves the state of the art in this particular context.

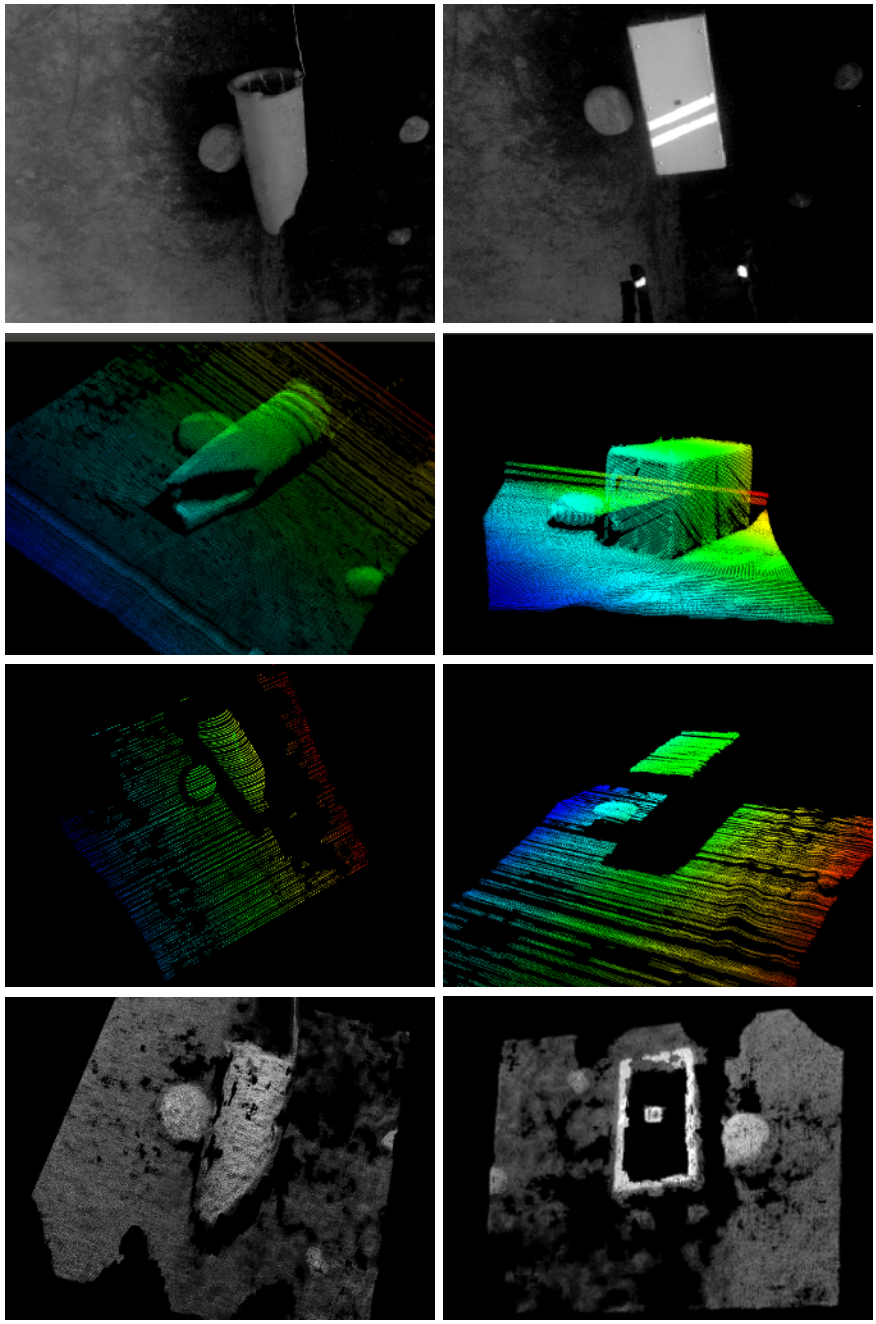


Figure 2.8: Reconstruction of two objects: broken amphora and black-box (left to right). Scene picture, multi-view, fixed camera and stereo approach (top to bottom)

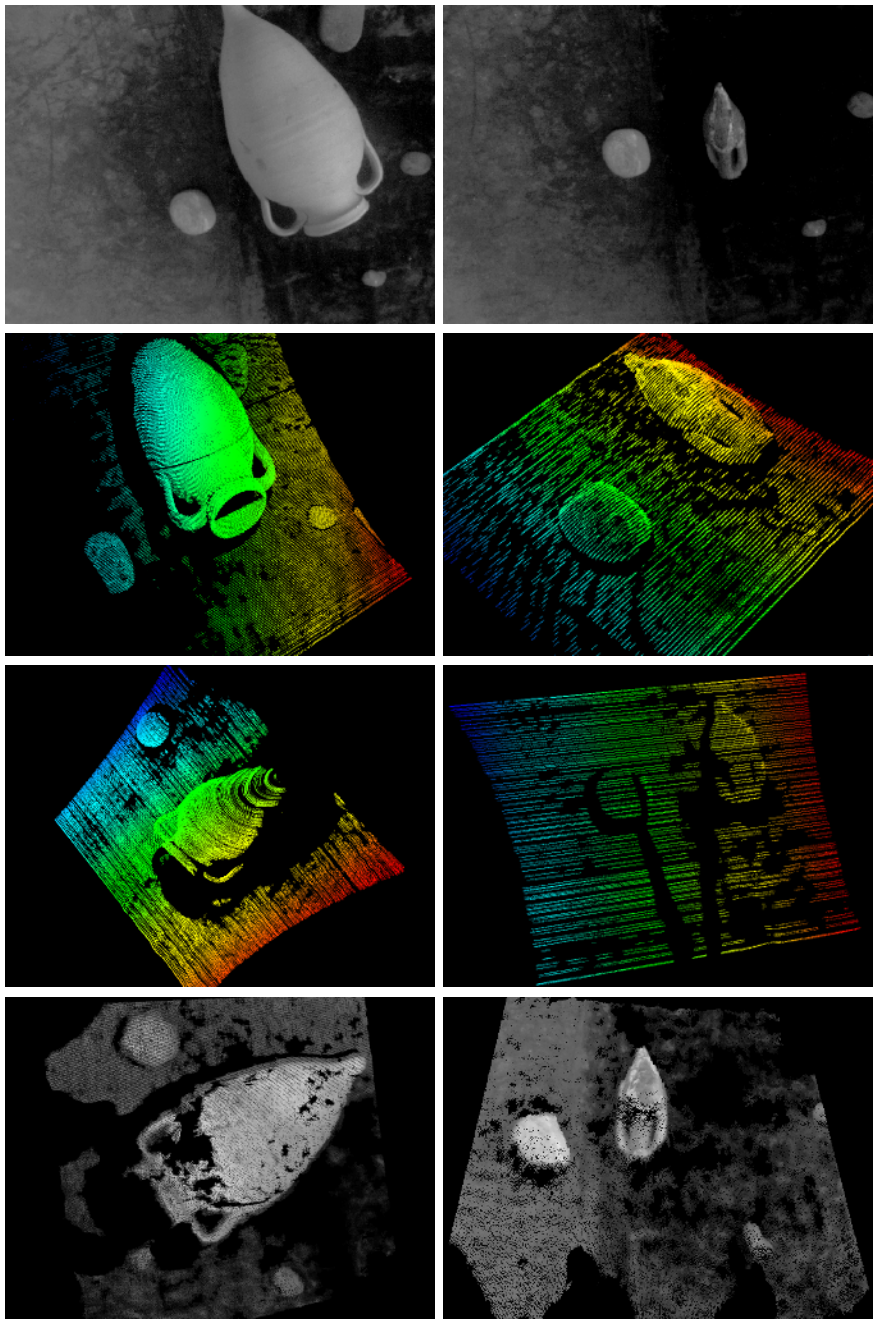


Figure 2.9: Reconstruction of two objects: big amphora and small amphora (left to right). Scene picture, multi-view, fixed camera and stereo approach (top to bottom)

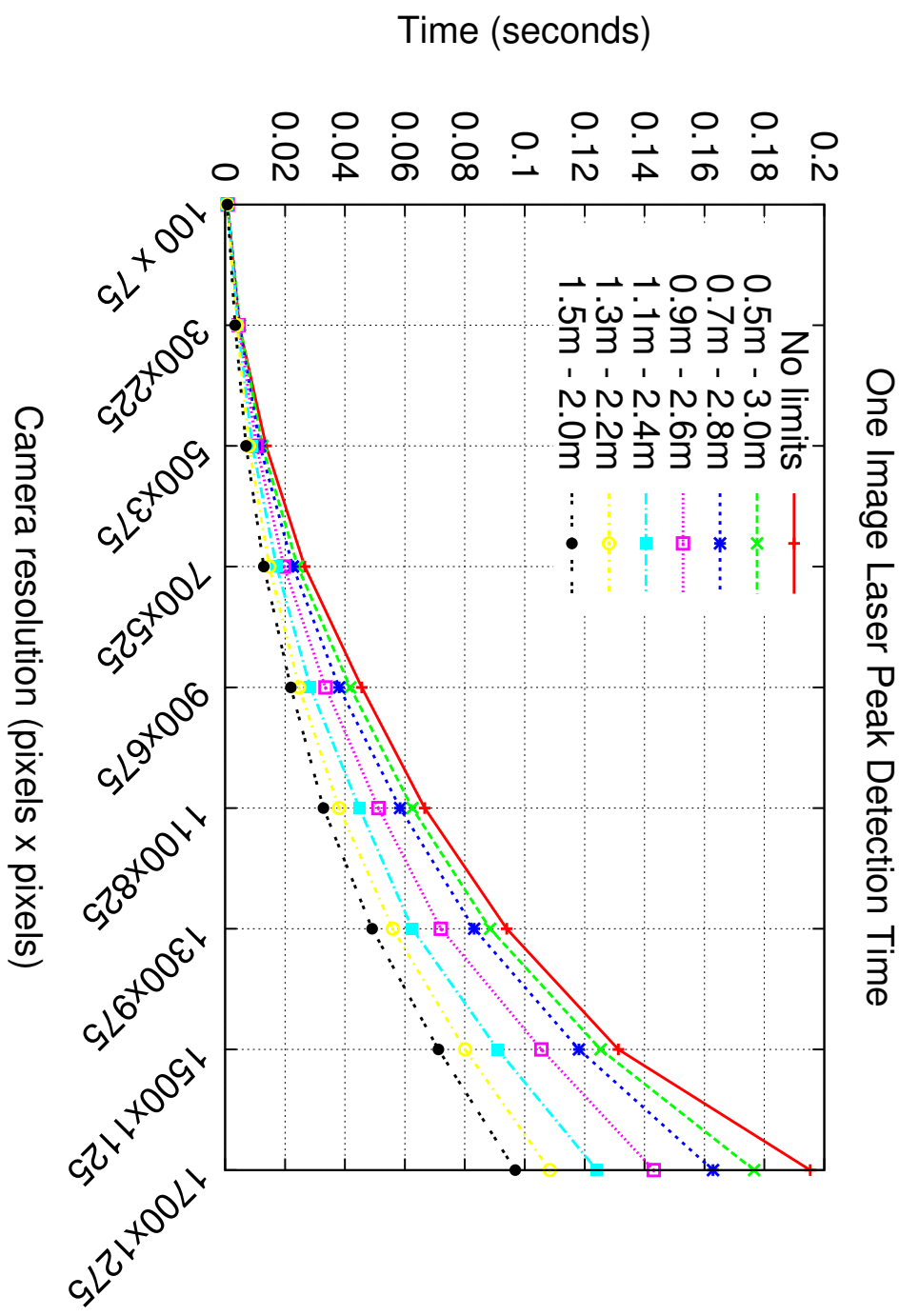


Figure 2.10: Test of the optimization algorithm

Chapter 3

Multi-Task Priority

This chapter introduces a kinematic control framework for redundant robots. This framework is the main contribution of the Grasp Execution phase (chapter 5), but as it is also used in the the Grasp Planning phase (chapter 4), a specific chapter for describing the framework has been incorporated.

The framework integrates, in a unified program, the treatment of multiple tasks, multiple kinematic chains, different joint priorities and hard constraints; that are usually addressed separately. The management of multiple tasks is based on the Reverse Priority method, that has been modified so that it makes possible the assignment of different priorities to each joint in order to accomplish any task. This framework is also suitable for robotic systems with multiple kinematic chains, which could share several joints. Moreover, it can deal with bilateral and unilateral joint and Cartesian constraints, that are considered at each task priority level.

The framework description has been published in [Peñalver et al., 2018].

3.1 Motivation

The ability of robots to perform dexterous tasks is highly related to the redundancy of the system. Kinematic redundancy occurs when a robotic manipulator has more degrees of freedom than those strictly required to execute a given task. Redundancy introduces additional complexity to the system but, if properly managed, it may be used to avoid singularities, joint limits, obstacles, etc.

I-AUVs redundancy must be properly controlled to allow the manipulator to reach a desired pose without exceeding any limit and, if it is possible, fulfilling a set of tasks with different priorities.

3.2 State of the Art

The robot operations are usually specified in the task space indicating their location and pose, but the manipulator is naturally described in the joint space. The objective of inverse kinematics (IK) is to compute the joint space variables that must be commanded to the manipulator in order to accomplish the assigned task [Siciliano and Khatib, 2008]. In a first-order differential kinematics, the relationship between the task space and joint space velocities direct-kinematic is established by the task Jacobian, which plays a central role in redundancy resolution techniques.

The simplest solution to the redundancy problem is the local inversion of direct-kinematic based on the pseudoinverse of the Jacobian matrix [Whitney, 1969]. This solution corresponds to the minimization of the joint velocities in a least-squares sense. Since a redundant system has more degrees of freedom than required to accomplish a given task, there exist multiple solutions to the IK problem. Among all feasible solutions satisfying the task requirements, the local inversion of the Jacobian provides the joint velocity with the minimum norm [Siciliano and Khatib, 2008].

In case that other solution than the one with minimum norm joint velocity is preferred, augmented Jacobian techniques [Egeland, 1987] propose the addition of constraint tasks that restrict the region of suitable solutions to the IK problem. Task augmentation permits the fulfilment of additional goals at the expense of reducing redundancy. If any added constraint task does cause conflict with the main task might lead to the arising of algorithmic singularities, even if the original task Jacobian is full rank [Antonelli, 2014]. When the robot is very close to a singularity, the IK problem becomes ill-conditioned. In these scenarios, the local IK solution requires high joint velocities in order to obtain relative small task velocities. A widely used solution to deal with singularities is based on a damped pseudoinversion of the task Jacobian [Chiaverini et al., 1994]. The damping factor can be chosen according to different techniques, but all of them share a common drawback, large damping factors might produce large task errors.

Algorithmic singularities can also be handled in the framework of task-priority strategies, where tasks are ordered according to their priority [Maciejewski and Klein, 1985]. The lower-priority tasks are only satisfied in the null space of the higher-priority tasks. Thus, when the exact solution does not exist, the reconstruction error only affects the lower priority tasks. The singularity-robust IK solver method proposed in [Chiaverini, 1997] does not enforce the execution of singular secondary tasks and thus preventing primary tasks being affected by singularities in lower priority tasks.

The reverse priority (RP) method detailed in [Flacco and De Luca, 2014] is based on the idea of computing joint motion contributions starting from

the task of lowest priority and moving up to the primary task. A special projection matrix is derived, which ensures the preservation of the correct priority order. Not enforcing secondary tasks, as in singularity-robust IK method, preserves task hierarchy but tasks might not execute accurately even when it would be possible. The execution of higher priority tasks at the end avoids possible deformations caused by singularities occurring in lower priority tasks. The RP method allows executing at the best all tasks while still preserving the desired hierarchy [Flacco and De Luca, 2014].

Tasks are often described as a reference trajectory for a geometric component of a robot (the Cartesian pose of the end-effector, the position of a control point on the robot body, a joint angle, etc.). At each time instant, a desired value for this component is specified in the form of a bilateral (equality) constraint. However, sometimes they do not reflect properly the desired robot operation. In many applications it is not strictly necessary to assign a particular value to some given robot component, but it would be preferable to specify a region to which this component should belong. This requirement can be coded with unilateral (inequality) constraints. Unilateral constraints can be used to avoid obstacles in the workspace or take account for hardware limitations. The treatment to this last kind of task is more complex and an active research topic.

The current approaches for dealing with the problem of unilateral constraints in redundant robots can be divided in three groups [Flacco and De Luca, 2015]:

- The first group of approaches takes an optimization point of view. The unilateral constraints are transformed into linear inequality constraints at the control differential level. Quadratic programming is used to find an optimal solution. These approaches require a high computational cost, becoming useless for real-time robot applications when the number of constraints increases [Kanoun et al., 2011]. In [Simetti et al., 2014], a framework based on the solution of a sequence of linearly constrained least-squares problems was presented. And a hierarchical method was used in [Escande et al., 2014]. Both approaches offer a significant reduction of computational cost.

- The second group of approaches transform unilateral constraints on a robot component into bilateral constraints at the control differential level [Chaumette and Marchand, 2000], [Mansard et al., 2009], [Flacco and De Luca, 2015]. These tasks are only activated when a component is close to its range limit, but a smooth evolution of the joint commands during the activation phase has to be guaranteed. However, the results obtained by these approaches are typically suboptimal, in the sense that one unilateral constraint may be activated even if it

not strictly needed for accomplishing the desired task. On the other hand, the implementation of these approaches is in general simpler and less computationally demanding than approaches included in the first group.

- The approaches in the third group use the null space of the equality tasks for taking into account unilateral constraints. These methods are very common thanks to their simple implementation. But, they do not guarantee the accomplishment of unilateral tasks when they are in contrast with the fulfilment of the equality tasks. [Liegeois, 1977] and [Marey and Chaumette, 2010] present solutions based on a projected gradient formulation and [Chan and Dubey, 1995] on resorting weighted pseudoinverse.

3.3 Background

A robot can be kinematically defined by its kinematic chains. At the same time, a kinematic chain is defined by the joints and links it contains. Different kinematic chains of the same robot can share some of the joints and links. Thus, the number of degrees of freedom of a robot system is determined by the number of independent joints the robot has.

Otherwise, the last link of a kinematic chain is known as end-effector and most of the tasks a robot must perform consist in achieving a specific movement with these kind of links.

For a given robot with n degrees of freedom, let $q \in \mathbb{R}^n$ be the position of their joints and $\dot{q} \in \mathbb{R}^n$ their velocities. At a robot configuration, the differential kinematics of a generic task $x \in \mathbb{R}^m$ with dimension m, being $m \leq n$, is

$$\dot{x} = \frac{\partial x}{\partial q}\dot{q} = J\dot{q}, \tag{3.1}$$

where $J = J(q)$ is the $m \times n$ task Jacobian matrix.

The inverse solution of the differential kinematics mapping can be obtained using

$$\dot{q} = J^{\#}\dot{x}, \tag{3.2}$$

where $J^{\#}$ is the $n \times m$ Moore-Penrose pseudoinverse of the Jacobian matrix. The obtained $\dot{q}$ enjoys the minimum-norm property. However, the whole manifold of solutions that satisfy the task can be generated as

$$\dot{q} = J^{\#}\dot{x} + P\dot{q}_0, \tag{3.3}$$

where P denotes the $n \times n$ orthogonal projector in the Jacobian null space and $\dot{q}_0 \in \mathbb{R}^n$ an n-dimensional arbitrary joint velocity vector.

For highly redundant systems, it would be possible to introduce multiple tasks of different nature, and then decide an order of priority between them. The execution of tasks of lower priority should not interfere with the execution of task with higher priority. This hierarchy is obtained by projecting a task in the null space of all higher priority tasks. A standard method for solving the problem is using the recursive formula [Siciliano and Slotine, 1991]

$$\dot{q}_p = \dot{q}_{p-1} + (J_p P_{p-1})^{\#}(\dot{x}_p - J_k \dot{q}_{p-1}), \tag{3.4}$$

for $p = 1, ..., l$, being l the number of tasks. In order to initialize the method, $\dot{q}_0 = 0$ and $P_0 = I$. The matrix P_p is the projector in the null space of the augmented Jacobian of the first p tasks.

However, finding a joint velocity solution that satisfies the tasks is not enough. Robots are subject to a various sets of constraints which reduce the space of possible motions. Thus, these constraints need to be taken into account when calculating the best feasible solution.

Most common constraints are related to robotic joints i.e. joint limits, maximum joint velocity and maximum acceleration. However, in some cases, it is desirable to limit also the Cartesian movements of the end-effectors, usually because of safety reasons.

3.4 Framework Solution

In this section, we present a new approach for solving multi-task kinematic problems in redundant robots with one or more kinematic chains (see Algorithm 1). The approach is able to find the joint velocities that better fit with a list of hierarchical tasks avoiding to surpass a set of previously defined joint and Cartesian constraints. For each level of the task hierarchy, it is also possible to introduce a joint prioritization.

3.4.1 Inverse Kinematics

The Reverse Priority approach presented in [Flacco and De Luca, 2014] has been used in the proposed framework for solving the IK problems. As was said in the State of the Art, they propose an IK solver handling prioritized robotic tasks. This method guarantees the task execution in the correct priority order even when kinematic singularities appear. It is based on the idea of calculating first the joint velocities for the lowest priority task and then the contribution of higher priority task are added recursively.

Considering a set of l tasks

Algorithm 1: Multi-Task Priority Framework

$T_0 = [\ ], \dot{q}_{l+1} = 0, J^{RA}_{l+1} = 0$
$H = calculateHardConstraints(q)$
for $p = l \rightarrow 1$ **do**
  $joints_{p,0} = [\ ], \dot{q}_{p,0} = \dot{q}_{p+1}$
  for $j = 1 \rightarrow m_p$ **do**
    $active\_joints_{p,j} = joints_{p,j} \cup active\_joints_{p,j-1}$
    $J_{p,j} = adaptJacobian(J_p, active\_joints_{p,j})$
    $J^{RA}_{p,j} = \begin{pmatrix} J_{p,j} \\ \\ J^{RA}_{p+1} \end{pmatrix}$
    $T_{p,j} = getT_p(J^{RA\,\#}_{p,j}, size(J^{\#}_p))$
    $\dot{q}_{p,j} = \dot{q}_{p,j-1} + T_{p,j}(J_{p,j}T_{p,j})^{\#}(\dot{x}_p - J_p\dot{q}_{p,j-1})$
    $\dot{q}_{p,j} = hardConstraints(\dot{q}_{p,j}, J^{RA}_{p,j})$
    if $|(\dot{x}_p - J_p\dot{q}_{p,j})| - |(\dot{x}_p - J_p\dot{q}_{p,j-1})| < \epsilon$ **then**
      $\dot{q}_{p,j} = \dot{q}_{p,j-1}$
    end
  end
  $\dot{q}_p = \dot{q}_{p,m_p} + J^{\#}_p(\dot{x}_p - J_p\dot{q}_{p,m_p})$
  $\dot{q}_p = hardConstraints(\dot{q}_p, J_p)$
  if $|(\dot{x}_p - J_p\dot{q}_p)| - |(\dot{x}_p - J_p\dot{q}_{p,m_p})| < \epsilon$ **then**
    $\dot{q}_p = \dot{q}_{p,m_p}$
  end
end

$$\dot{x}_p = J_p \dot{q}_p \qquad p = 1, ..., l, \tag{3.5}$$

where $\dot{x}_p$ is the task with priority p and this priority is higher than $p+1$. The results previously mentioned are achieved thanks to the recursive formula

$$\dot{q}_p = \dot{q}_{p+1} + T_p (J_p T_p)^{\#} (\dot{x}_p - J_p \dot{q}_{p+1}), \tag{3.6}$$

where $\dot{q}_p$ is the solution of the task with priority p which takes into account all lower priority tasks. In order to obtain T_p, consider the reverse augmented Jacobian J_p^{RA}, bordered on top with J_p

$$J_p^{RA} = (J_p^T \; J_{p+1}^T \; ... \; J_l^T \;)^T = \begin{pmatrix} J_p \\ J_{p+1}^{RA} \end{pmatrix}, \tag{3.7}$$

then T_p is the $n \times m_p$ matrix bordered on left with the pseudoinverse of J_p^{RA}

$$J_p^{RA\,\#} = \begin{pmatrix} T_p & * \end{pmatrix}. \tag{3.8}$$

For further details, please refer to [Flacco and De Luca, 2014].

3.4.2 Multi-Chain Tasks and Jacobians

A property of the prosed framework is the possibility to execute tasks that affect different kinematic chains, even in the same priority level.

Considering a set of r tasks with the same priority p, the general task $\dot{x}_p$ that includes all of them can be obtained by stacking the individual tasks

$$\dot{x}_p = \begin{pmatrix} \dot{x}_{p,1} \\ \dots \\ \dot{x}_{p,r} \end{pmatrix}. \tag{3.9}$$

The task Jacobian J_p associated to the global task $\dot{x}_p$, is also obtained by stacking the Jacobians associated to each individual tasks. However, as was said before, not all the tasks must be related with the same kinematic chains, namely with the same joints. Thus, in order to construct J_p it is necessary to normalize the shape of the task Jacobians, and all of them must be extended to have n columns, being n the number of joints of the system. Then, the cth column of the extended Jacobian $J_{p,r}^E$, will be filled

by either the column of the Jacobian $J_{p,r}$ associated to the joint c, in case this joint it is associated to the Jacobian, or with zeros otherwise. Finally, J_p is calculated by

$$J_p = \begin{pmatrix} J^E_{p,1} \\ \dots \\ J^E_{p,r} \end{pmatrix}. \tag{3.10}$$

3.4.3 Joint Priority

Apart from prioritizing the tasks, sometimes it is also desirable to define with which joints it is preferable to solve a determined task. For instance, when an I-AUV is performing a manipulation task, it could be convenient to execute the task, if it is possible, using only the manipulator. This is due to manipulators usually perform more accurate movements than vehicles. Thus, this way of proceeding avoids unnecessary risks.

The presented framework allows the users to define a joint hierarchy for each task level. Then, the tasks are tried to be solved using only the joints with higher priority. If it is not possible, next hierarchy group is also used to find a solution, and so on.

Considering that for solving the task $\dot{x}_p$, there are m_p groups of joints. For the jth group of joints, the active joints ($active\_joints_{p,j}$) are all those with priority higher or equal to j,

$$\begin{aligned} active\_joints_{p,j} &= [joints_{p,j} \cup ... \cup joints_{p,1}] \\ &= [joints_{p,j} \cup active\_joints_{p,j-1}]. \end{aligned} \tag{3.11}$$

Then, the Jacobian associated to the task J_p needs to be adapted to use only these active joints. To do so, a $n \times n$ identity matrix A_j is created and the column k, with $k \leq n$, is set to zero in case the joint k is not active, it means $k \notin active\_joints_{p,j}$. The adapted Jacobian $J_{p,j}$ is obtained by

$$J_{p,j} = J_p A_j \tag{3.12}$$

This adapted Jacobian is used to execute the Reverse Priority method and finding a solution for the remaining task error.

3.4.4 Hard Constraints

So far, the IK solver used, does not take into account the system constraints. Thus, after finding the joint velocities that solve a task, it is needed to

check that these velocities do not produce any constraint violation. In case it happens, the velocities must be modified in order to satisfy these constraints and continue accomplishing the tasks hierarchy.

The presented framework allows the users to limit joints movement, as well as end-effector Cartesian movements.

The robot motion capabilities are defined by a maximum joint velocity ($|\dot{q}_i| \leq \dot{Q}_{max}$), a joint range for each joint ($Q_{min,i} \leq q_i \leq Q_{max,i}$), a maximum Cartesian velocity ($|\dot{x}_i^k| \leq \dot{X}_{max,i}^k$) and a Cartesian range for each translational and rotational axis (x, y, z, roll, pitch, yaw) of each end-effector ($X_{min,j}^k \leq x_j^k \leq X_{max,j}^k$), where i is one of the joints of the system, k one of the kinematic chains and j one of the axes.

In the control implementation the joint velocities $\dot{q}$, which are the output of one framework execution, are sent to the system and kept constant during a sampling time of duration T. Taking this into account, the maximum velocities allowed must not produce any limit overrun during the sample time. Moreover, if one constraint are already exceeded, no movement that worsen this must be allowed.

In order to accomplish these two premisses, a maximum allowed positive and negative velocity for each joint and each axis of each end-effector are calculated before starting the task solver algorithm.

At a current joint configuration q, the velocity of the joint i must satisfy that

$$\dot{q}_{min,i} = \begin{cases} 0 & \text{if } q_i \leq Q_{min,i} \\ max\left\{\frac{Q_{min,i}-q_i}{T}, -\dot{Q}_{max}\right\} & \text{if } q_i > Q_{min,i} \end{cases} \tag{3.13}$$

$$\dot{q}_{max,i} = \begin{cases} 0 & \text{if } q_i \geq Q_{max,i} \\ min\left\{\frac{Q_{max,i}-q_i}{T}, \dot{Q}_{max}\right\} & \text{if } q_i < Q_{max,i} \end{cases} \tag{3.14}$$

$$\dot{q}_{i,min} \leq \dot{q}_i \leq \dot{q}_{i,max} \tag{3.15}$$

On the other hand, the position and velocity of the end-effector of a kinematic chain k, having a joint configuration q, can be calculated by $x^k = FK(q)$ and $\dot{x}^k = J^k\dot{q}$ respectively, where FK is the forward kinematics of the chain and J^k its Jacobian matrix at the system configuration q. Taking this into account, the Cartesian velocity for an axis j must accomplish that

$$\dot{x}_{min,j}^k = \begin{cases} 0 & \text{if } x_j^k \leq X_{min,j}^k \\ max\left\{\frac{X_{min,j}^k - x_j^k}{T}, -\dot{X}_{max}\right\} & \text{if } x_i^k > X_{min,j}^k \end{cases} \tag{3.16}$$

$$\dot{x}^k_{max,j} = \begin{cases} 0 & \text{if } x^k_i \geq X^k_{max,j} \\ min\left\{\frac{X^k_{max,j} - x^k_j}{T}, \dot{X}_{max}\right\} & \text{if } x^k_i < X^k_{max,j} \end{cases} \tag{3.17}$$

$$\dot{x}^k_{j,min} \leq \dot{x}^k_j \leq \dot{x}^k_{j,max} \tag{3.18}$$

Once the maximum velocities are calculated, each time the solver returns velocities the algorithm 2 is executed. This algorithm checks if these velocities produce any limit exceedance and modify them in case it is needed. The modification also tries to respect the tasks hierarchy.

Let be x^J and x^C two virtual tasks in charge of putting the joint and Cartesian exceeded velocities, respectively, in the allowed range. Let J^J and J^C be their task Jacobian matrices. Then, next steps are followed:

1. Check if there are one or more Cartesian velocities violating a limit. Let suppose that the kinematic chain k is exceeding the velocity limit in the axis j. Then, the error is calculated and added to the task $\dot{x}^C$ in order to force the velocity to be into the limits. Besides, the jth row of the Jacobian associated to the kinematic chain k, J^k_j, is added to J^C.

2. If in previous iterations a Cartesian velocity was not inside the range, which means that there is a row in $\dot{x}^C$ and J^C associated to the axis, but it is now accomplishing the limits, the associated task is set to 0, $\dot{x}^{C,k}_j = 0$. This is used to avoid that the velocity of this axis changes.

3. If all the Cartesian velocities are inside the limits, the joint velocity limits are checked. Otherwise, $\dot{x}^J$ and J^J are reset and the joint velocities are not checked in this iteration. This step tries to avoid first Cartesian overruns, in which usually are implied more than one joint, and after that, correct the joint velocities.

4. If the Cartesian limits are all into the limits, in order to check the joint velocities, steps 1 and 2 are executed but the task and Jacobian filled are $\dot{x}^J$ and J^J. In this case, the Jacobian row associated to a joint i is a row vector filled with 0s and with a 1 in the ith position.

5. Next step consists on joining both tasks $\dot{x}^{C,J} = \begin{pmatrix} \dot{x}^C \\ \\ \dot{x}^J \end{pmatrix}$ and Jacobians

$$J^{C,J} = \begin{pmatrix} J^C \\ \\ J^J \end{pmatrix}.$$

6. Finally the Reverse Priority method is used to find the velocities that better accomplishes the task $x^{C,J}$, having as a Jacobian $J^{C,J}$. It is also desired to take into account the previously solved tasks. Thus, the matrix $T^{C,J}$ is the projector in the null space of the augmented Jacobian of the already solved tasks.

7. If the new velocities do not break any limit, they are returned. Otherwise, a new algorithm iteration is executed.

Sometimes, the constraints may cause the non-fulfilment of the current task. Thus, if after applying the constraints control algorithm, the resultant velocities do not generate an improvement in the current task fulfilment higher than a certain value ϵ comparing with the velocities coming from the last iteration, these new velocities are discarded and the old ones are used in the next iteration.

3.4.5 Ensure Task Fulfilment

We found that RP method successfully maintains the task hierarchy, but the update equation in RP only distinguishes the current task and all tasks with less priority. It means, during the calculation of a given task, all tasks with less priority are placed at the same priority level. This leads to find a correct set of joint velocities as long as the IK problem has a valid solution that satisfies all the constraints. But, when the fulfilment of the hard constraints was forced to be satisfied, all the tasks in the hierarchy were modified without considering their priority. In order to mitigate such effect, an additional calculation of the joint velocities, only considering the Jacobian of the current task, is done at the end of each task calculation. If the joint velocities were not modified by the hard constraints, this step has no influence because the solution of the RP method is also valid for the current task. In case the joint velocities were modified by the hard constraints, this step permits the current task having a major influence on the solution than the previous ones.

Let's suppose a system with three joints $n = 3$ and two tasks in the hierarchy $\dot{x}_1 = (1)$ and $\dot{x}_2 = (\text{-}1)$ whose Jacobian matrices are $J_1 = (1\ 1\ \text{-}1)$ and $J_2 = (1\ 0\ 0)$. Let's also suppose that the 2nd joint is in the upper bound and the 3rd one in the lower bound, which means that the final velocities must accomplish that $\dot{q}_{max,2} = 0$ and $\dot{q}_{min,3} = 0$.

Algorithm 2: Hard Constraints

function $hardConstraints(\dot{q}, J^{RA})$
$J^C = [\ ], J^J = [\ ], x^C = [\ ], x^J = [\ ]$
$cartesianLimit,\ jointLimit = True$
while $cartesianLimit \parallel jointLimit$ **do**
  $cartesianLimit = False$
  $jointLimit = False$
  for $k = 1 \rightarrow kinematic\_chains$ **do**
    for $j = 1 \rightarrow axes$ **do**
      if $not(\dot{x}^k_{j,min} \leq \dot{x}^k_j \leq \dot{x}^k_{j,max})$ **then**
        $J^J = [\ ], x^J = [\ ]$
        $J^{C,k}_j = J^k_j$
        $error = \begin{cases} \dot{x}^k_{j,min} - \dot{x}^k_j & \text{if } \dot{x}^k_j < \dot{x}^k_{j,min} \\ \dot{x}^k_{j,max} - \dot{x}^k_j & \text{if } \dot{x}^k_j > \dot{x}^k_{j,max} \end{cases}$
        $\dot{x}^{C,k}_j = error$
        $cartesianLimit = True$
      end
      else if $\exists \dot{x}^{C,k}_j$ **then**
        $\dot{x}^{C,k}_j = 0$
      end
    end
  end
  if $not(cartesianLimit)$ **then**
    for $i = 1 \rightarrow joints$ **do**
      if $not(\dot{q}_{i,min} \leq \dot{q}_i \leq \dot{q}_{i,max})$ **then**
        $J^J_i = I_i$
        $error = \begin{cases} \dot{q}_{i,min} - \dot{q}_i & \text{if } \dot{q}_i < \dot{q}_{i,min} \\ \dot{q}_{i,max} - \dot{q}_i & \text{if } \dot{q}_i > \dot{q}_{i,max} \end{cases}$
        $\dot{x}^J_i = error$
        $jointLimit = True$
      end
      else if $\exists \dot{x}^J_i$ **then**
        $\dot{x}^J_i = 0$
      end
    end
  end
  $J^{C,J} = \begin{pmatrix} J^C \\ J^J \end{pmatrix}$
  $\dot{x}^{C,J} = \begin{pmatrix} \dot{x}^C \\ \dot{x}^J \end{pmatrix}$
  $J^{RA,C,J} = \begin{pmatrix} J^{C,J} \\ J^{RA} \end{pmatrix}$
  $T^{C,J} = getT(J^{RA,C,J}, size(J^{C,J\,\#}))$
  $\dot{q} = \dot{q} + T^{C,J}(J^{C,J}T^{C,J})^{\#}(\dot{x}^{C,J})$
end
return $\dot{q}$

The velocities obtained after solving the task with lower priority x_2,

$$\dot{q}_2 = 0 + T_2(J_2T_2)^{\#}(\dot{x}_2) = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}, \tag{3.19}$$

do not exceed any limit.

Regarding the task with higher priority x_1, the solution proposed by the IK solver is,

$$\dot{q}_1 = \dot{q}_2 + T_1(J_1T_1)^{\#}(\dot{x}_1 - J_1\dot{q}_2) = \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}. \tag{3.20}$$

In this case, both the velocities of the joints 2 and 3 are not inside the allowed range. So, they must be modified using the Hard Constraints algorithm. The virtual task and the Jacobian needed for solving the problem are

$$x^J = \begin{pmatrix} -1 & 1 \end{pmatrix}^T, \qquad J^J = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3.21}$$

So, the velocities obtained after the algorithm would be

$$\dot{q}_{1,HC} = \dot{q}_1 + T^J(J^JT^J)^{\#}(\dot{x}^J) = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}. \tag{3.22}$$

It is easy to prove that the new velocities do not accomplish the task with higher priority, so the methodology is not working properly in this case. In order to solve these cases, before leaving a task the methodology assures that it is satisfied as much as possible. To do so, the velocities that solve the remaining error are calculated without taking into account the null space of the previous tasks. And, in case the obtained joint velocities violate a hard

constraint, the Hard Constraint algorithm is called but now using only the null space of the currently solved task.

Continuing with the proposed example, after avoiding the constraint exceedances, the IK is used to find a solution without taking into account the task with lower priority,

$$\dot{q}_1 = \dot{q}_{1,HC} + J_1^{\#}(\dot{x}_1 - J_1\dot{q}_{1,HC}) = \begin{pmatrix} -0.3333 \\ 0.6667 \\ -0.667 \end{pmatrix}. \tag{3.23}$$

Once again, the velocities for the joint 2 and 3 are breaking the limits. So, the Hard Constraints algorithm is executed but having as reverse augmented Jacobian the task Jacobian of the current task

$$J^{RA} = \begin{pmatrix} 1 & 1 & -1 \end{pmatrix}, \tag{3.24}$$

and as a virtual task and its Jacobian,

$$x^J = \begin{pmatrix} -0.6667 & 0.6667 \end{pmatrix}^T, \quad J^J = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3.25}$$

Executing the algorithm the velocities returned are

$$\dot{q}_{1,HC} = \dot{q}_1 + T^J(J^J T^J)^{\#}(\dot{x}^J) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \tag{3.26}$$

This new velocities completely fulfil with the task with higher priority and do not surpass any limit.

3.5 Experimental Results

Two simulated experiments have been performed to evidence the functionalities of the framework. Both experiments were conducted with the UWSim simulator [Prats et al., 2012d].

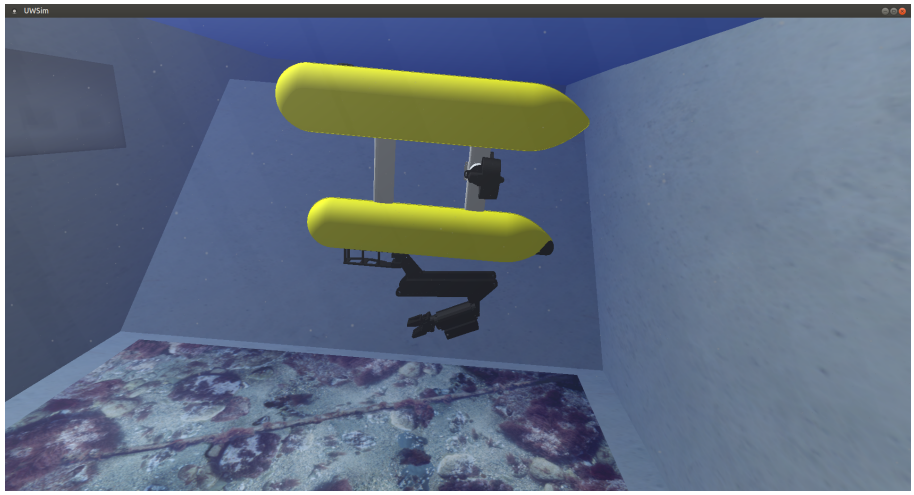


Figure 3.1: UWSim 3D visualization of the Girona500 AUV with the ARM 5E manipulator attached.

In the two experiments the robotic system used is a simulated model of the Girona 500 AUV [Ribas et al., 2012], with the robotic manipulator ECA-CSIP Light-weight ARM 5E [Fernández et al., 2013] attached (see Fig. 3.1). The vehicle has been configured with 4 degrees of freedom (x,y,z and yaw), and the arm has 4 more (slew, shoulder, elbow and jawRotate). Two kinematic chains have been defined. The first one uses the 4 degrees of freedom of the vehicle and its end-effector is in the center of the vehicle. The second one uses the 8 degrees of freedom, and its end-effector is in the gripper of the arm.

Concerning the constraints, additionally to the joint limits indicated in the Table 3.1, the maximum joint velocity was set to $0.1 rad/s$ and the maximum Cartesian velocity to $0.2 m/s$.

Table 3.1: Joint Limits of the Girona 500 and the ARM5E. (angular units in radians)

Joints	x	y	z	yaw	slew	shoulder	elbow	jawRotate
max_limit	∞	∞	∞	∞	0.4	1.37	1.45	∞
min_limit	$-\infty$	$-\infty$	$-\infty$	$-\infty$	-1.0	0.1	0.1	$-\infty$

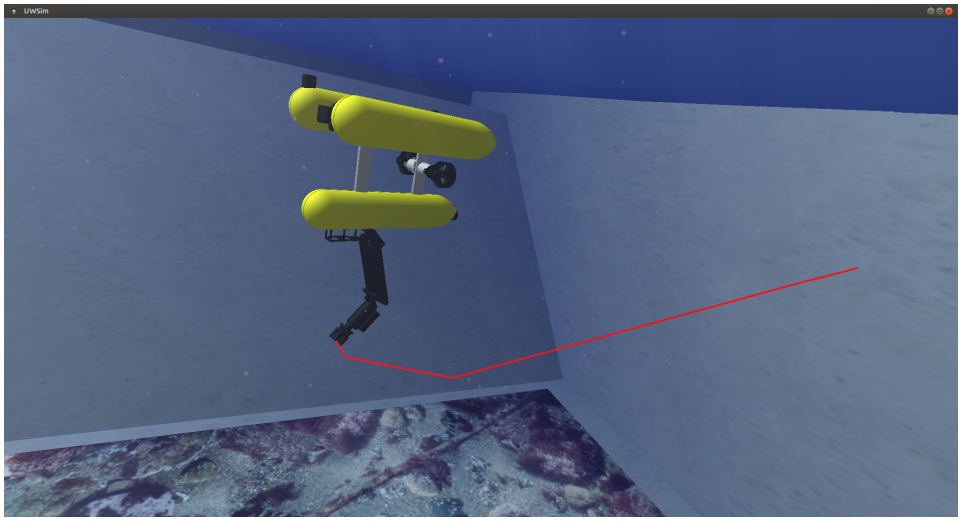


Figure 3.2: UWSim 3D visualization of the gripper trajectory during the first experiment.

3.5.1 First Experiments

The idea of the first experiment is to demonstrate how the framework is able to deal with multiple tasks at the same time that Cartesian limits and joint priorities are defined.

Two tasks are defined for this experiment. The task with the lowest priority, tries to move the component z of the vehicle to the position $z = -10\ m$. And the goal of the highest priority task is to reach the position $(x = 4.0, y = 2.0, z = 4.0)\ m$ with the gripper (see Fig. 3.2). At the same time, in the task with the highest priority, the joints of the vehicle have less priority than the ones of the arm.

Apart from the joint limits, two Cartesian limits have been defined. The component x of the vehicle cannot be higher than $3.0\ m$ and the component z of the gripper is also positively limited to $3.0\ m$.

Fig. 3.3 demonstrates how the vehicle tries to move in $-z$ direction as long as the joints of the arm can move the gripper in the opposite direction. When the arm alone is not able to fulfil the highest priority task, the vehicle must help to accomplish it, ignoring the task with lower priority.

It is also worth noting, how the two end-effectors never exceed their Cartesian limits, even when it provokes that the tasks are not going to be fulfilled (see Fig. 3.3 and Fig. 3.4). At the end of the experiment, the component y of the end-effector, which has not been limited, is the only one able to reach its goal value.

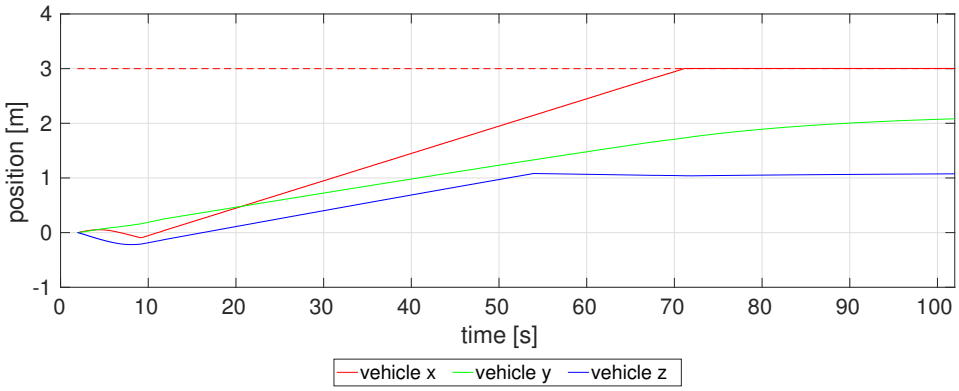


Figure 3.3: Trajectory of the x (red), y (green) and z (blue) components of the vehicle during the first experiment. The Cartesian limit ($x \leq 3$) on the movement of the vehicle is represented with a dashed line.

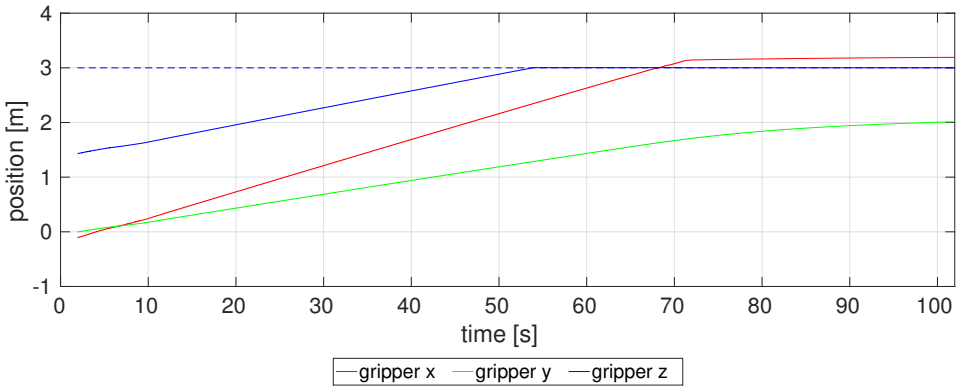


Figure 3.4: Trajectory of the x (red), y (green) and z (blue) components of the gripper during the first experiment. The Cartesian limit ($z \leq 3$) on the movement of the gripper is represented with a dashed line.

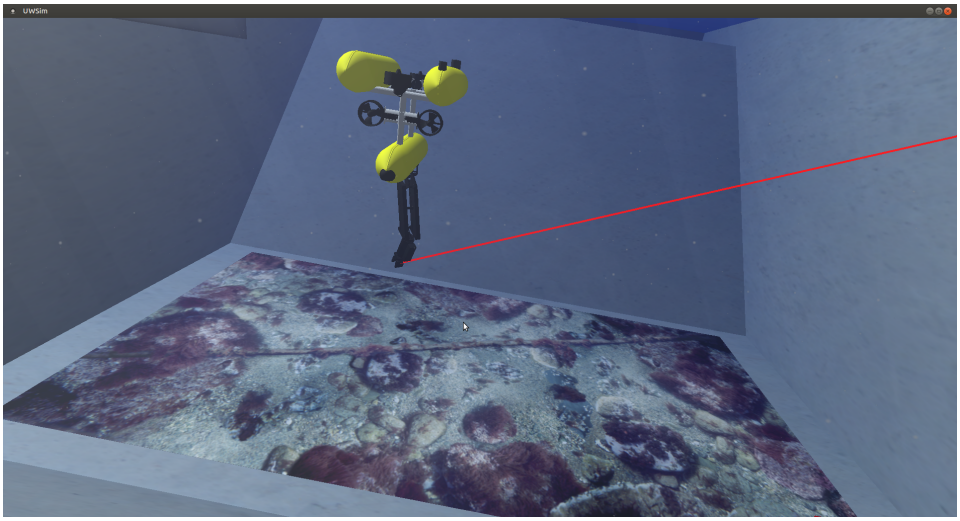


Figure 3.5: UWSim 3D visualization of the gripper trajectory during the second experiment.

3.5.2 Second Experiment

The second experiment tries to show the framework ability to deal with more than one task in the same priority level even when the task do not become at the same kinematic chain. At the same time, the framework controls that no joint exceeds its limits.

This experiment is composed by one task in the highest level of priority and two tasks with the same priority in the second level. The highest priority task tries to move the gripper to the position $(x = 4.0, y = 2.0, z = 4.0)$ m. In the second level, one task is in charge of positioning the yaw component of the vehicle in the value 1.5 rad. And the other one tries to keep the slew, shoulder and elbow joints as close as possible to the values -0.7, 2.0 and 0.0 rad, respectively (see Fig. 3.5).

Fig. 3.6 shows how the gripper reaches the expected position, and the vehicle the correct orientation.

The movement of the joints of the arm can be seen in the Fig. 3.7. They try to reach the values defined by the task but if the commanded joint value is out of range, the joint stops at the limit. It is also worth noting that, if one joint is out of bounds, the *elbow* in this experiment, it can just do movements in the direction that do not worsen this exceeding, until it is inside the limits, when it recovers a normal behaviour.

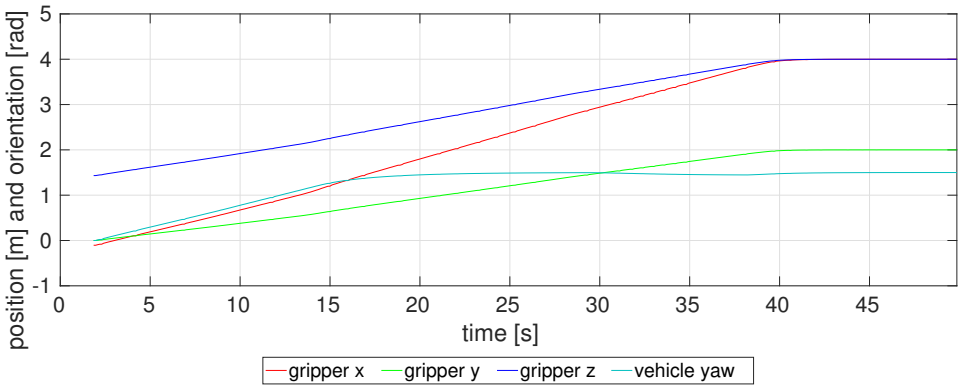


Figure 3.6: Trajectory of the x (red), y (green) and z (blue) components of the gripper, and the vehicle *yaw* (cyan) component during the second experiment.

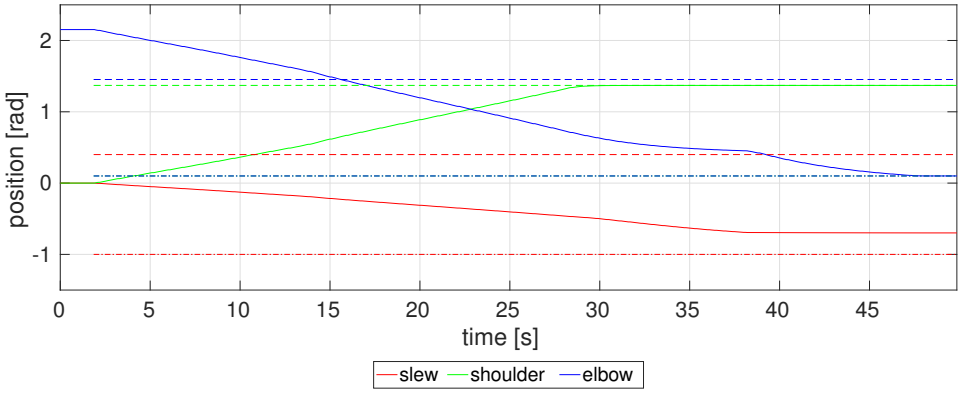


Figure 3.7: Arm joint values of the *slew* (red), *shoulder* (green) and *elbow* (blue) joints during the second experiment. The joint limits of the *slew*, *shoulder* and *elbow* joints are shown in dashed lines.

3.6 Conclusions

In this chapter, a kinematic control framework for redundant robots has been introduced. It permits the management of multiple tasks, multiple kinematic chains, joint priorities and hard constraints in a unified framework.

The Reverse Priority method has been considered to solve multiple tasks following a task priority strategy. It has been extended in order to introduce the possibility of assigning different priorities to each of the joints at each task level.The treatment of hard constraints has also been integrated within the proposed framework.

Two simulated experiments shown in this work have proven the validity of the proposed framework. In the first experiment is appreciated how the system tries to satisfy the highest priority task solely with the joints of the arm. When required, also the vehicle contributes to the fulfilment of the highest priority task. In both experiments the performance of the implementation of the hard constraints is demonstrated.

Chapter 4

Grasp Planning

This chapter presents a software framework which estimates the best feasible pose to grasp an object of interest taking as input the partial reconstruction of the scene where the object is located (see Fig. 4.1).

Firstly, the point cloud is properly segmented to separate the different objects of the scene. Then the dimensions of these objects are estimated. The object that better fit with a predefined criteria is chosen as object of interest. Finally, a grasp planner based on the Multi-Task Priority algorithm, explained in the last chapter, is used to obtain the best feasible grasping pose for this object, taking into account a series of goals and constraints.

The algorithm for separating the object from the rest of the scene has been described in an international conference [Peñalver et al., 2017].

4.1 Motivation

Once the I-AUV has reconstructed the scene, the manipulation requires to be planned.

The object position, orientation, shape or size are very important to properly planning a grasping. But, the point cloud generated, even if it has been obtained from different points of view, cannot provide all this information. Thus, the estimation of these hidden details is crucial for obtaining a good grasp plan.

Additionally, the system should be also taken into account during the grasp planning. On the one hand, the kinematic constraints limit the number of valid grasping poses. And on the other hand, some of these poses are preferable than others.

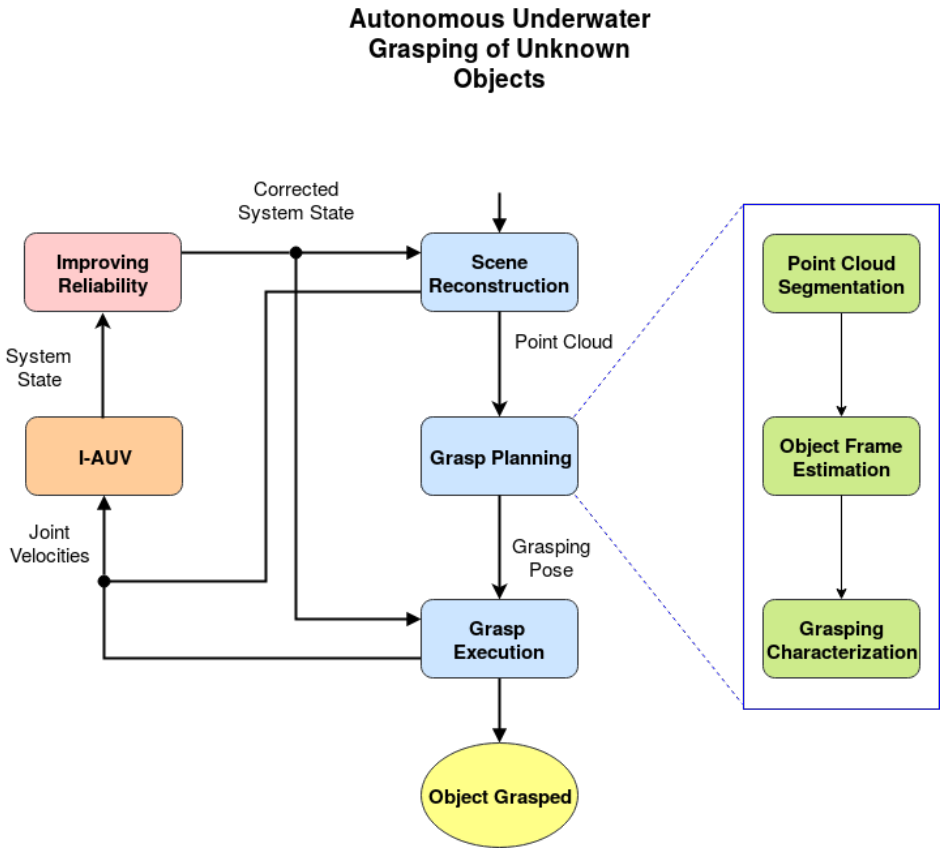


Figure 4.1: Flowchart of an autonomous underwater grasping intervention. The second of the phases of the intervention consists on planning the grasp. This phase is composed by three subphases (green). First, the point cloud received from the first phase is segmented to separate the object of interest from the rest, then the object is studied to obtain an appropriated object frame and finally the best feasible grasping pose is calculated.

4.2 State of the Art

The problem of finding a suitable grasp pose among the infinite set of candidates is a challenging problem which has been strongly studied during last years. In [Bohg et al., 2014], they divide the solutions for this problem depending on what the system knows a priori about the object of interest.

- *Known Objects*: In these approaches, the systems have access to a database containing geometric object models associated with a number of good grasps. In most cases, this database has been build offline. Once the object has been recognized, the goal is to estimate its pose and retrieve a suitable grasp.

- *Familiar Objects*: These approaches assumes that similar objects can be grasped in a similar way. Thus, they estimate the similarity between the new object with the previously studied ones. And then calculate which is the best grasp for this new object.

- *Unknown Objects*: Approaches in this group do not use previous information about objects or grasps. They focus on determining the features of the object using sensory data. These features are used to generate grasp candidates and ranking them.

In this thesis, it is assumed that the object of interest is unknown. For that reason, the state of the art focuses on the last group of methodologies.

Unknown object grasping approaches can also be categorized depending of the way the incomplete and noisy data received from the real sensors is treated [Lei et al., 2017].

4.2.1 Approximating Unknown Object Shape

These approaches use partial information of an object to estimate its complete shape and then determine the best grasp.

In [Bone et al., 2008], 2D images and structured light data from multiple views are being used to create a 3D model of the unknown object. From the 2D images, the silhouette are extracted to create a 3D visual hull, then it is merged with the more accurate 3D shape data retrieved from the structured light technique. Once the model is estimated, they search for a pair of reasonably flat and parallel surfaces that are best suited for their parallel gripper.

[Dune et al., 2008] use multi-view measurements to determine the quadratic that best resemble the shape of the object. The minor axis completed object is used to define the wrist orientation and the centroid serves as approach target.

An approach for grasping unknown objects with a multifingered hand is presented in [Lippiello et al., 2013]. They place a virtual elastic surface around the point cloud of the object, then this surface is shrunk at every iteration step (new image acquisition) until this intercepts with some points of the object. A pregrasp shape is defined in which the fingertip contacts on the virtual surface are aligned with its two minor axes. This grasp is then refined, given the local surface shape close to the contact point.

[Bohg et al., 2011] overcomes the problem of completing the object by considering planar symmetries found in human-made objects. Their algorithm first tries to determine the planar symmetry on which the detected point cloud of the object will be mirrored about. Two simple methods to generate grasp candidate on the resulting completed object models are proposed and evaluated.

A different way of completing the model of the object is to look at the available point cloud and reconstruct a fitting surface of the object using those points. In the work of [Lee et al., 2003], a 3D model is retrieved by using stereo matching. Then, a three-dimensional interpolation is applied using the triangular mesh method. The grasp candidate is obtained by using genetic techniques to solve an optimization problem.

4.2.2 From Low-Level Features to Grasp Hypotheses

In this case, the approaches have a predefined set of grasps postures. Thus, when the partial information of the object is acquired, they search for 2D or 3D visual features that fit with any of the predefined postures. Finally, the pairs features-postures are ranked following a set criteria.

In [Baumgartl and Henrich, 2012], Hough transformation is used to find edges in 2D images. Edges shorter than a minimum length are discarded. Parallel edges are sought in the remained set. If the distance between them is smaller than the gripper aperture, they are considered good contact points for grasping.

[Hsiao et al., 2010] place a bounding box around the available point cloud of an unknown object. Several heuristics are used to generate grasp hypotheses taking as input the bounding box. These can be grasps from the top, from the side, or applied to high points of the object. Finally, the hypotheses are ranked using a weighted list of features.

The proposed grasping approach in [Ala et al., 2015] uses blob detection to retrieve the boundaries of the point cloud of the object. These boundary lines are then transformed into straight lines. The grasp planner tries to find parallel contact points in order to execute an envelope grasp. When an object has a desirable thickness, one contact point can be retrieved in order to execute a boundary grasp.

The approach proposed by [Klingbeil et al., 2011] consists on searching for a pattern in the scene similar to the 2D cross section of the robotic gripper interior. A depth image of the scene is sampled to find a set of grasp hypotheses based on this pattern. Then the hypotheses are ranked according to an objective function that takes into account the grasp hypothesis and its local structure.

In [Lenz et al., 2015] a deep learning approach is used for detecting a good grasp from a RGB-D view of the scene. A small deep network is used to search potential grasps represented using oriented rectangles. A larger deep network is then used to rank the candidates based on the features extracted from the RGB-D image region contained inside their corresponding rectangles.

4.2.3 From Global-Shape to Grasp Hypotheses

The approaches included in this last group infer a good grasp hypothesis using the global shape of the object.

In [Richtsfeld and Vincze, 2008], a 3D mesh generation algorithm is applied on the segmented point cloud. Then, the surface is extracted using 2D DeLauney triangulation. One grasp point is found by calculating the edge point with minimum distance with respect to the center of masses. For obtaining the second grasping point, a virtual line that crosses the first grasping point and the center of masses is drawn. The intersections between this virtual lines and the edges of the object are the two grasping points.

The authors of [Maldonado et al., 2010] model the object as a 3D Gaussian distribution around the estimated object center. The grasp configuration is chosen by optimizing a criterion in which the distance between the center of the palm and object is minimized and the distance between the fingers and the object is maximized.

Only a single depth image is used in [Suzuki and Oka, 2016] to grasp unknown objects on a planar surface. They estimate the principal axis and centroid of the object. Then, the gripper approaches the object in a perpendicular direction to the plane, and grasps it in an orientation determined by the normal of the plane and the obtained principal axis.

[Stückler et al., 2011] generate grasp hypotheses based on eigenvectors of the object's footprints on a plane surface. Footprints refer to the 3D object point cloud projected onto the supporting surface. Then, the hypotheses are ranked depending on the distance to the object center, grasp width, grasp orientation and distance from the robot.

In [Calli et al., 2011], the grasping algorithm uses curvature information of the silhouette of an unknown object. Using elliptic Fourier descriptors, the silhouette of the object can be modelled from a 2D image. To find grasping

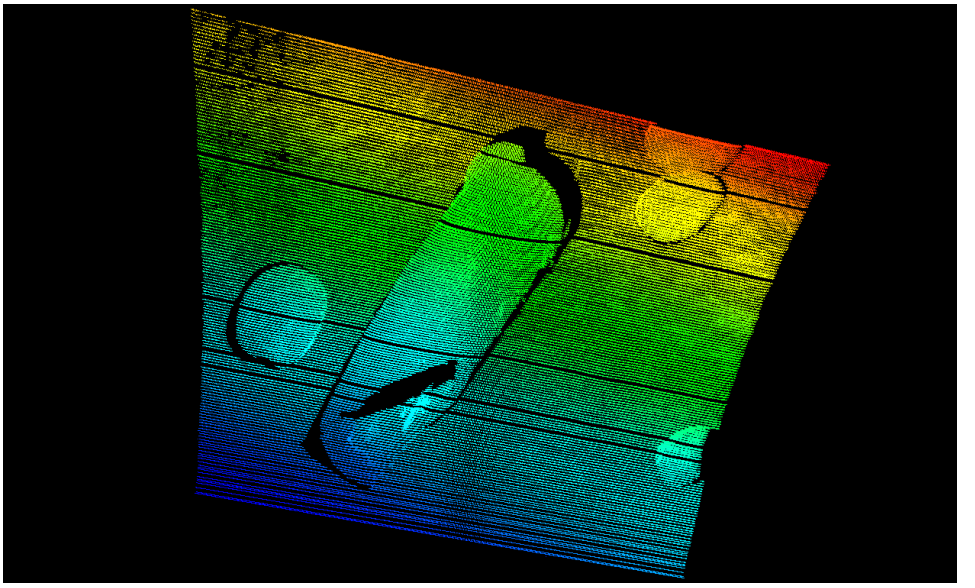


Figure 4.2: Underwater scene reconstructed using the Multi-View Laser Reconstruction algorithm

points, local minima and maxima curves of the silhouette are evaluated.

[Speth et al., 2008] uses images of the target object from different point of views to recover critical 3D information like the size, location and the pose. This information is used to plan a grasping using an extension of the 2D contour-based algorithms.

4.3 Grasp Planner for Unknown Objects

The second phase of the autonomous grasping starts when the scene, where the target is located, has been reconstructed. At that moment, the I-AUV has only a global knowledge of the scene (see Fig. 4.2). But using this information, it should be able to detect the different objects placed in the scene and choose the one that better fits with a set of conditions defined by the user previous to the starting of the intervention. After that, the grasp planner must determine which is the best pose for grasping this object taking into account the system characteristics.

Two assumptions have been made for calculating the grasping pose. First, the scene is composed by a surface more or less plane with one or more objects placed on it. And second, the use of a parallel jaw gripper.

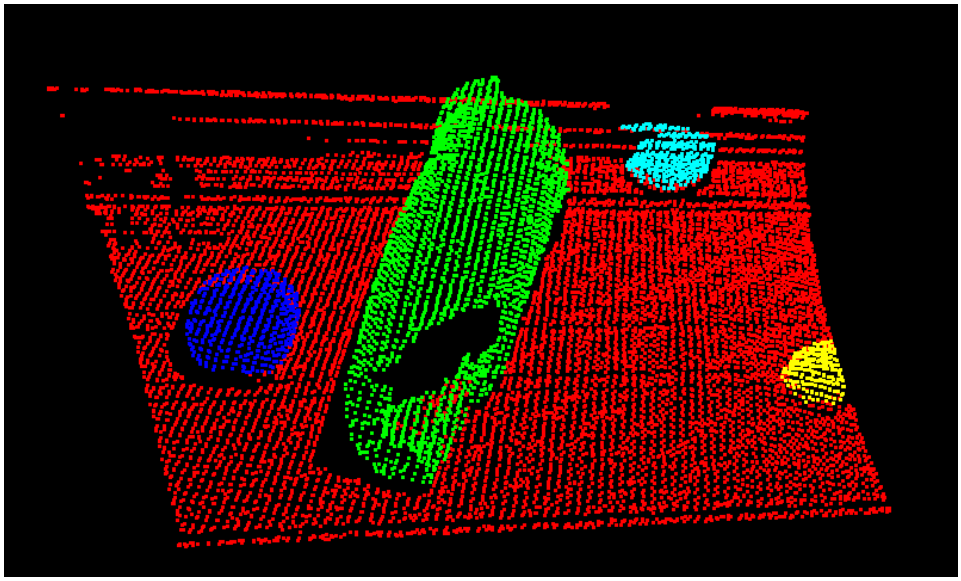


Figure 4.3: Point Cloud segmented. In red the floor, in green the object of interest and in blue, cyan and yellow other objects placed in the scene.

4.3.1 Object Detection

In the first phase of the algorithm, the point cloud is studied in order to recognise where the object of interest is placed and which points belong to it.

First of all, the plane where the objects are resting is estimated. To do so, the RANSAC segmentation algorithm [Fischler and Bolles, 1987] is applied to the scene reconstruction. This iterative algorithm returns the parameters of the plane that better fit with a set of points.

All the points closer to a certain distance d to the plane are considered part of it, and then, removed (see Fig. 4.3). Being a point $P(x_0, y_0, z_0)$ and the plane $\pi \equiv Ax + By + Cz + D = 0$, the minimum distance between them is defined by:

$$d(P, \pi) = \frac{\mid Ax_0 + By_0 + Cz_0 + D \mid}{\sqrt{A^2 + B^2 + C^2}} \tag{4.1}$$

After removing these points, a clustering algorithm (Euclidean Cluster Extraction [PCL-Segmenation,]) is used to group the remaining ones. Obtaining the set of points that partially define the shape of each of the objects in the scene (see Fig. 4.3).

Next step consists on estimating the size of the different objects in order to choose the one that better fits with some predefined conditions. As we suppose that all the objects are resting on the floor, the points of each

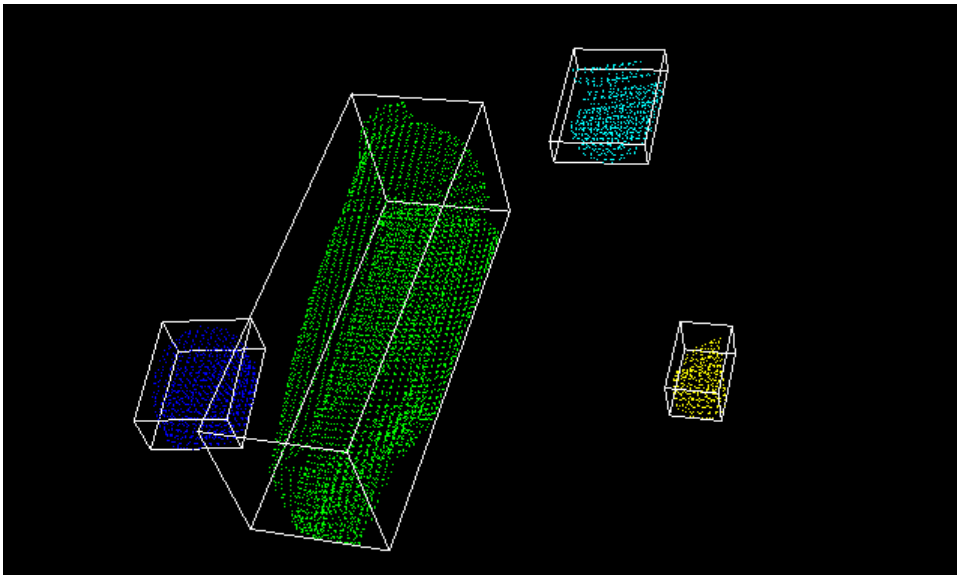


Figure 4.4: Bounding boxes of the objects presented in the scene.

object are orthogonally projected over the plane. Let be $P(x_0, y_0, z_0)$ one point belonging to one of the objects and $\pi \equiv Ax + By + Cz + D = 0$ the plane, $l = (x_0, y_0, < z_0) + t(A, B, C)$ is the line perpendicular to the plane π that crosses the point P. Thus, the intersection point between this line l and the plane π, is the orthogonal projection of the point P over the plane. This intersection can be obtained as:

$$A(x_0 + tA) + B(y_0 + tB) + C(z_0 + tC) + D = 0 \tag{4.2}$$

$$t = \frac{Ax_0 + By_0 + Cz_0 + D}{A^2 + B^2 + C^2} \tag{4.3}$$

$$P\_projected = (x_0 + tA, y_0 + tB, z_0 + tC) \tag{4.4}$$

The estimated size of the different objects is then calculated by obtaining the minimum bounding box, parallel to the plane, that encloses all the object points together with their projections (see Fig. 4.4).

As said before, the user defines some object size limits before the intervention. These limits are defined taking into account the size of the gripper, the kind of object that the user is interested in recovering, etc. Thus, the bounding boxes that are not inside the limits are discarded. The remaining objects are ranked according to a criterion also predefined by the user. A naive example of criterion could be to grasp the object with the highest volume that accomplishes the size limits. The best ranked object is selected.

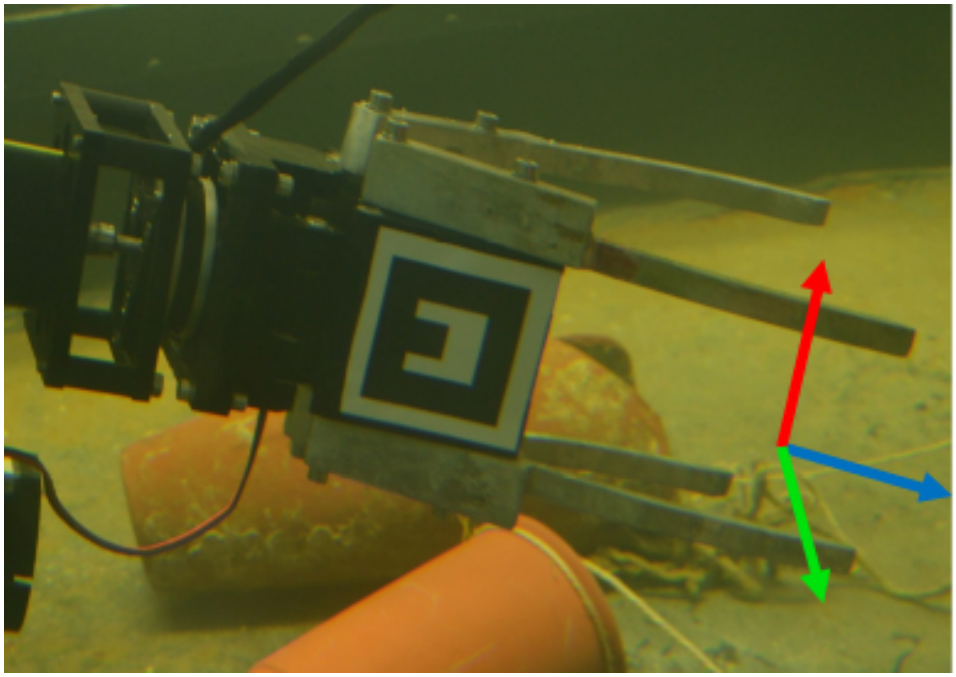


Figure 4.5: Frame of the UJIOne gripper.

4.3.2 Object Frame Estimation

Once the object of interest is separated from the rest of the scene, an object frame is estimated. This frame will indicate to the grasp planner where the object is located, which orientation it has and how the gripper should be positioned to perform a good grasp.

It is worth remembering that the point cloud is referenced from the frame of the base of the arm. For this reason, all the frames cited hereinafter took this as origin.

Continuing with the object frame estimation, first step consists on positioning it. The ideal reference position would be the center of masses of the object. But, as it cannot be obtained from the partial reconstruction of the object, the center of the bounding box is taken as estimated center of masses. Thus, the object frame is placed there.

Concerning the orientation. It may vary depending on the strategy that the user wants to follow to grasp the object and depending on the orientation of the manipulator frame. In order to facilitate the explanation, it has been supposed that the gripper used is the UJIOne [Ribas et al., 2015] (see Fig.4.5) and the object will be grasped by wrapping it through its minimum moment of inertia. Next steps are followed to find the correct orientation:

- It is desirable to grasp the object through its minimum moment of inertia. The shortest edges of the bounding box of the object are parallel to this moment of inertia. Therefore, the component y of the frame should be placed parallel to these edges. But, if they are perpendicular to the plane, the grasp is not possible by this sides as one of the fingers would collide with the floor. In this cases, the component y is placed parallel to the middle moment of inertia.

- The grasp should also be carried out along its maximum moment of inertia. The longest edges define the direction of this moment. Thus, the component x of the frame is placed parallel to these edges.

- The component z of the object frame, is placed perpendicular to the components x and y. It is, following the resulting direction from cross-multiplying the directions of the components x and y. This last component, z, defines the approximation direction, thus, it must always be pointing opposite to the position of the base of the arm. In order to check if it is happening, we calculate the distance between the center of the object and the base of the arm. If a point displaced a bit along the z direction is further from the base of the arm than the center of the object, then, the object frame is well oriented. Otherwise, the frame must be rotated π radians around component x.

Following these steps, the frame is properly oriented(see Fig. 4.6), and prepared for the next phase.

4.3.3 Grasping Characterization

Once the planner has all needed information about the object of interest, this must calculate an end-effector pose that allows the manipulator to grasp the object properly. A grasping pose is considered reliable if it is positioned and oriented inside a Cartesian ranges. These ranges are referenced from the previously calculated object frame, and can vary depending on the manipulator and object sizes.

It may be infinite poses that accomplish these limits. Thus, in order to define which is the best one, a set of tasks with different priorities are defined. The grasping pose which better fulfil these tasks will be the selected one.

In order to reach the best grasping pose, the grasp planner uses the Multi-Task Priority algorithm, explained in the Chapter 3. This algorithm allows to simulate the movements that the manipulator would perform while it is trying to accomplish the defined tasks, at the same time that the constraints are respected.

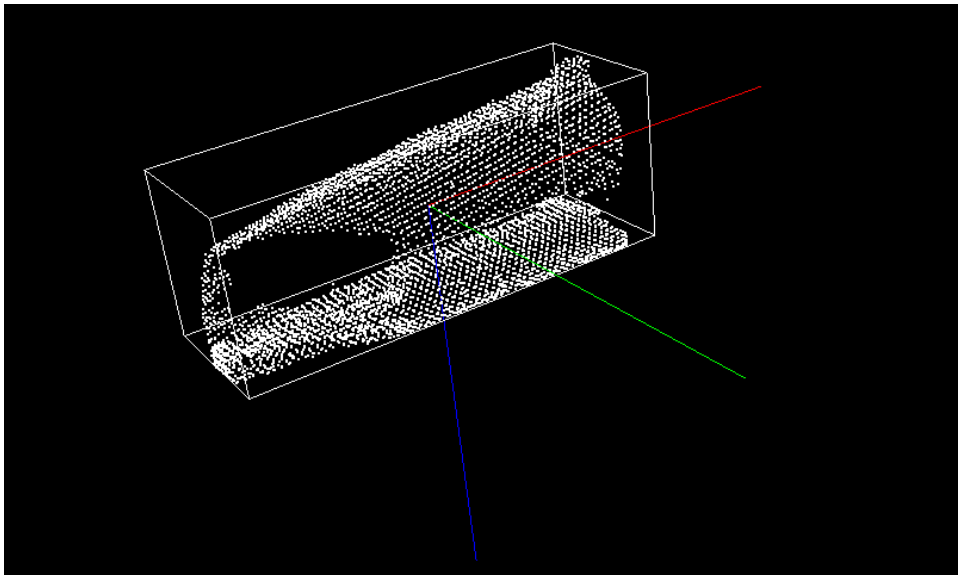


Figure 4.6: Reconstruction of the object of interest with its frame properly positioned and oriented.

Moreover, the use of the Multi-Task Priority algorithm allows to define virtual end-effectors and apply tasks and constraint over them. For example, the palm of a gripper could be defined as virtual end-effector. Then, a task which tries to position the palm as close as possible to the object could be defined and also a constraint that does not allows the palm to contact with it.

Concerning the simulation, if all the task and constraints were applied directly, it may happens that there exists a valid solution but it will never be reached because it would suppose to overcome momentarily any of the constraints, and the Multi-Task Priority algorithm will not permit that. In order to avoid this problem, the simulation is carried out in three phases:

- In the first phase, the algorithm only takes into account the constraints related with the joints, it is the joint limits. The tasks applied in this phase try that all the end-effectors reach a position inside their range defined as valid to perform the grasping. In this phase the orientations are not taken into account. When a stopping criterion, which will be explained later, is reached, this phase is finished. If the end-effectors are positioned inside their allowed ranges, the next phase is executed, otherwise, the algorithm concludes without finding a valid solution.

- For the second phase, the Cartesian constraints related with the position are added to the joint limits. Next, the tasks try to orientate

the different end-effectors so they are inside the allowed range of orientations. As in the previous phase, when the stopping criterion is reached, if the end-effectors accomplish the limit conditions, the next step is executed, otherwise, there is not solution and the algorithm concludes.

- If the third phase is reached, it means that there exist valid poses to perform the grasp. Therefore, in this phase, the algorithm decides which of them is the ideal one. To do so, all the constraints are added to the algorithm. Next, the different tasks that define the suitability of a grasping pose, are tried to be fulfilled. When the stopping criterion is reached, the current pose of the real end-effector defines the best feasible grasping pose.

Concerning the stopping criterions, there are three conditions that can stop the current phase of the algorithm. A maximum number of iterations have been performed, the sum of the joint velocities returned is lower than a certain value or the highest priority tasks error is lower than a certain value.

4.4 Experimental Results

Three experiments have been performed to evidence the functionality of the detailed methodology. As in the last chapter, the used system is composed by the robotic manipulator ECA-CSIP Light-wight ARM 5E [Fernández et al., 2013] which has been attached to the Girona 500 AUV [Ribas et al., 2012]. The vehicle has been configured with 4 degrees of freedom (x, y, z, yaw) and the manipulator with 4 more ($slew$, $shoulder$ and $jawRotate$).

At the same time, the manipulator has a gripper in its extreme. This gripper is formed by 4 parallel fingers of 26 cm length (G_{length}), which can achieve a maximum aperture of 30 cm (G_{open}).

I-AUV's manipulators usually perform more accurate movements than AUVs. Therefore, performing a grasping using only the manipulator, while the vehicle is trying to keep its current position, increases the probability of success in the manipulation. Taking advantage of the Multi-Task Priority Algorithm characteristics, in a first iteration, a valid grasping pose reachable moving only the manipulator is searched. If no one is found, the algorithm is executed again allowing also vehicle movements.

4.4.1 Experiments definition

The input of the three experiments is a partial reconstruction of a scene where several objects are placed over a flat surface.

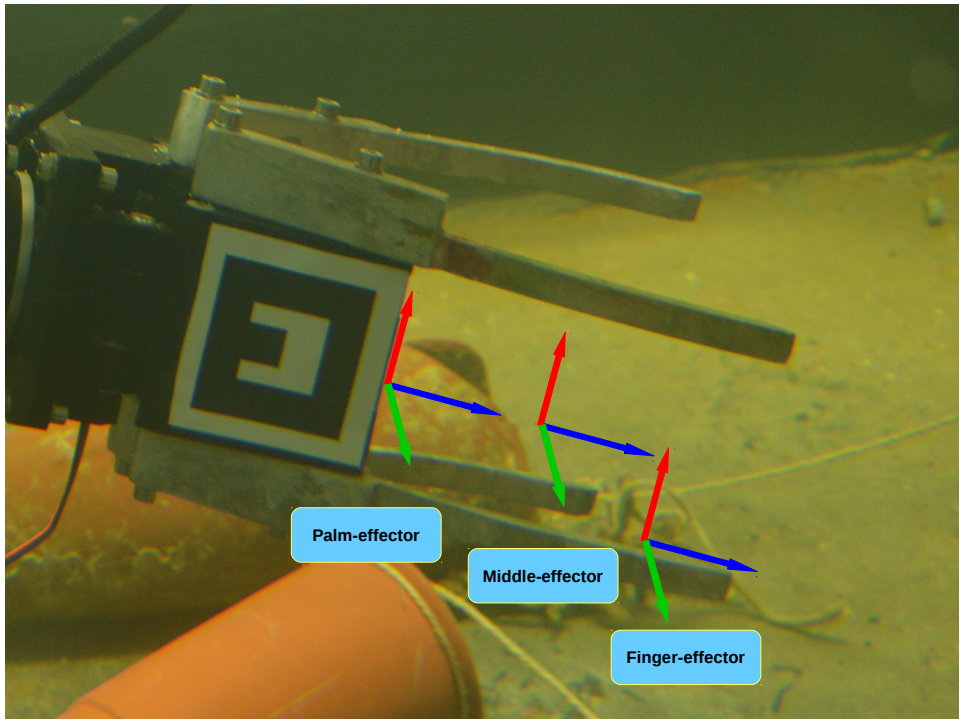


Figure 4.7: Frames of the three virtual end-effectors used for calculating the grasping pose.

This point cloud is segmented and the objects that do not accomplish the conditions described below, are discarded. These conditions have been defined taking into account the size limitations of the used gripper.

1. All the bounding box axes must measure less than 50cm
2. None of the bounding box axes can measure less than 5cm
3. The bounding box axis perpendicular to the plane must measure more than 10cm
4. One of the two remainder bounding box axes must measure less than 25cm

From the remaining objects, the one with highest volume is chosen. In this case the green object

The strategy chosen for grasping the object consists on wrapping it. Thus, the object frame has been estimated as in the example of the Section 4.3.3.

Concerning the simulation, three virtual end-effectors have been defined. The first one, finger-effector, is positioned between the end of the two lower fingers of the gripper. The second one, palm-effector, is positioned in the palm of the gripper. The last one, middle-effector, can be found in the middle of the gripper. The three frames are oriented with the same orientation, which coincide with the real end-effector orientation (see Fig. 4.7).

The Cartesian ranges that define a valid grasp for each object, depend of the size of this object. Supposing that the sizes of the bounding box that enclose the object are(x_{bb}, y_{bb} and z_{bb}), these ranges are defined by:

Table 4.1: Cartesian limits of the three virtual end-effectors for the Grasp Planning experiments. (angular units in radians)

End-effector	Limits	x	y	z	roll	pitch	yaw
finger-effector	**max_limit**	$0.4 * x_{bb}$	$\frac{G_{open}-y_{bb}}{2}$	$0.45 * z_{bb}$	0.4	$\frac{\pi}{2}$	0.1
	min_limit	$-0.4 * x_{bb}$	$-\frac{G_{open}-y_{bb}}{2}$	0	-0.4	$-\frac{\pi}{2}$	-0.1
palm-effector	**max_limit**	$0.4 * x_{bb}$	∞	$-0.5 * z_{bb}$	2π	2π	2π
	min_limit	$-0.4 * x_{bb}$	$-\infty$	$-\infty$	-2π	-2π	-2π
middle-effector	**max_limit**	∞	∞	∞	2π	2π	2π
	min_limit	$-\infty$	$-\infty$	$-\infty$	-2π	-2π	-2π

The limits of the z components, force the lower fingers two cross at least half of the object, without allowing these fingers to cross it completely, avoiding the possibility of contact with the floor. At the same time, the palm of the gripper is not allowed to contact with the object.

The y components have been limited to avoid that the fingers contact with the object during the grasping approximation.

The component x is limited to force that the final position of the gripper is always over the object.

The limitations of the *roll* component also avoid that the fingers contact with either the object or the floor.

The *pitch* component has been limited to avoid a wrong approximation direction, which could produce an unexpected contact.

The last component, *yaw*, has been strictly limited. This is because a small rotation in this component could trigger a finger collision with the object during the approximation.

Once the constraints that defines a valid grasping pose have been presented, next step consists on defining the tasks used in each of the phases

of the grasp planning.These tasks have been structured depending on their priority level, having the task with priority n more priority that the task with priority $n + 1$.

In the first phase, the goal is to position the end-effectors inside the previously defined limits related to the position (see Tab. 4.2. To do so, the algorithm tries to cross with the finger-effector half of the object, which is the minimum object depth allowed, and tries to position the middle-effector also inside the allowed limits.

Table 4.2: Cartesian tasks for ranking the valid grasping poses

Priority	**End-effector**	**Task**
1	middle-effector	$x = 0$
	middle-effector	$y = 0$
	finger-effector	$z = 0$

The goal of the second phase is to properly orientate the end-effectors (see Tab. 4.3). As all the end-effectors have the same orientation, it is not relevant which of the end-effectors accomplishes the tasks. The tasks have been sorted depending on how restrictive the associated limit is.

Table 4.3: Tasks for reaching a position inside the limits.

Priority	**End-effector**	**Task**
1	finger-effector	$yaw = 0$
2	finger-effector	$roll = 0$
3	finger-effector	$pitch = 0$

The tasks of the third phase is in charge of finding the preferable grasping pose among all the poses that are inside the limits (see Tab. 4.4).

The two tasks with the highest priority try that the largest surface of the fingers were around the object during the grasp. To do so, they try to deep the ending of the fingers as much as possible around the object. At

the same time the palm of the gripper is positioned as close to the object as possible without contacting with it.

The rest of the tasks try to position the final pose in the middle of the object, keeping the orientation defined by the object frame.

It is worth mentioning that the used constraints and tasks have been obtained trial and error.

Table 4.4: Task for reaching an orientation inside the limits.

Priority	**End-effector**	**Task**
1	finger-effector	$z = 0.45 * zsize$
	palm-effector	$z = -0.55 * z_{size}$
2	finger-effector	$roll = 0$
3	finger-effector	$yaw = 0$
4	middle-effector	$y = 0$
5	finger-effector	$x = 0$
	palm-effector	$x = 0$

4.4.2 First Experiment

The input of the first experiment is the point cloud shown in the Fig. 4.2. The detection of the floor and the 4 objects placed over it can be seen in the Fig. 4.3).

Next, the bounding boxes of these objects have been calculated (see Fig. 4.4), obtaining the following estimated object sizes (see Tab. 4.5).

Only the green object accomplishes all the conditions that define a graspable objects. All the axis are higher than 5cm and lower than 50. The axis perpendicular to the plane, which is the middle one, is higher than 10cm, and one of the other is lower than 25cm.

The three remainder objects do not accomplish the condition of having all the axis higher than 5cm, thus, they cannot be grasped.

Table 4.5: Estimated size of the objects placed in the scene of the First Experiment. The values show the size in cm of the axes of the bounding boxes that enclose the objects.

Object color	**Major**	**Middle**	**Minor**	**Graspable**
Green	31.488	13.976	10.423	✓
Blue	8.852	6.264	4.077	
Cyan	9.156	4.481	1.919	
Yellow	6.357	2.942	2.057	

As there is only one graspable object, this is the selected one.

The frame estimated for the selected object can be seen in the Fig. 4.3.2. This frame is located, with respect to the base of the arm at the position($x = -0.035, y = -0.052, z = 1.073$)m and with an orientation ($roll = -0.013, pitch = -0.08, yaw = 2.27$)rad. This frame is considered the origin during the simulation, and all the tasks are defined with respect to it.

Having the object frame defined, the simulation can start. In a first iteration, the vehicle movements are not allowed. If no solution is found, a second simulation allowing these movements will be performed.

At the end of the first phase of the simulation (see Fig.4.8), the three tasks have been practically fulfilled. It is, the components x and y of the middle-effector and the z component of the finger-effector have reached the position 0. Moreover, the arm joints have respected their limits (see Fig. 4.9). Thereby, the simulation can continue with the second phase.

The goal of the second phase is to orientate the end-effectors like the object frame without overpassing the Cartesian limits related to the position and the joint limits. Fig.4.10 shows the evolution of the orientations during this phase. The algorithm stops when there is no allowed movement that improve the end-effectors orientation. Even though the task have not been completely fulfilled, the final orientation is inside the allowed orientation range, thus the algorithm can continue. Fig. 4.11 demonstrates that neither the joint nor the end-effectors have overpassed their limits.

At that moment of the experiment, it is already known that there is valid solutions for grasping the object without moving the vehicle. This phase looks for the best solution among all the valid ones following the task hierarchy previously defined. In this phase the orientation limits are also

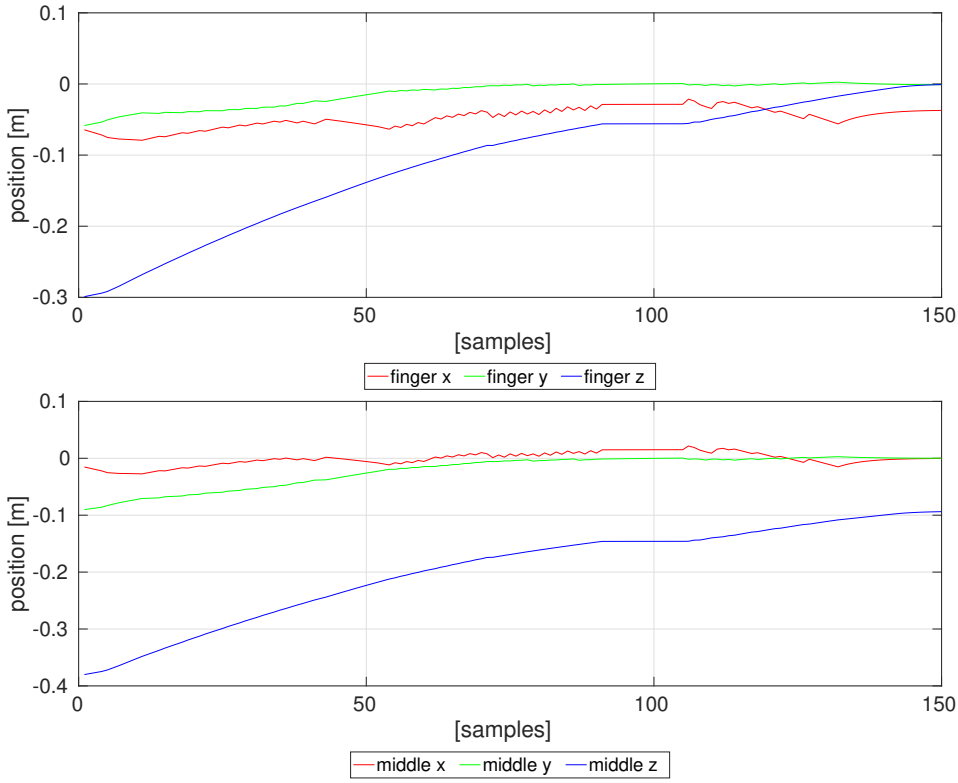


Figure 4.8: Trajectory followed by the finger-effector (top) and middle-effector (bottom) during the first phase of the First Experiment.

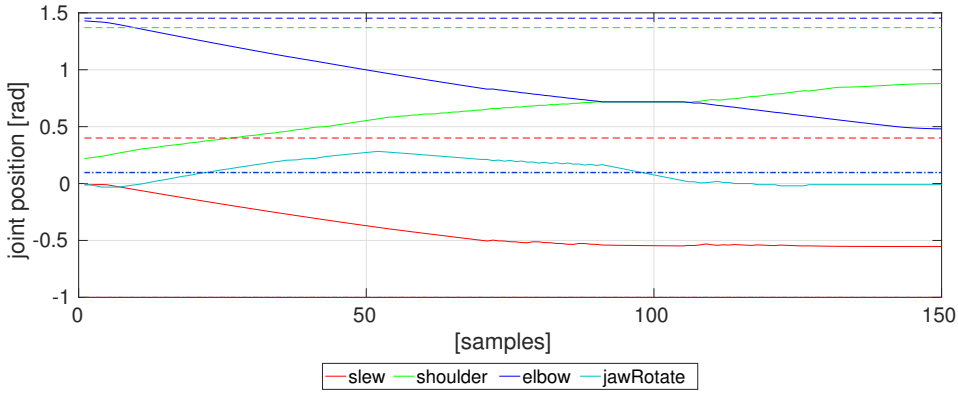


Figure 4.9: Arm joint values during the first phase of the First Experiment. Upper and lower limits described with dashed lines.

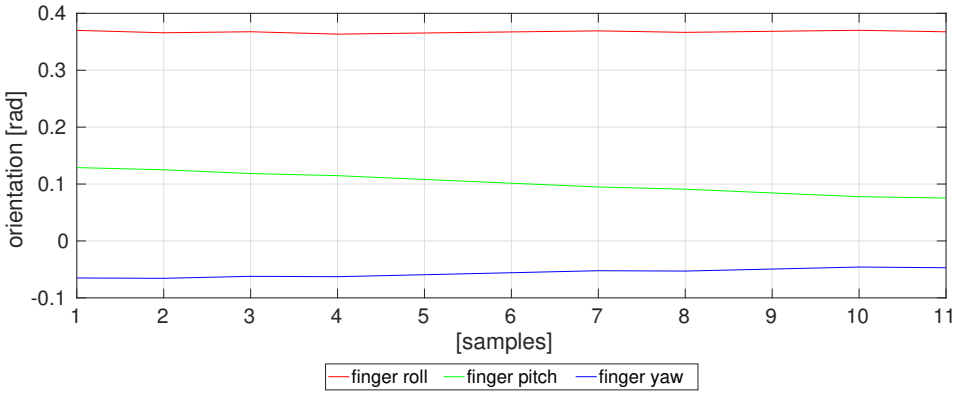


Figure 4.10: Orientation reached by the finger-effector during the second phase of the First Experiment.

introduced.

Fig. 4.12 shows the final position and Fig. 4.13 the final orientation reached by the different end-effectors at end of the third phase of the experiment. And Fig. 4.14 displays the joint values during the same phase of the experiment.

Concerning this third phase, the tasks with the highest priority are related to the z component of the finger-effector and palm-effector. Thus, the movements performed during this phase tries specially to accomplish these tasks. Taking into account that the arm has only 4 degrees of freedom, it is understandable that some movements needed to accomplish these tasks, can worse the state of tasks with lower priority. At the end, both z components have improve their initial positions. On the other hand, the x component of the finger-effector has clearly worse its initial state.

After successfully finishing the first iteration of the simulation, the second iteration is not needed. The final pose reached by the real end-effector is the output of the algorithm.

4.4.3 Second Experiment

After that I-AUV has reconstructed the scenario of the Second Experiment, the point cloud obtained is segmented (see Fig. 4.15), realizing that the scene is composed by four objects over a flat surface.

The bounding boxes of these objects are then extracted (see Fig. 4.16). These boxes approximate the real sizes of the objects. These obtained sizes are detailed in the Tab. 4.6.

Both green and blue objects accomplish all the conditions that define a graspable object. All the axis are higher than 5cm and lower than 50. The axis perpendicular to the plane, which is the major for the green object and

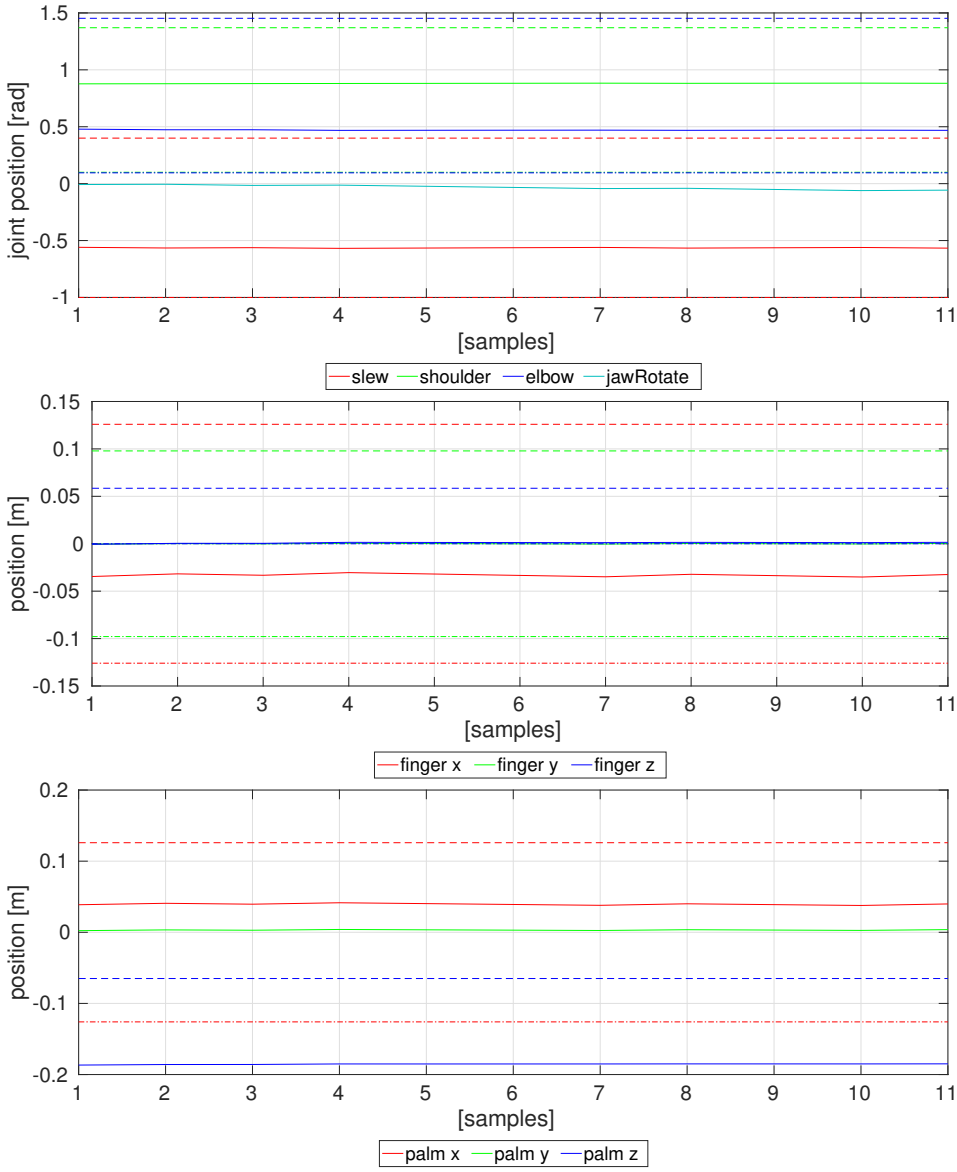


Figure 4.11: Joint values (top) and trajectory followed by the finger-effector(middle) and palm-effector during the second phase of the First Experiment. Upper and lower limits described with dashed lines.

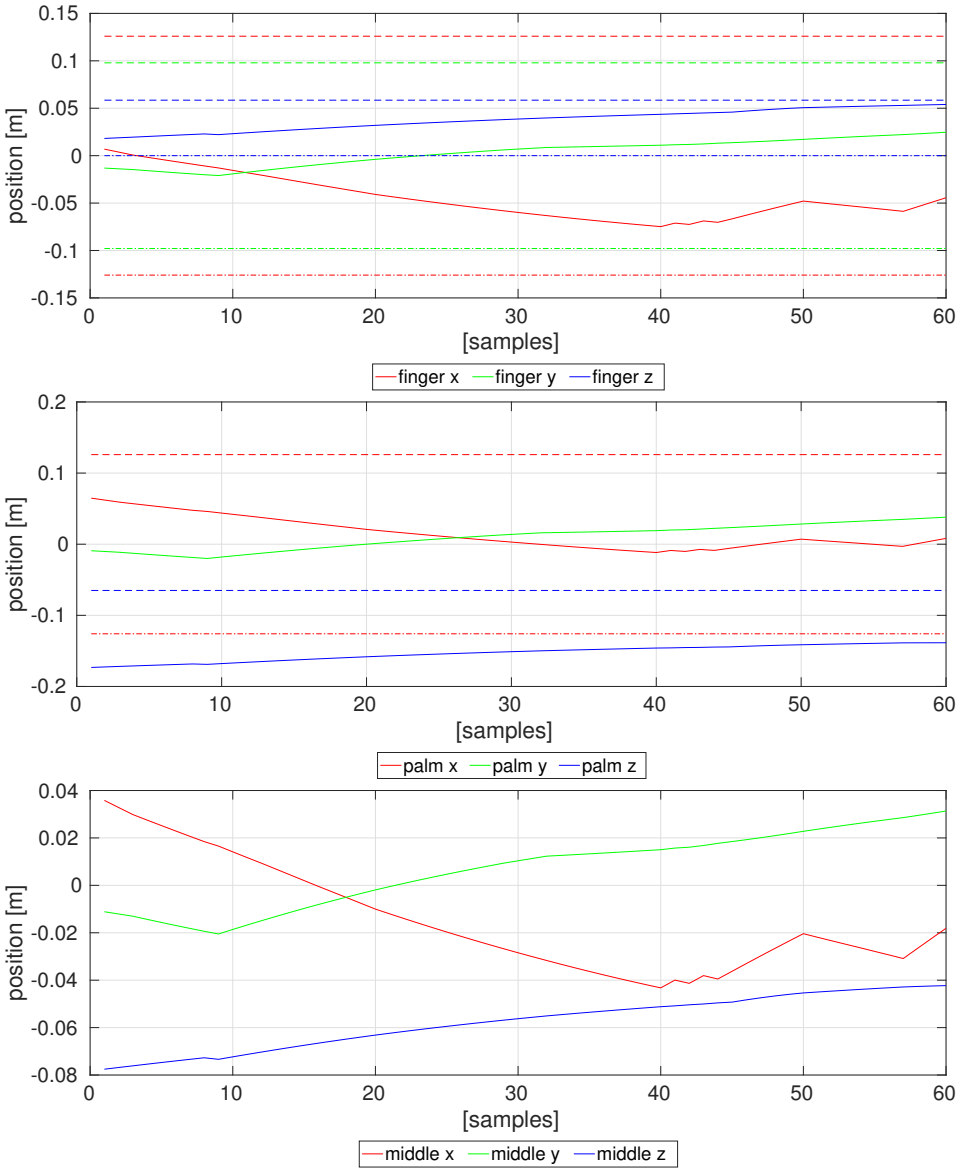


Figure 4.12: Trajectory followed by the finger-effector (top), palm-effector (middle) and middle-effector (bottom) during the third phase of the First Experiment. Upper and lower limits described with dashed lines.

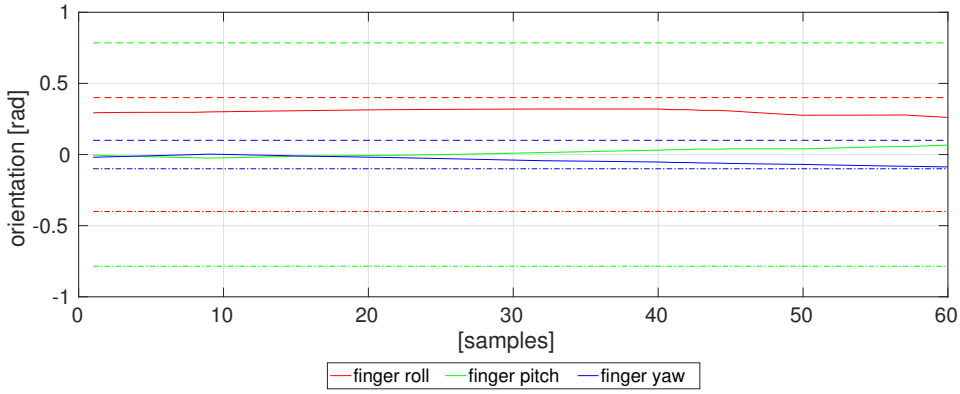


Figure 4.13: Orientation reached by the end-effectors during the third phase of the First Experiment. Upper and lower limits described with dashed lines.

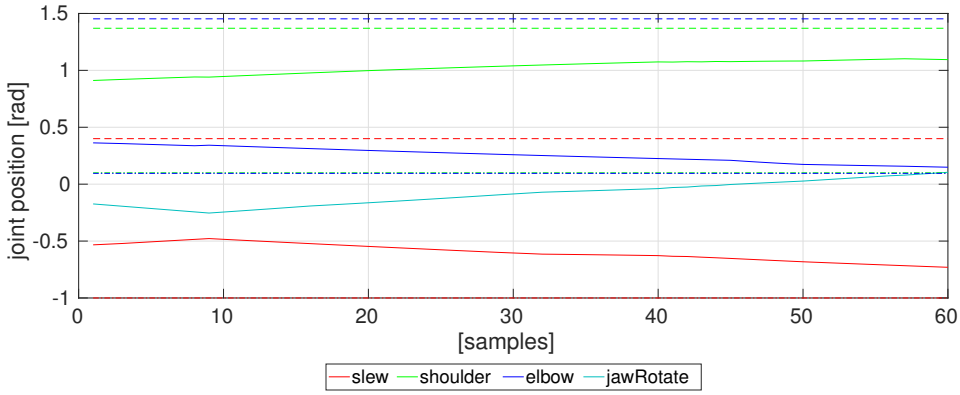


Figure 4.14: Arm joint values during the third phase of the First Experiment. Upper and lower limits described with dashed lines.

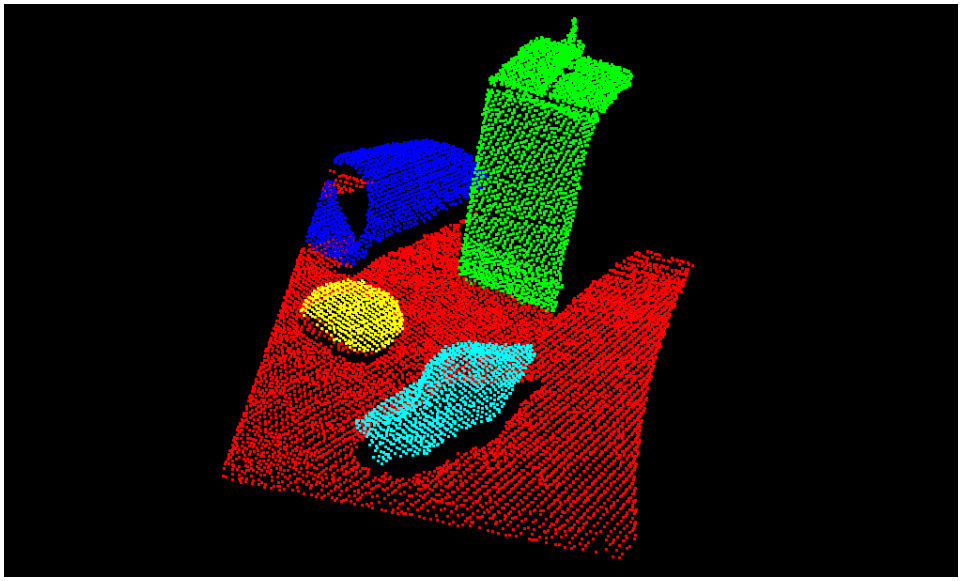


Figure 4.15: Second Experiment scene segmentation.

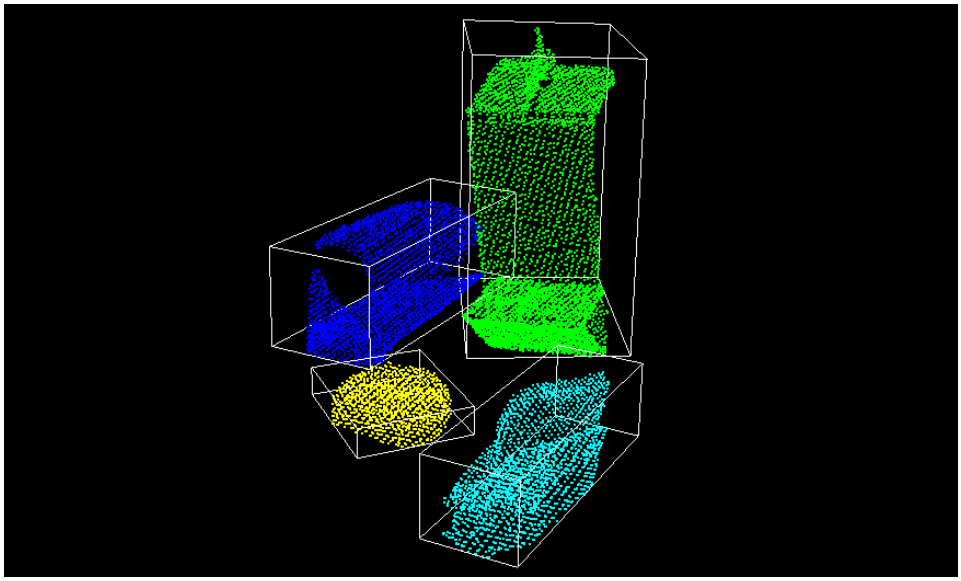


Figure 4.16: Bounding boxes of the objects placed in the scenario of the Second Experiment.

Table 4.6: Estimated size of the objects placed in the scene of the Second Experiment. The values show the size in cm of the axes of the bounding boxes that enclose the objects.

Object color	Major	Middle	Minor	Graspable
Green	29.849	16.478	12.587	✓
Blue	26.486	12.432	11.832	✓
Cyan	19.451	8.354	7.088	
Yellow	11.356	9.877	3.453	

the middle for the blue one are higher than 10cm, and at least one of the remaining axis is lower than 25cm.

The cyan object accomplishes the condition of having all the axis higher than 5cm and lower than 50, but the axis perpendicular to the plane, the middle one, is not higher than 10cm.

Last, the yellow object does not have all the axis higher than 5cm, thus it is also discarded.

Among the two graspable objects, the green one is the selected due to its volume ($6190.939cm^3$) is higher than the volume of the blue object ($3895.969cm^3$).

Next step consists on calculating the frame of the green object (see 4.17). Unlike the First Experiment, this object has its major axis perpendicular to the plane, thus the x component of the frame is placed perpendicular to it. The z component is placed parallel to the plane, which means that the desired approximation direction would be parallel to it.

The frame is located, with respect to the base of the arm, at the position($x = -0.36, y = 0.0, z = 0.67$)m and with an orientation ($roll = 0.0, pitch = -1.57, yaw = 3.14$)rad. As in the last experiment, this frame is considered the origin during the simulation, and all the tasks are defined with respect to it.

Having the object frame defined, the first simulation, where the vehicle movements are not allowed, can start.

Like in the First Experiment, at the end of the first phase of the simulation, the three tasks are completely fulfilled (see Fig. 4.18). Additionally, Fig. 4.19 shows how the arm joints have respected their limits. Elbow joint is close to its upper limit but it never crosses it.

The second phase tries to properly orientate the end-effectors. But, at

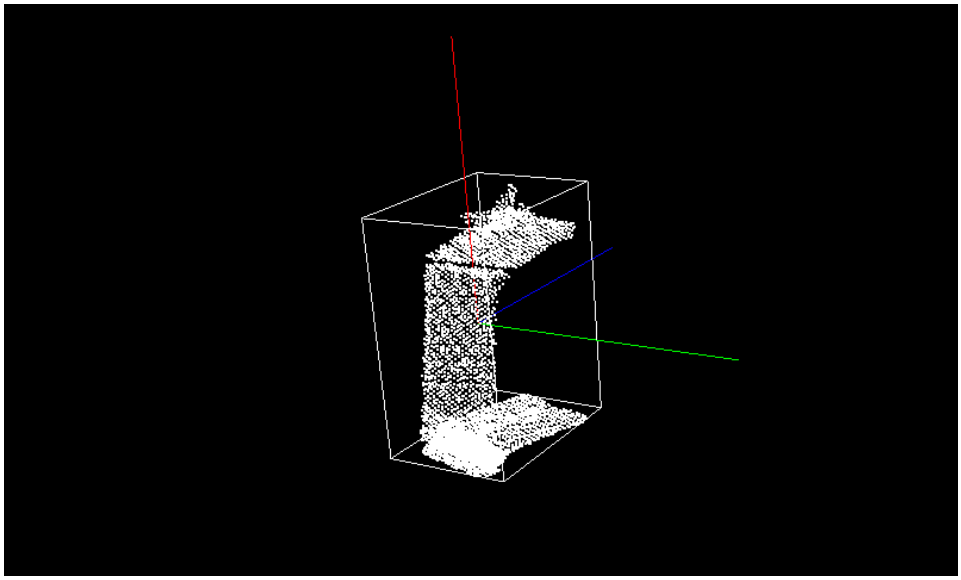


Figure 4.17: Frame of the object selected to be grasped.

the first iteration of this phase, the algorithm finds that there is not allowed movements that improve that current state of the tasks defined for this phase without breaking the limits. Fortunately, as can be seen in the Fig. 4.20, the first phase finished having the end-effectors an orientation which is inside the orientation limits. For that reason, the algorithm can continue with the third phase despite not having done any movement.

Next, it is time for the third phase, where the best solution among all the valid ones is searched. Fig. 4.21 shows the final position and Fig. 4.22 the final orientation reached by the different end-effectors at end of the third phase of the experiment. And Fig. 4.23 displays the joint values during the same phase of the experiment.

As in the first experiment, the number of joints of the arm limits the fulfilment of the tasks, providing the feeling that only the tasks with the highest priority are taken into account. Thus, the components z of finger-effector and palm-effector perform the most prominent movements towards their desired position. Meanwhile, the remaining components perform the necessary movements to allow the desired movement of the z components, without being able to avoid worsening their state.

Again, the first iteration of the algorithm finds a good grasping pose, and the second iteration is not needed.

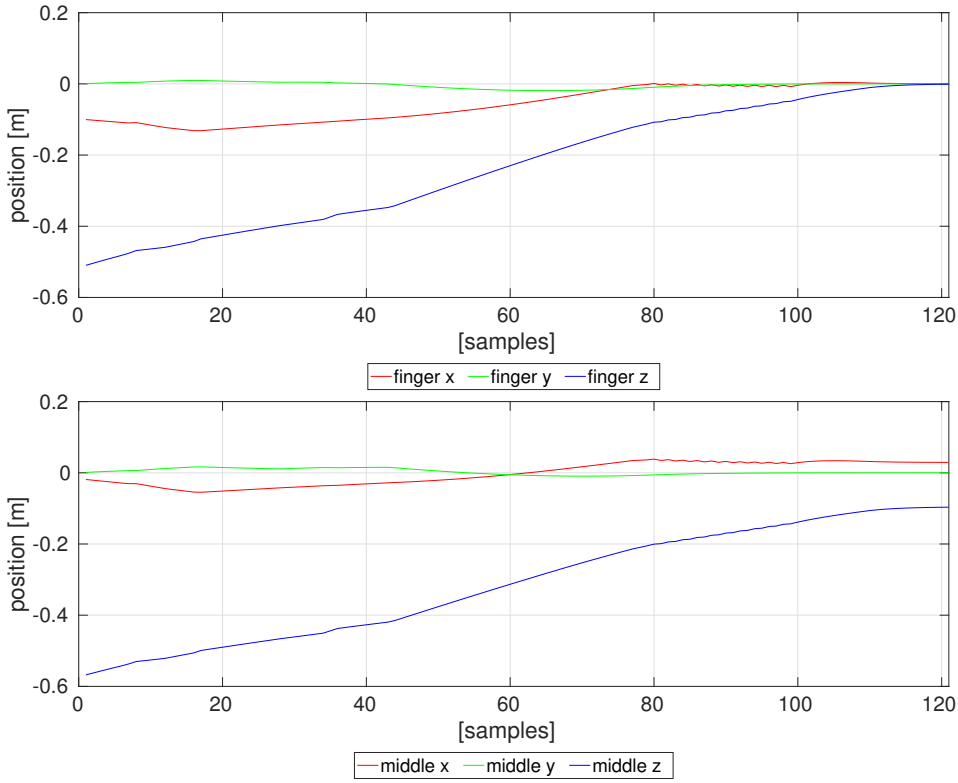


Figure 4.18: Trajectory followed by the finger-effector (top) and middle-effector (bottom) during the first phase of the Second Experiment.

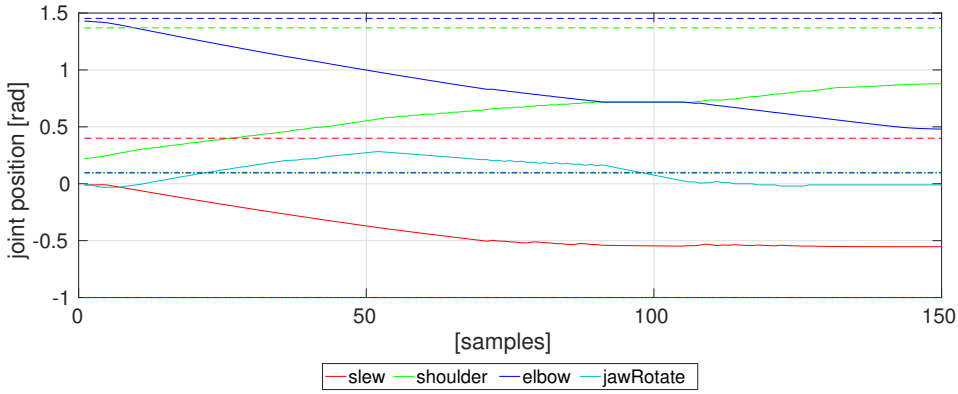


Figure 4.19: Arm joint values during the first phase of the Second Experiment. Upper and lower limits described with dashed lines.

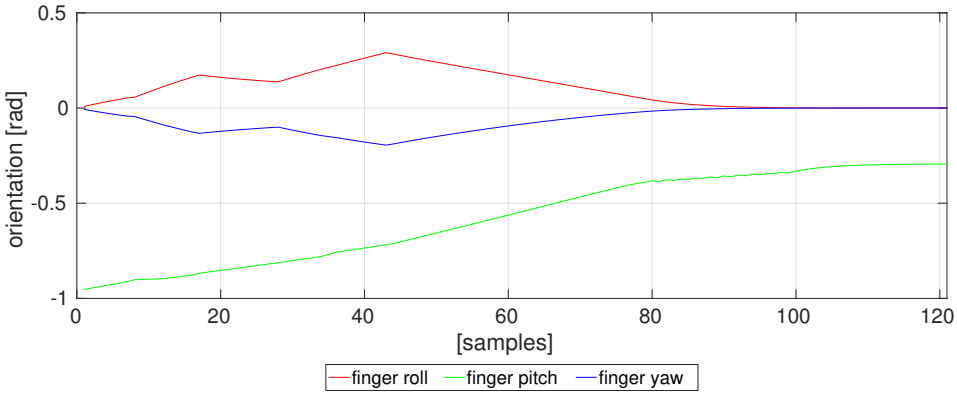


Figure 4.20: Orientation reached by the finger-effector during the first phase of the Second Experiment.

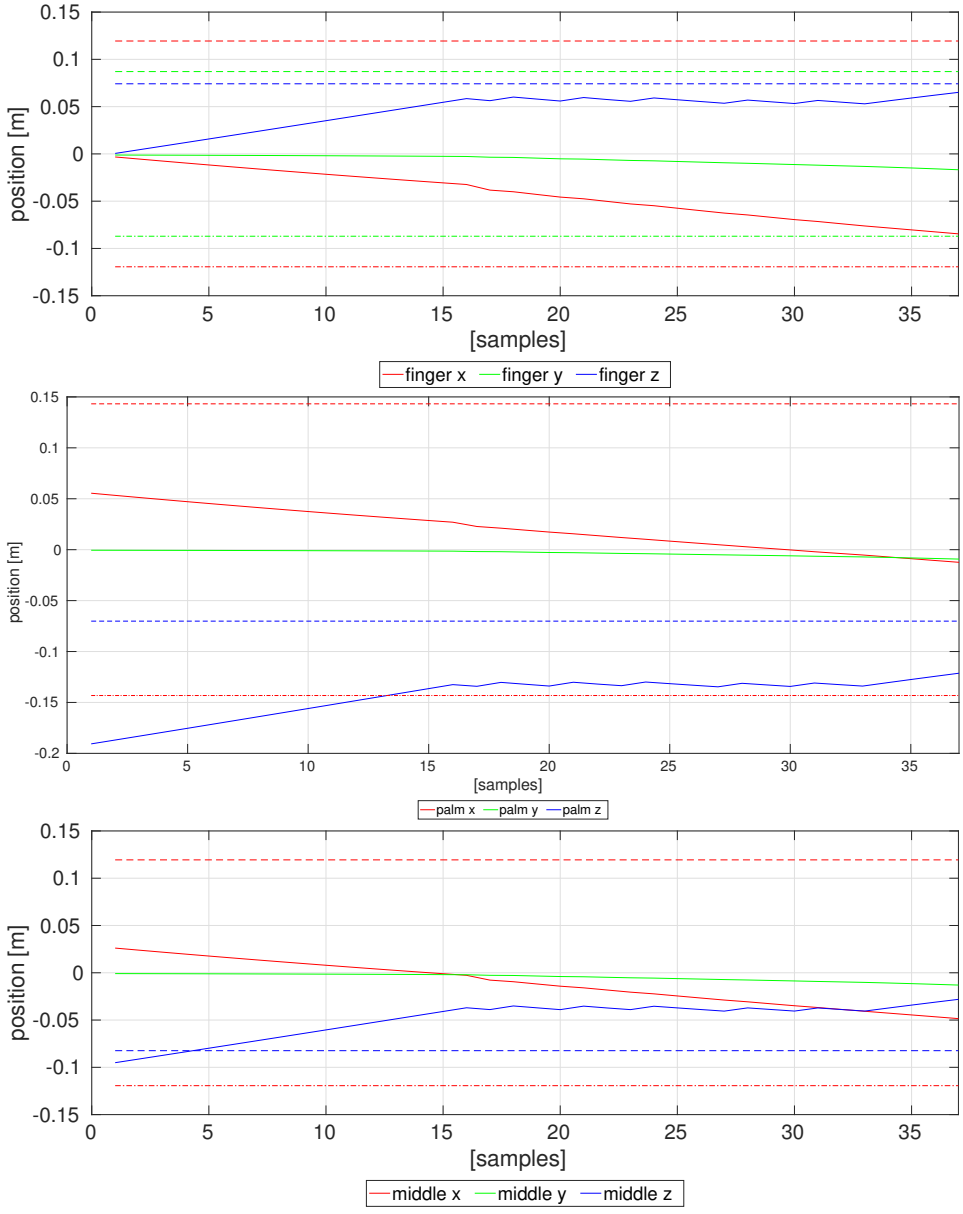


Figure 4.21: Trajectory followed by the finger-effector (top), palm-effector (middle) and middle-effector (bottom) during the third phase of the Second Experiment. Upper and lower limits described with dashed lines.

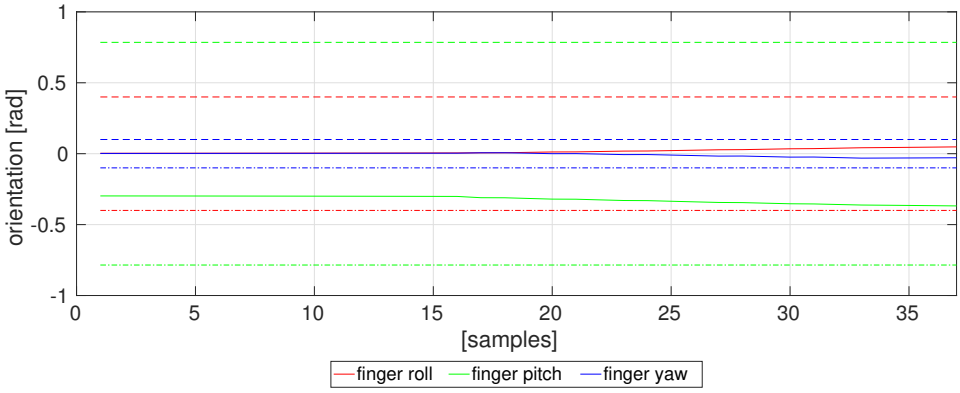


Figure 4.22: Orientation reached by the end-effectors during the third phase of the Second Experiment. Upper and lower limits described with dashed lines.

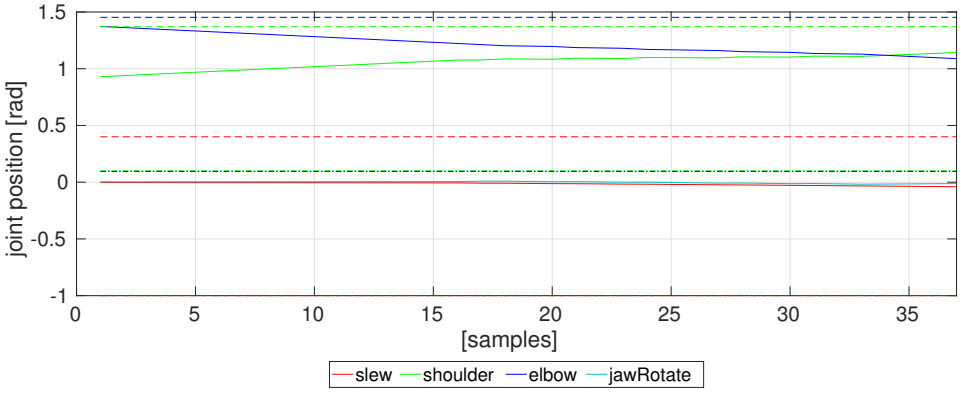


Figure 4.23: Arm joint values during the third phase of the Second Experiment. Upper and lower limits described with dashed lines.

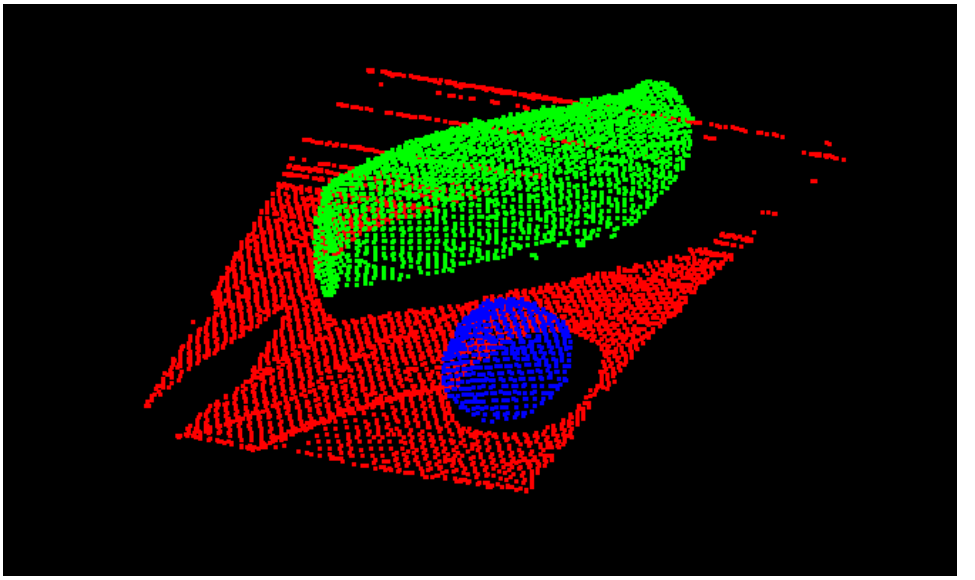


Figure 4.24: Third Experiment scene segmentation.

4.4.4 Third Experiment

The scenario of the Third Experiment is composed by two objects over a flat surface. The segmentation of the point cloud taken as input can be seen in the Fig.4.24.

After segmenting the scene, the bounding boxes of the detected objects are obtained (see Fig. 4.25). Using these boxes, the real sizes of the objects are approximated. These sizes can be seen in the Tab. 4.7.

Table 4.7: Estimated size of the objects placed in the scene of the Third Experiment. The values show the size in cm of the axes of the bounding boxes that enclose the objects.

Object color	Major	Middle	Minor	Graspable
Green	29.857	15.343	12.457	✓
Blue	10.865	9.032	4.532	

The green object accomplishes all the conditions for being grasped. All the axes are higher than 5cm and lower than 50. The axis perpendicular to the plane, which is the middle one, is higher than 10cm, and at least one of the remaining axis is lower than 25cm, the minor one in this case.

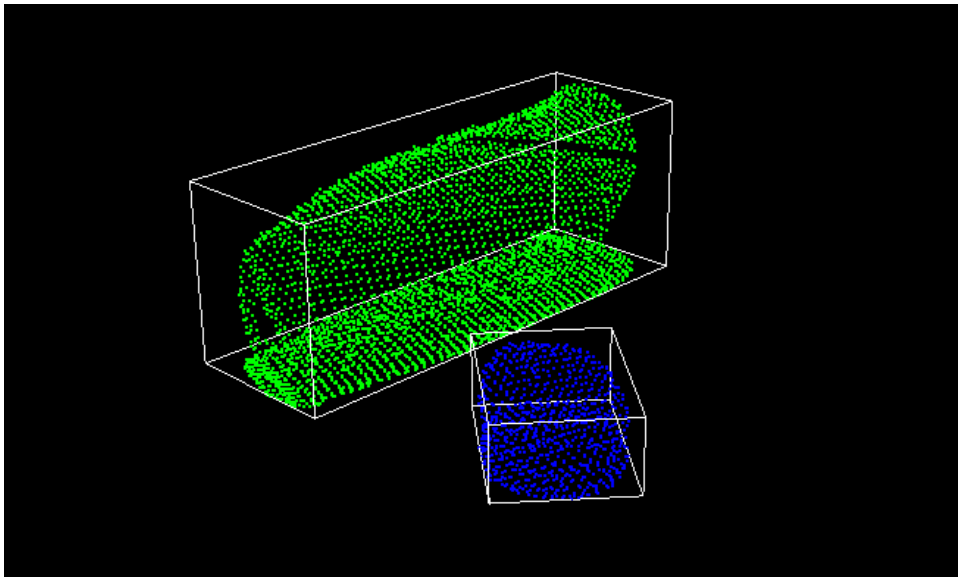


Figure 4.25: Bounding boxes of the objects placed in the scenario of the Third Experiment.

On the other hand, the blue object does not accomplish the condition of having all the axes higher than 5cm, thus it is discarded.

As the green is the only graspable object, it is the selected one.

Next, the frame of the selected object is calculated (see 4.26). In this case, the x component is placed parallel to the plane and the desired approximation direction, perpendicular to the plane.

The frame is located, with respect to the base of the arm, at the position($x = -0.273, y = 0.432, z = 1.3$)m and with an orientation ($roll = -0.007, pitch = -0.2, yaw = 2.64$)rad. Like in the other experiments, this frame is considered the origin during the simulation, and all the tasks are defined with respect to it.

Once the object frame has been obtained, the simulation can start. The first iteration without vehicle movements and, if needed, a second one allowing these movements.

When the first phase of the first iteration finishes, the tasks have not been fulfilled (see Fig 4.27). The slew and elbow joints have reached their limits, and they do not permit to continue improving the state of the tasks (see Fig. 4.28). Further, all the components have stopped out of their limit ranges. It means that there is no valid grasping pose reachable without moving the vehicle. For that reason, the first iteration finishes and a second iteration where the vehicle is used must start.

As can be seen in the Fig. 4.29, when the vehicle is used, all the tasks

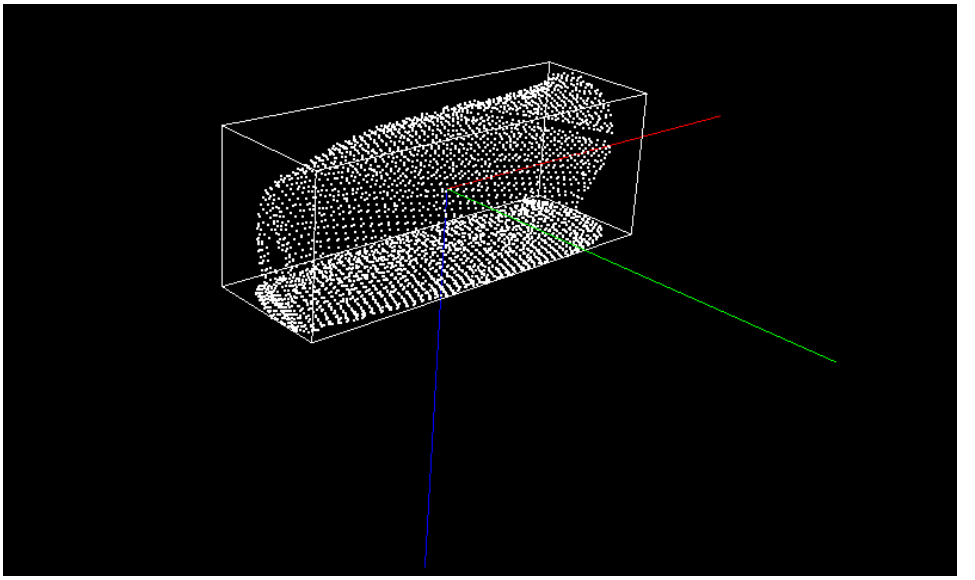


Figure 4.26: Frame of the object selected to be grasped.

defined for the first phase are accomplished. The x and y components of the middle-effector and the z component of the finger-effector have reached the 0 position. Fig. 4.30 demonstrates how both arm joints and vehicle have been used to reach the desired position.

Next, the second phase tries to properly orientate the end-effectors. The result of this phase (see Fig. 4.31) shows that the task related to the *pitch* and *yaw* have been completely fulfilled. The component *roll* have not being able to reach its desired value, but, nonetheless, it has stopped inside its permitted range. Thus, this phase have finished successfully. Fig. 4.32 and Fig. 4.33 demonstrate that all the limits have been respected during this phase.

It is worth noting, that due to there is not tasks related to the position of the end-effectors, the trajectory followed by the components x, y or z are not controlled and it can look strange. A clear example is the component y of finger- and palm-effector. This is not important as long as they do not affect the trajectory of the components which participate in the algorithm.

The last phase will look for the best grasp among the valid solutions.

Fig. 4.34 shows the final position and Fig. 4.35 the final orientation reached by the different end-effectors at the end of the third phase of the experiment. And Fig. 4.36 displays the arm and vehicle joints values during the same phase of the experiment.

As expected, the last phase of this experiment accomplishes much more tasks than the last phases of the previous experiments. The use of the vehicle

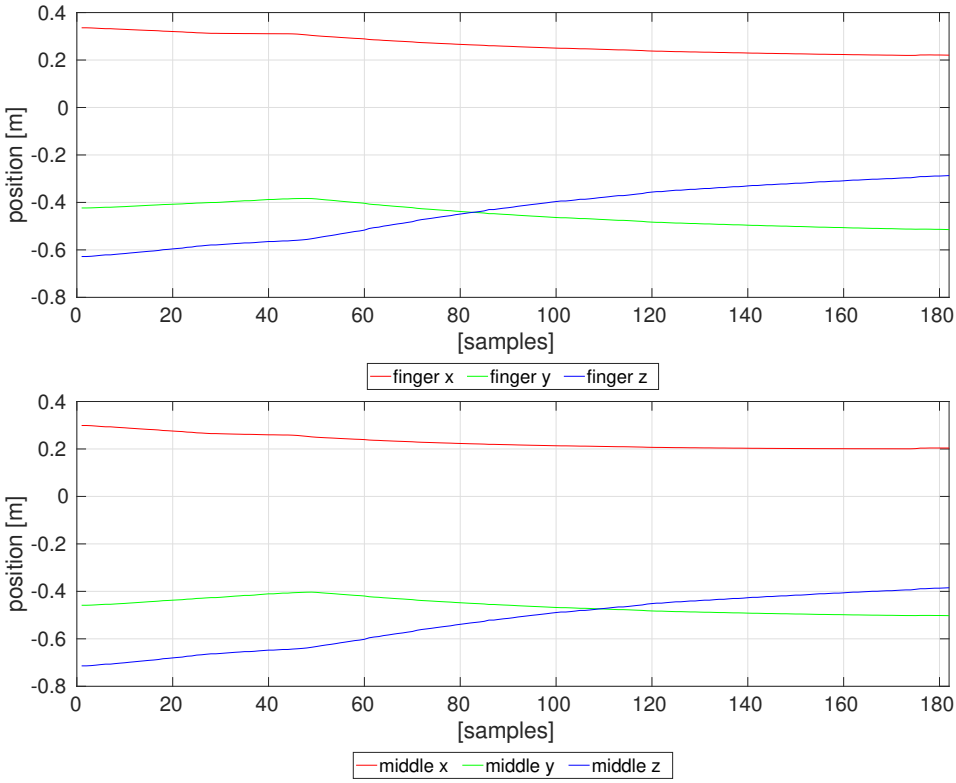


Figure 4.27: Trajectory followed by the finger-effector (top) and middle-effector (bottom) during the first phase of first iteration of the Third Experiment.

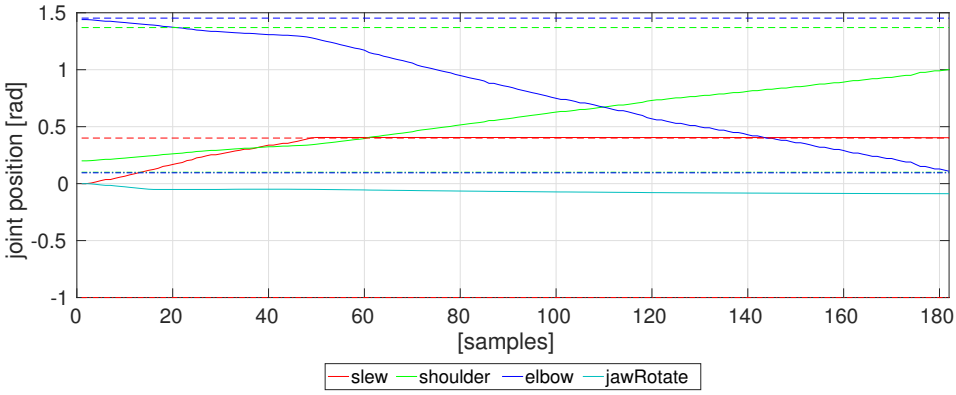


Figure 4.28: Arm joint values during the first phase of the first iteration of the Third Experiment. Upper and lower limits described with dashed lines.

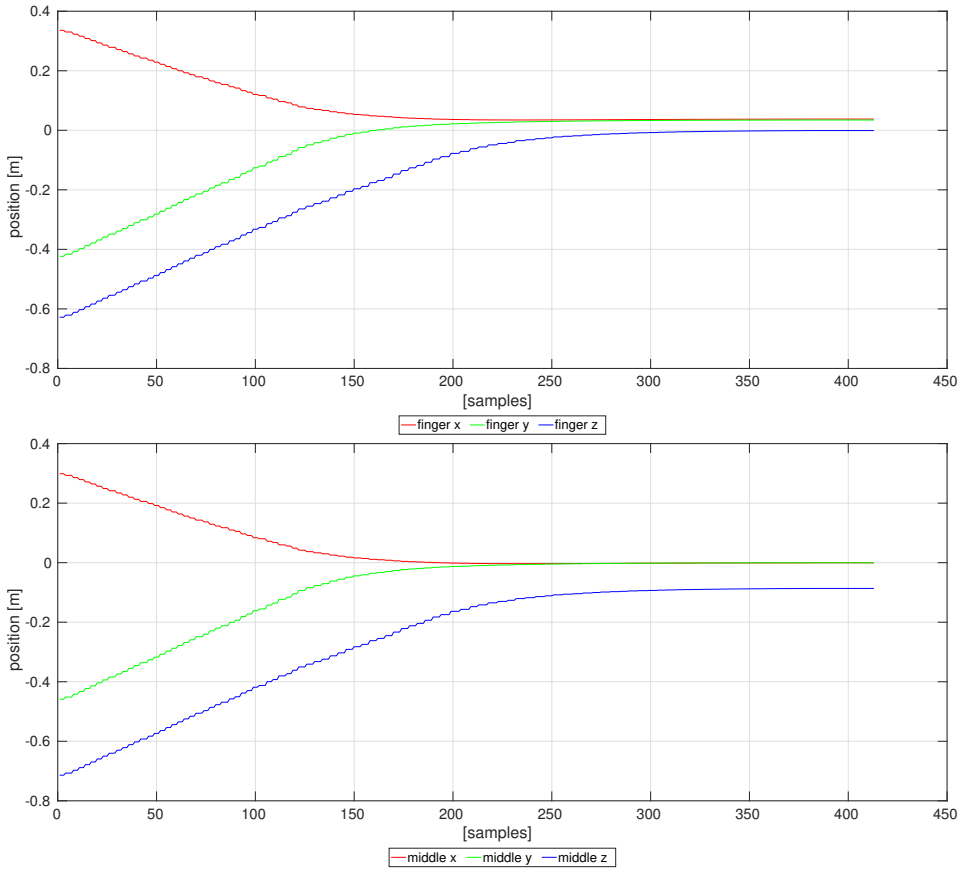


Figure 4.29: Trajectory followed by the finger-effector (top) and middle-effector (bottom) during the first phase of the second iteration of the Third Experiment.

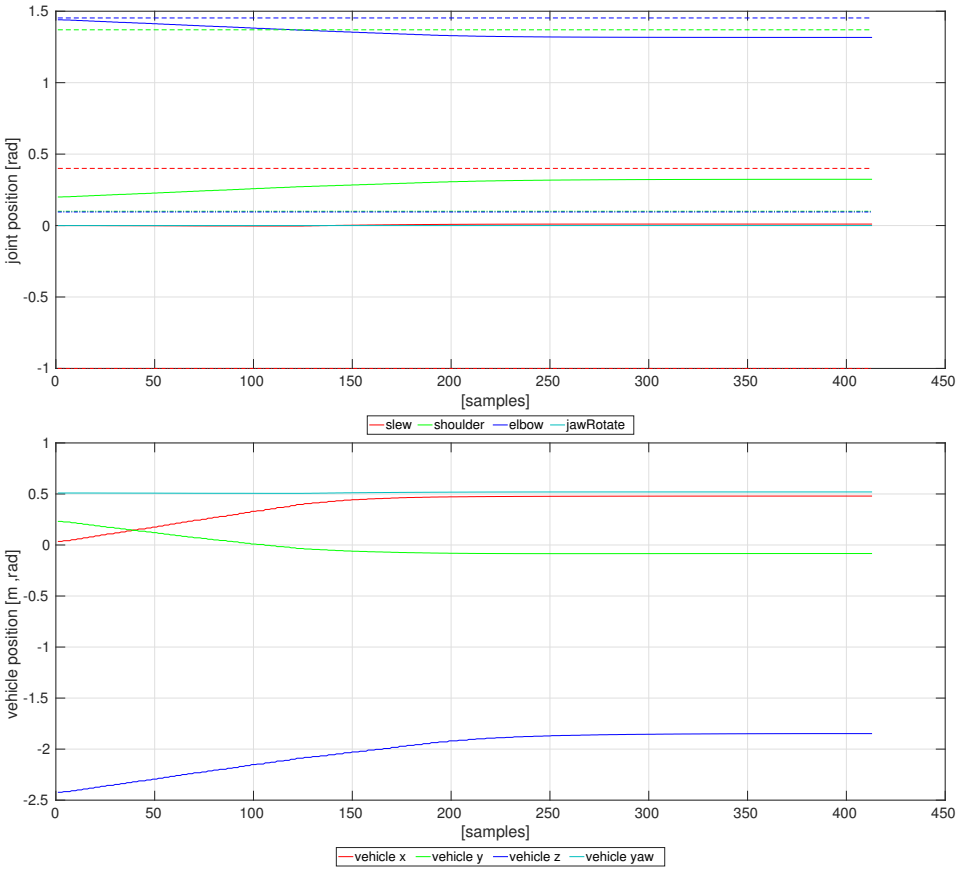


Figure 4.30: Arm (top) and vehicle (bottom) joint values during the first phase of the second iteration of the Third Experiment. Upper and lower limits described with dashed lines.

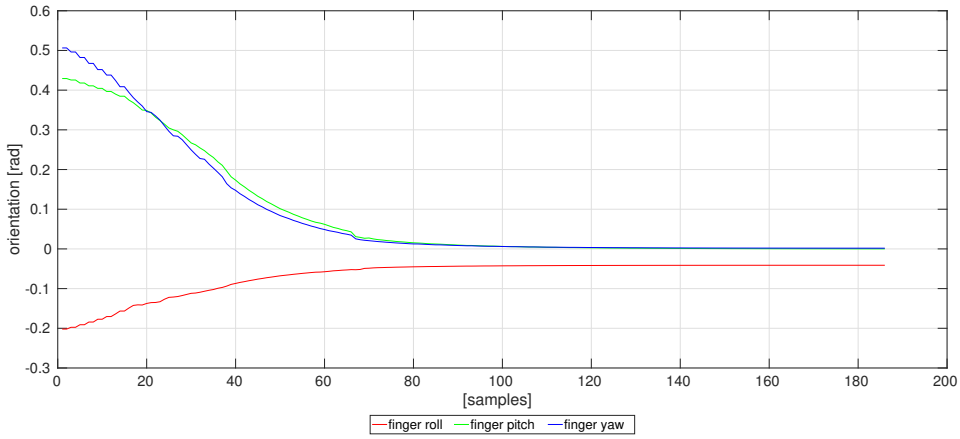


Figure 4.31: Orientation reached by the finger-effector during the second phase of the second iteration of the Third Experiment.

duplicate the number of joints available, increasing the variety of configuration reachable by the system.

The pose reached by the real end-effector is then given as output of the algorithm.

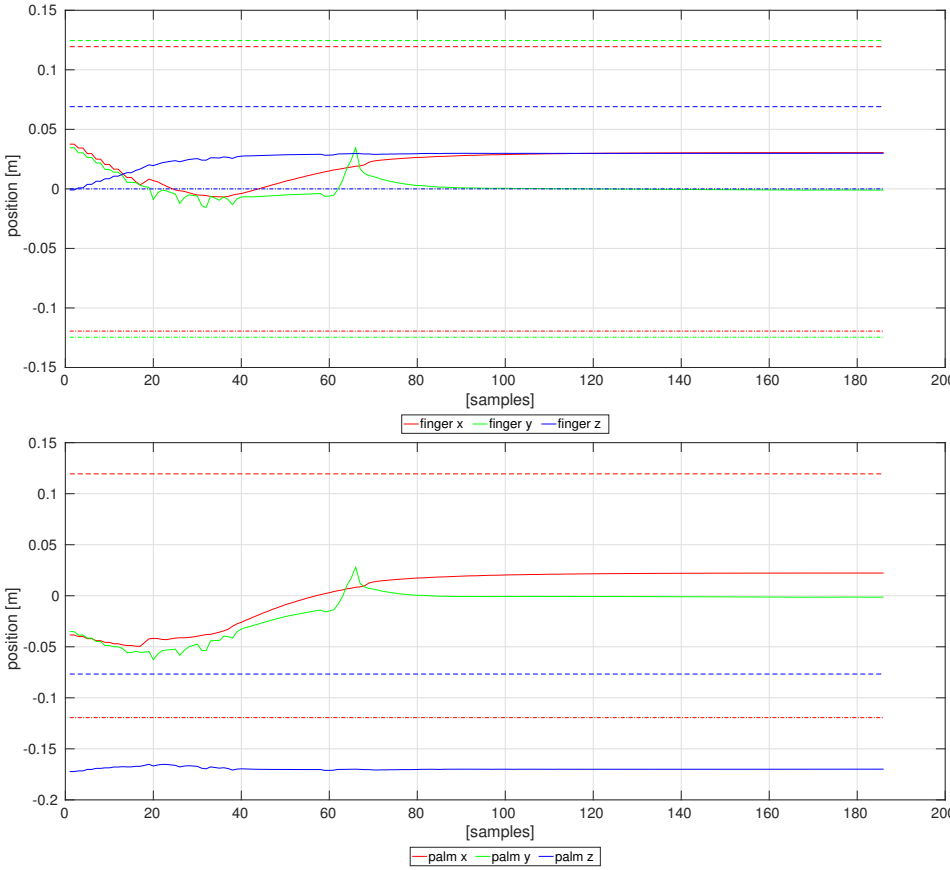


Figure 4.32: Trajectory followed by the finger-effector(top) and palm-effector (bottom) during the second phase of the second iteration of the Third Experiment. Upper and lower limits described with dashed lines.

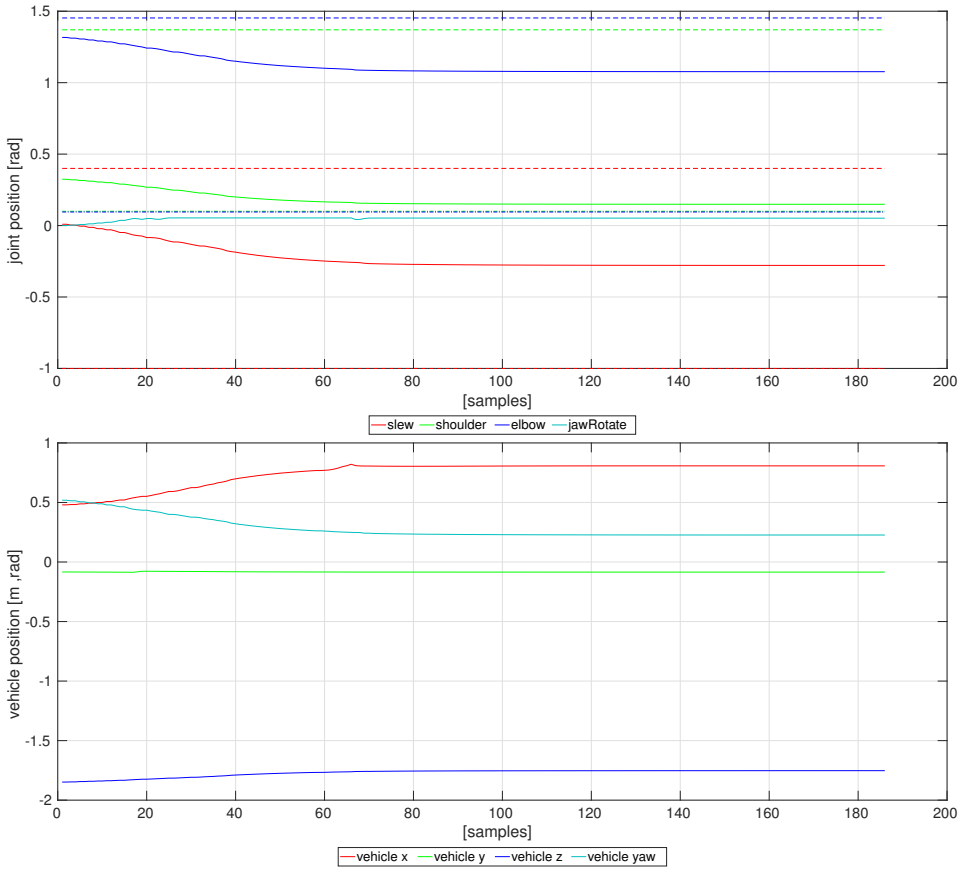


Figure 4.33: Arm (top) and Vehicle (bottom) joint values during the second phase of the second iteration of the Third Experiment. Upper and lower limits described with dashed lines.

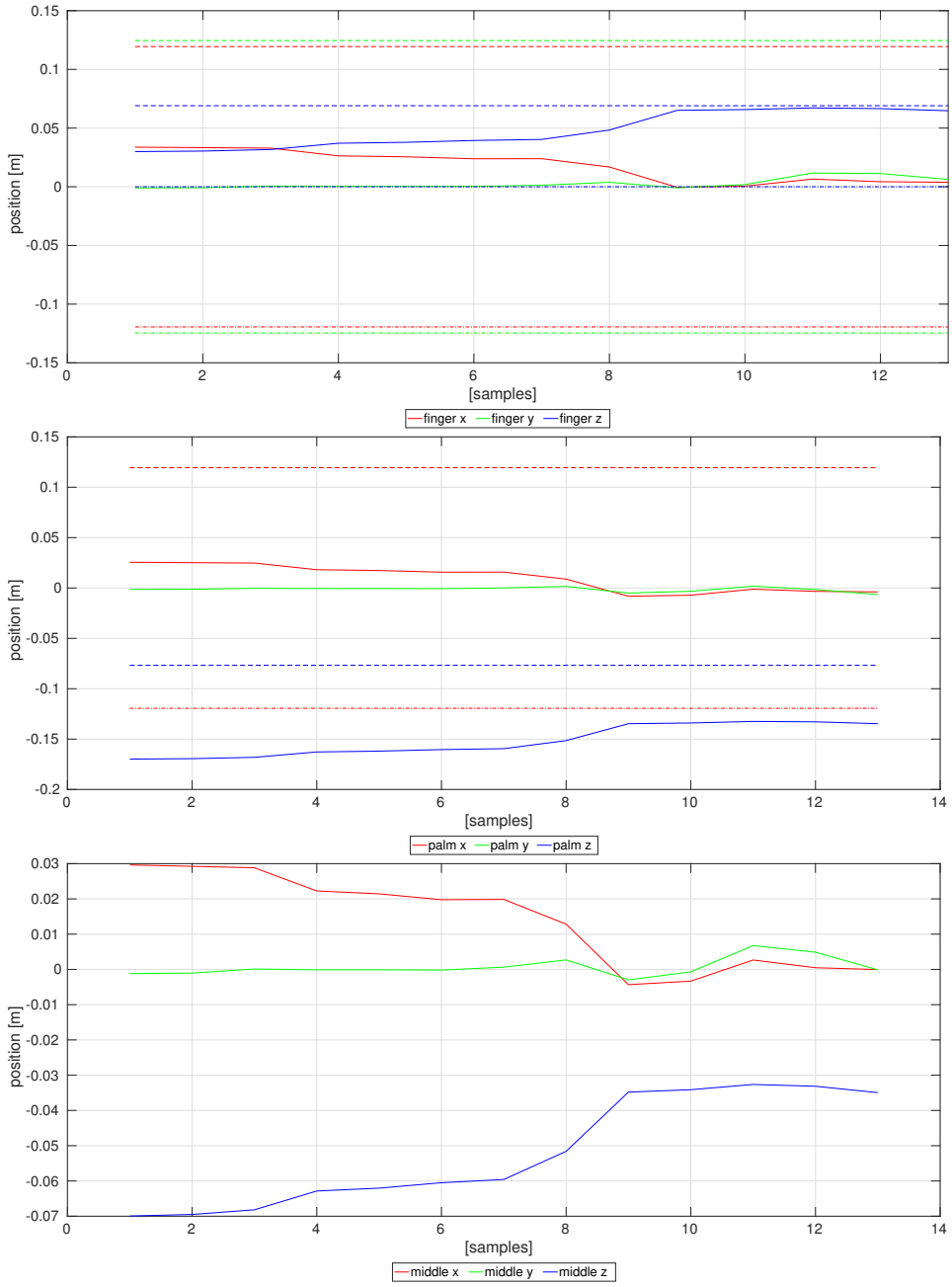


Figure 4.34: Trajectory followed by the finger-effector (top), palm-effector (middle) and middle-effector (bottom) during the third phase of the second iteration of the Third Experiment. Upper and lower limits described with dashed lines.

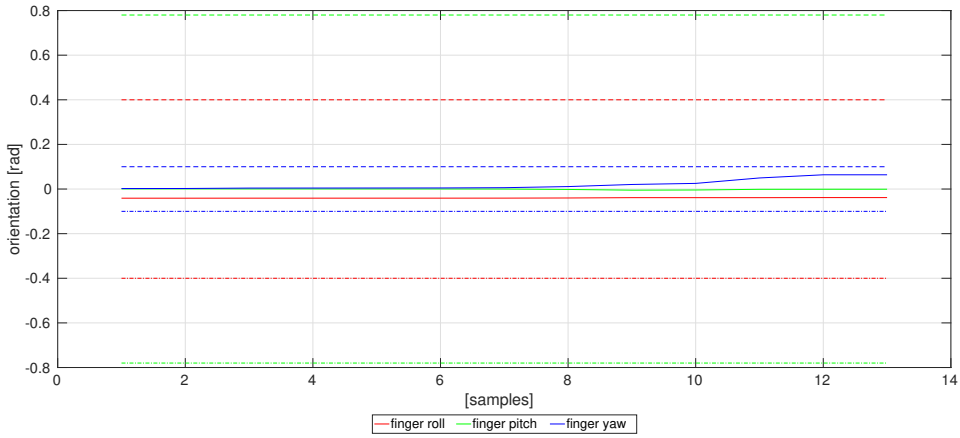


Figure 4.35: Orientation reached by the end-effectors during the third phase of the second iteration of the Third Experiment. Upper and lower limits described with dashed lines.

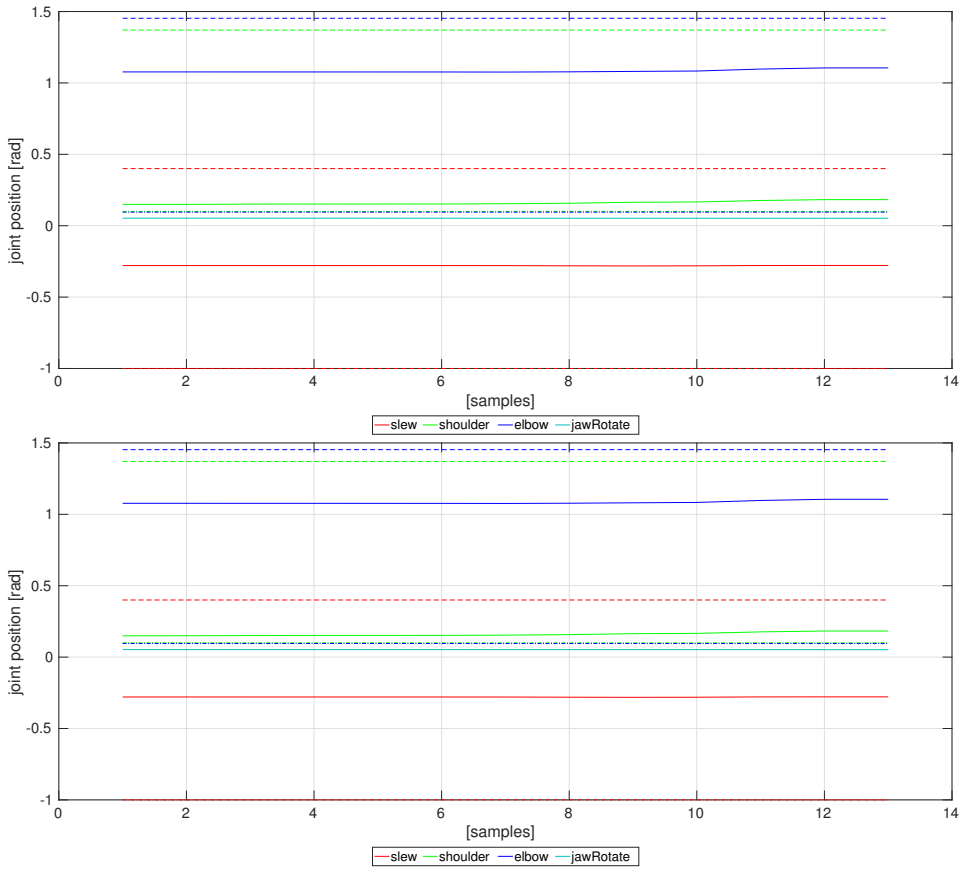


Figure 4.36: Arm (top) and vehicle (bottom) joint values during the third phase of the second iteration of the Third Experiment. Upper and lower limits described with dashed lines.

4.5 Conclusions

Once the scene has been reconstructed, the next step consists on separating the object of interest from the rest of the scene and calculating a grasping pose that ensure a robust grasp.

A grasp planning framework that make it possible has been introduced in this chapter. First, the framework takes as input the point cloud of the scene, detects the objects placed there and estimates their sizes. The object that better fits with a series of predefined conditions is selected.

Then, a grasp planning algorithm, based on the Task-Priority algorithm, is in charge of calculating the grasping pose that fulfil in a greater way a hierarchy of tasks and that strictly accomplishes a set of constraints that depend on the object and gripper sizes and the configuration of the robotic system. To do so, it simulates the movements that the system would perform while it is trying to reach a valid pose.

Three experiments have proven the correct functionality of the framework. In the first two experiments, the framework is able to calculate a valid grasping pose reachable without moving the vehicle. This avoids the higher inaccuracies produced by the vehicle displacements. The object of the third experiment is not reachable without moving the vehicle, but the framework is also able to overcome this problem and find a solution by allowing vehicle movements during the simulation.

Chapter 5

Grasp Execution

This chapter presents a methodology for grasping an object taking as input its grasping pose. The methodology uses the Multi-Task Priority Framework, explained in the Chapter 3, for deciding the movements the system must do to reach the desired pose.

The grasping process has been divided in four phases. The first phase moves the end-effector to a pose that facilitate the posterior approximation movements. In the second phase the end-effector reaches the grasping pose following the approximation direction previously defined by the Grasp Planner. In the third phase the object is grasped. And finally, the system is able to recover the object. For each one of the phases, different tasks and constraints are defined that guarantee the correct execution of the grasping.

5.1 Motivation

After the object of interest has been detected and the grasping pose calculated, the grasping must be executed. To do so, it is not a good idea to move the end-effector directly to this pose, since the fingers of the gripper could collide with the object when they are reaching their final pose. Thus, a methodology that guarantee that the grasping pose is reached following a proper approaching direction is needed.

Moreover, I-AUVs have joint configurations that favour their performance. For example, keeping the center of masses of the arm align with the center of masses of the vehicle usually improve the stability control of the vehicle. As a consequence, if the system is able to perform the grasping keeping theses configurations, the likelihood of success increases.

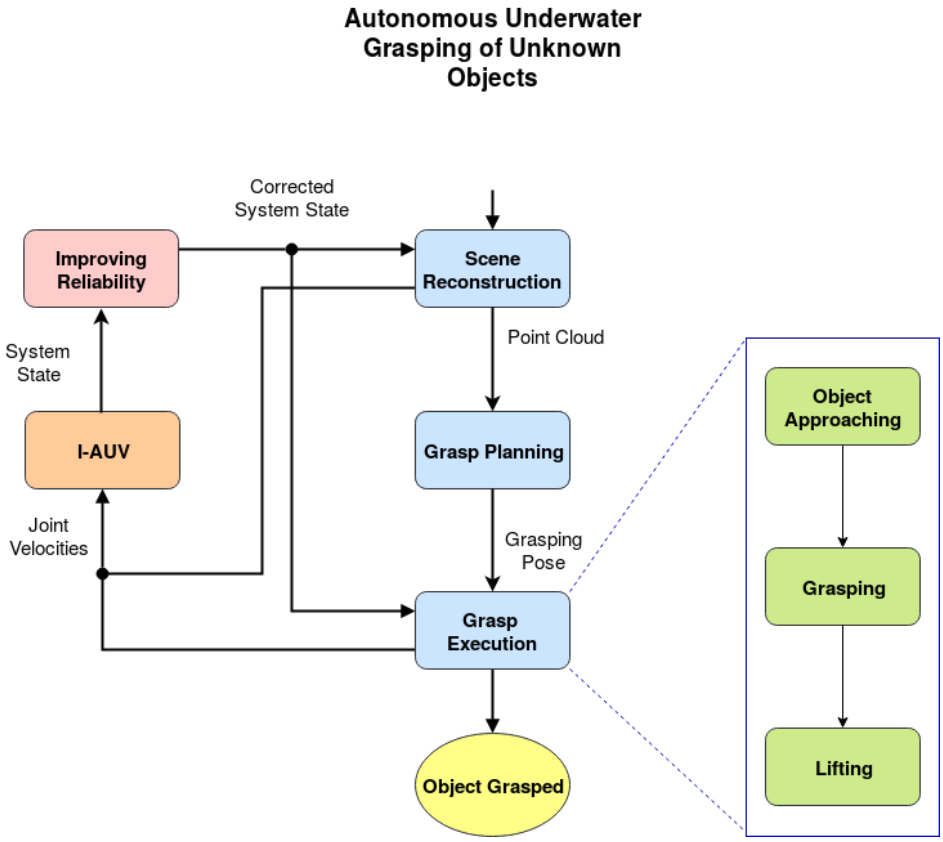


Figure 5.1: Flowchart of an autonomous underwater grasping intervention. The third of the phases of the intervention consists on executing the grasp. This phase is composed by three subphases (green). First, the gripper is of the manipulator is guided to a position close to the object, then it is conducted to the grasping pose and the gripper is closed and finally, the object is lifted from the floor.

5.2 Methodology

This section details the methodology followed for executing an object grasping once the grasping pose has been planned. This methodology guide the end-effector of the system to the final position respecting the approaching direction defined by the grasping frame. Moreover, it uses the Multi-Task Priority algorithm for calculating in each moment the velocities that have to be sent to each of the system joints. Thus, it permits the definition of several hierarchical tasks for each of the methodology phases. These tasks are in charge of achieving the main goal but respecting, as far as possible, the configuration system preferences.

The methodology has been separated in four phases that are explained hereinafter.

5.2.1 Pre-Grasping Pose

The goal of the first phase is to conduct the end-effector to a position that facilitate the posterior approaching movements. If the end-effector goes directly to the final pose, it is quite probable that the fingers collide with the object during the trajectory.

To avoid that, a pre-grasping pose is defined. This pose has the same orientation than the grasping pose but it is displaced in the $-z$ direction a distance large enough to avoid that any part of the gripper is in contact or around the object when it is reached. It is worth remembering that the z component of the grasping pose defines the approaching direction.

Supposing that the grasping pose is considered the origin, and the distance between the pre-grasping and the grasping poses is d. The tasks defined for this phase can be seen in the Table 5.1.

Table 5.1: Tasks defined for reaching the pre-grasping pose

Priority	**Task**
1	$x, y = 0$
	$z = -d$
2	$roll, pitch, yaw = 0$
3	$FavourableConfiguration$

For this phase, the tasks with highest priority try to reach the position defined by the pre-grasp pose. In the second stage appear the tasks that

orient the end-effector properly. Apart from these needed tasks, optional tasks can be added with lower priorities. These tasks intend to keep a system configuration that favours the grasping performance.

This phase finishes when the velocities returned by the Multi-Task Priority Algorithm are lower than a certain value. If in that moment, the end-effector is not close enough to the pre-grasp position, the intervention is cancelled. Otherwise, next phase starts.

5.2.2 Grasping Pose

During the second phase of the methodology, the end-effector is guided to the grasping pose. The definition of the pre-grasping position displaced in the $-Z$ direction, enforce the system to move the end-effector following the approaching direction. If the grasping pose has been properly defined, this direction will avoid the collision between the gripper fingers and the object during the system movements.

The tasks of this phase are detailed in the Table 5.2.

Table 5.2: Tasks defined for reaching the grasping pose

Priority	**Task**
1	$x, y, z = 0$
	$roll, pitch, yaw = 0$
2	$FavourableConfiguration$

In this case, it is as important the orientation as the position, thus all the tasks that conduce the end-effector to the grasping pose, are in the same level. As in the last phase, the lowest priority level is used to keep an appropriated system configuration.

When the algorithm returns low velocities the phase finishes. If the end-effector is close enough to the correct pose, the methodology can continue.

5.2.3 Grasping

In the third phase, the gripper must close its fingers until the object is properly grasped, at the same time that the I-AUV is keeping the current end-effector pose.

It is assumed that, when the gripper is completely closed, its joint value is 0, the tasks defined for this phase are described in the Table 5.3.

Table 5.3: Tasks defined for grasping the object.

Priority	Task
1	$gripperOpening = 0$
2	$x, y, z = 0$ $roll, pitch, yaw = 0$
3	$FavourableConfiguration$

The task with higher priority tries to close the gripper. Moreover, in the second level, the system is trying to keep the end-effector in the grasping pose. Last, in the third level, a good system configuration is maintained.

This phase finishes when the system detects that the object has been robustly grasped.

5.2.4 Lifting

The last phase of the methodology starts when the object of interest has already been grasped. In this phase, the object is moved away from the sea floor.

To do so, the tasks described in the Table 5.4 are used.

Table 5.4: Tasks defined for grasping the object.

Priority	Task
1	$gripperOpening = currentValue$
2	$world_z = currentValue - d$

The gripper must keep its current opening value in order to continue applying the force over the object and not to loose it. In the second level, the end-effector moves up a predefined value, d, in order to separate the object from the floor and being able to perform next movements avoiding possible collisions with the floor.

At the end of this phase, the intervention has finished and the I-AUV can go back to its home position with the object grasped.

5.3 Experimental Results

Three experiments have proven the suitability of the methodology. The manipulator used in the three experiments is the ECA-CSIP Light weight ARM5E [Fernández et al., 2013]. This manipulator is composed by 4 joints (*slew*, *shoulder*, *elbow* and *jawRotate*) plus an extra joint for closing and opening the gripper (*jawOpening*). In the first two experiments, the arm have been attached to a fixed platform inside a water tank. This platform substitute a vehicle performing station keeping. In the last experiment, the manipulator is attached to the Girona 500 AUV [Ribas et al., 2012], which has 4 degrees of freedom (x, y, z, *yaw*).

In order to detect when the gripper has grasped the object with enough force, the current used by the arm is monitored. Thus, during the grasping phase, when the current reaches a peak higher than a certain value, the gripper stops closing and the phase finishes.

5.3.1 Favourable Configuration

A series of tasks have been defined for keeping a favourable system configuration during the three experiments ($slew = 0$, $shoulder = 1$ and $elbow = 0.7$).

If the *slew* joint is at its 0 value, the arm is placed aligned to the vehicle, thus, the center of masses of the arm and vehicle are also aligned. This position favours the stability of the vehicle when performing movements.

On the other side, it is preferable to avoid that the joints reach their limits, since an error detecting these limits could produce motors overheating. Another handicap is that the used manipulator is more accurate and performs better when the joints are far from their limits. For that reasons, two tasks that tries to keep the *shoulder* and *elbow* joint in the middle of their ranges have been defined.

These tasks have been placed in the hierarchy level reserved for the $FavourableConfiguration$ during the experiments.

5.3.2 First Experiment

The object grasped in the First Experiment is an amphora. This experiment continues the $FirstExperiment$ described in the Chapter 4. In that experiment, the planner returned a grasping pose, and assured that it could be reached moving only the manipulator.

The grasping pose given by the planner with respect to the arm base is ($x = 0.1428$, $y = -0.1279$, $z = 0.9724$, $roll = 0.0226$, $pitch = 0.2122$, $yaw = 2.5180$). And, the pre-grasping pose have been defined at $15cm$ from the grasping pose in the $-Z$ direction.

Three stopping criteria have been defined for this experiment:

- The sum of the velocities returned by the Task Priority Algorithm is lower than $0.001 rad/s$.

- The current position of the end-effector is closer than $0.5cm$ from the goal position and the angle between the current and the desired orientation is lower than $0.1rad$.

- In the grasping phase, if the current used by the manipulator surpass $1.2A$.

When a phase finishes, the decision of continuing or cancelling the intervention depends of the phase.

- At the end of the first phase, in order to continue, the distance between the current position of the end-effector and the desired position must be lower than $3cm$.

- In the second phase, it is as important the position as the orientation, and also more accuracy is needed. Thus, the maximum allowed error is $0.1cm$ regarding the position and $0.05rad$ regarding the orientation.

- The success of the third phase depends on whether the object has been grasped or not.

- At the end of the last phase, the I-AUV always goes back to the home position.

Three figures show the results of the experiment. The Fig. 5.2 represents the position and orientation of the end-effector with respect to the grasping pose. The Fig. 5.3 presents the arm joint values. And the Fig. 5.4 displays the current used by the manipulator.

Four dashed lines divide the figures in the four phases of the methodology plus one initial phase that has as goal to open the gripper and get the manipulator ready to start with the methodology.

The first phase tries to move the end-effector to the pre-grasping pose. As can be seen in the Fig. 5.2, this pose cannot be completely reached. The correct position has been achieved but the component *pitch* has not been able to finish close to the 0 value. It means that this phase has finished because the Task Priority Algorithm has not found any movement that improve the current state and the velocities sent are lower than the predefined value. In spite of that, the final error is lower enough to continue with the next phase.

In the next phase, the Grasping Pose is the goal. As the Grasp Planner predicted, this pose is reachable moving only the arm. Thus, this phase

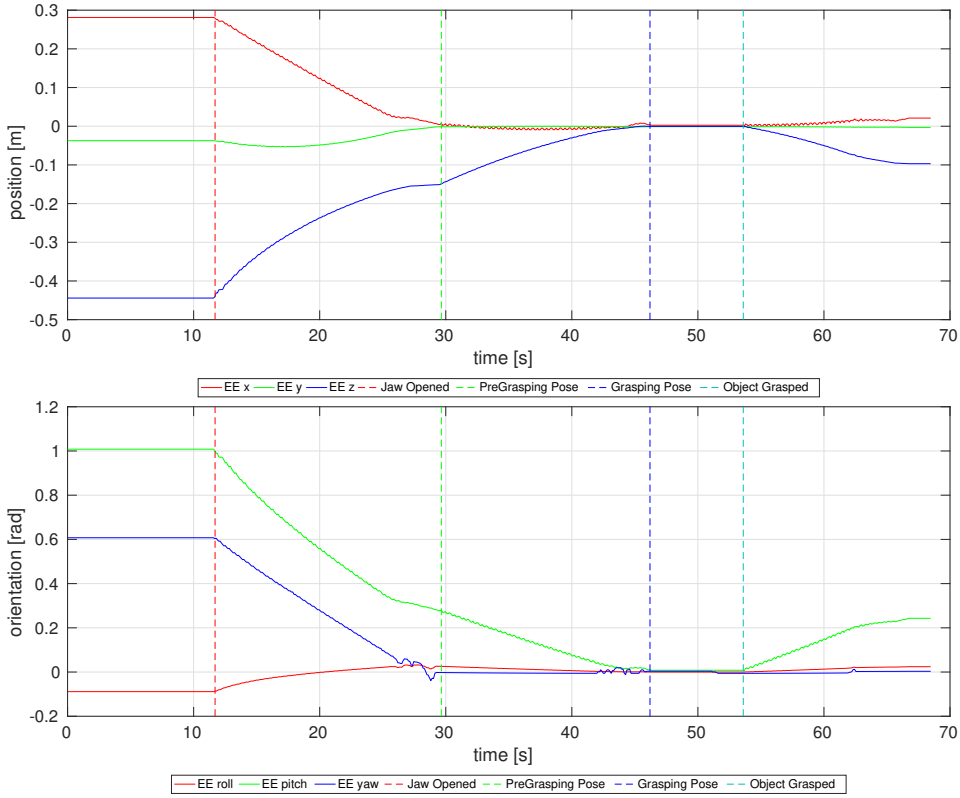


Figure 5.2: Position and Orientation of the end-effector with respect to the grasping pose during the First Experiment.

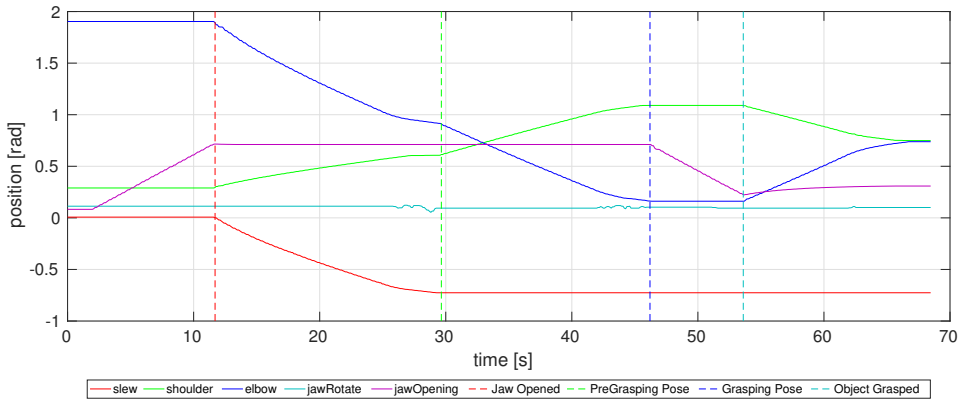


Figure 5.3: Arm Joint values during the First Experiment.

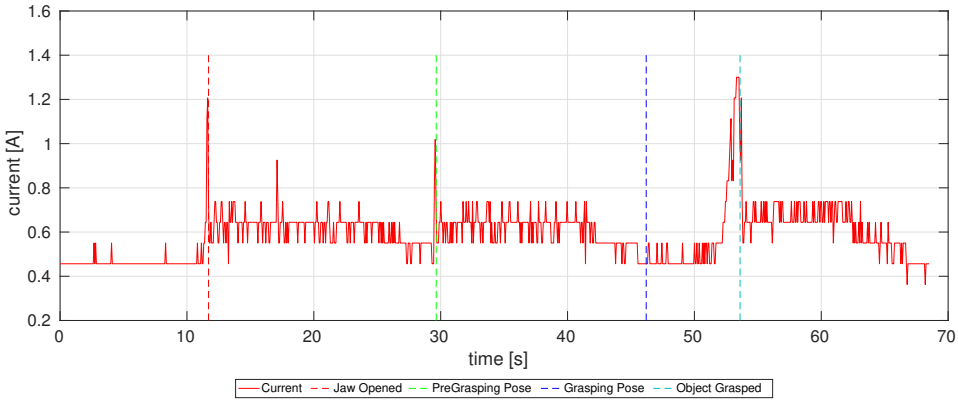


Figure 5.4: Current used by the Manipulator during the First Experiment.

finishes when the positional and orientation error are lower than the defined limits.

In the Figs. 5.3 and 5.4, it is easy to see how during the Grasping phase, the *jawOpening* joint is going to 0 until the current surpass the threshold. As the current limit has been reached before the gripper is completely closed, it means that the object has been grasped.

Finally, the arm separates the object $10cm$ from the tank floor. And the methodology concludes.

5.3.3 Second Experiment

The Second Experiment takes as input the pose calculated in the $SecondExperiment$ explained in the last chapter. In this case the object of interest is a black-box mock-up.

The planner also determines that it can be grasped using only the manipulator. The grasping pose given by the planner with respect to the arm base is ($x = -0.2753$, $y = 0.011$, $z = 0.7004$, $roll = -0.0316$, $pitch = 1.2007$, $yaw = 3.0674$). And as in the last experiment, the pre-grasping pose have been defined at $15cm$ from the grasping pose in the $-Z$ direction.

In this experiment the stopping criteria and the decisions for continuing with or cancelling the methodology are the same than in the First experiment.

Fig. 5.5 shows the position and orientation followed by the end-effector during the experiment, Fig. 5.6 the joint values and Fig. 5.7 the current consumed by the arm.

As in the last experiment, after opening the gripper, the Pre-Grasping Pose must be reached. The end-effector is again able to reach the correct

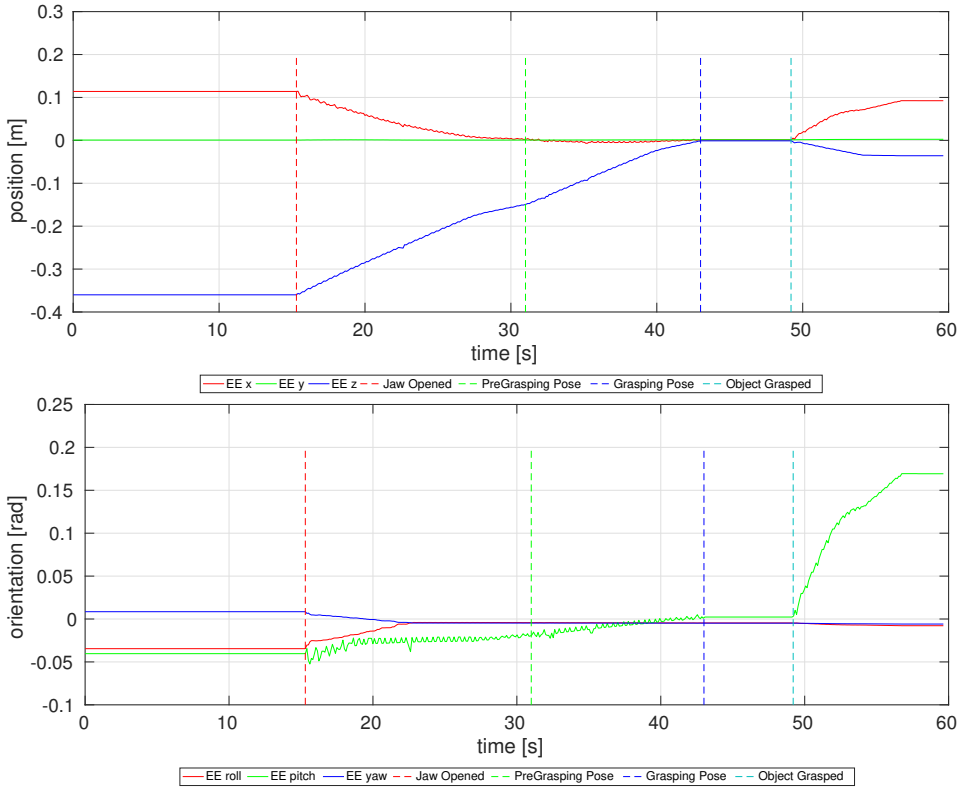


Figure 5.5: Position and Orientation of the end-effector with respect to the grasping pose during the Second Experiment.

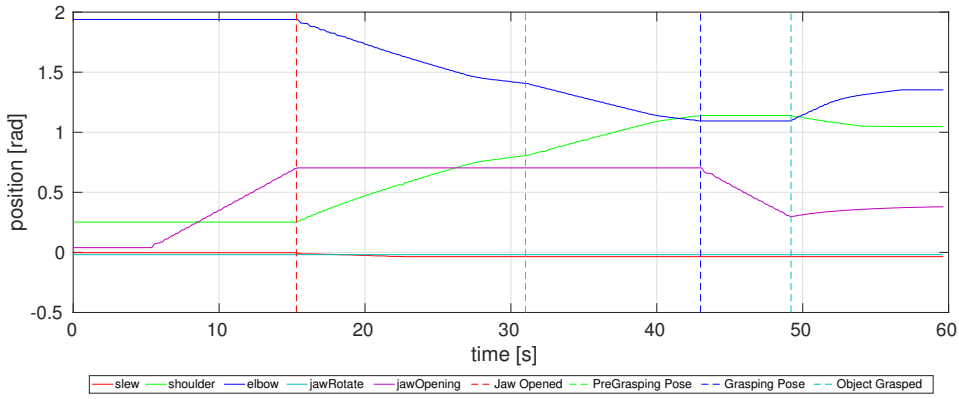


Figure 5.6: Arm Joint values during the Second Experiment.

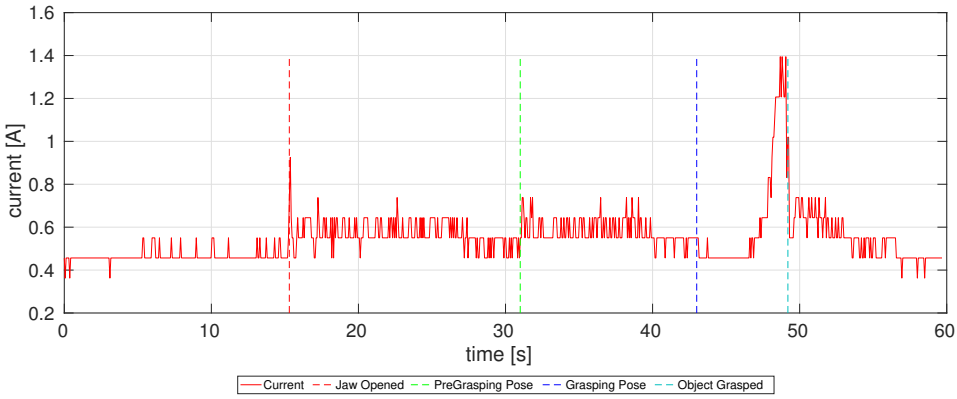


Figure 5.7: Current used by the Manipulator during the Second Experiment.

Pre-Grasping Position and only the *pitch* component of the orientation is a bit displaced from its desired value. Thus, the algorithm can continues with the next phase.

Once again, as can be seen in the Fig. 5.5, the prediction of the grasp planner is good and the grasping pose can be reached using only the manipulator.

After arriving at the Grasping Pose, the manipulator starts closing the gripper. At the moment that the fingers contact with the object with enough force, the arm consume more than $1.2A$ and, at that moment the current phase finishes.

Finally, the object is lifted $10cm$ from the tank floor and the algorithm ends.

5.3.4 Third Experiment

The third experiment has been performed in the Sant Feliu harbour (Girona, Spain) in 2017 during the experiments of the MERBOTS Project, coordinated by the Jaume I University and including the University of Girona and the University of the Balearic Islands.

In this experiment, the object of interest is an amphora placed in the sea floor (see Fig. 5.8). The planner detects that the object cannot be reached using only the manipulator. Thus, the planner returns the grasping pose and also indicates that the vehicle must be used in the experiment.

The grasping pose given by the planner is ($x = 0.598$, $y = 0.019$, $z = 1.458$, $roll = -0.452$, $pitch = 0.029$, $yaw = 1.748$). And the pre-grasping pose has been placed $30cm$ from the grasping pose in $-Z$ direction.

The stopping criteria defined for this experiment are:

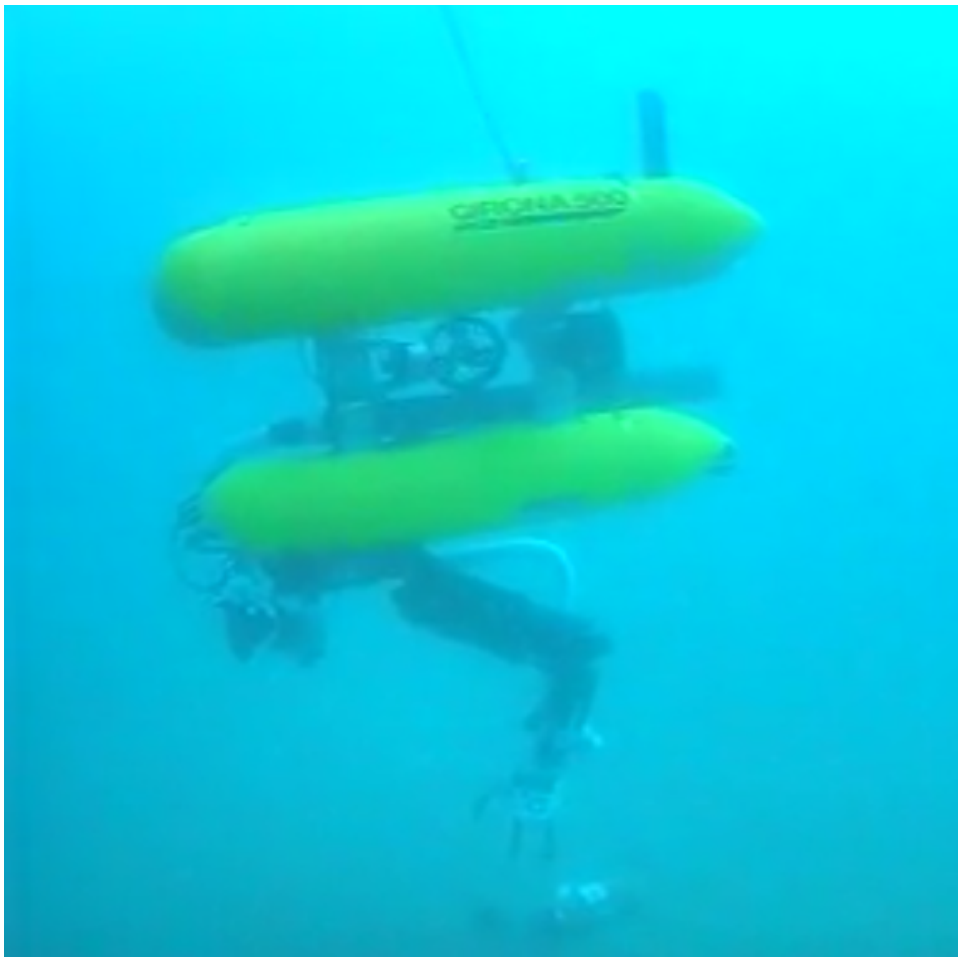


Figure 5.8: Experimental setup for the MERBOTS Project trial.

The use of the vehicle affects the definition of the stopping criteria and the range for advancing to the next phase. Usually the vehicles are less accurate than manipulators and take longer to stabilize in a position. Thus, the defined limits must be a bit more permissive but without affecting the result of the intervention.

The stopping criteria defined for the experiment are:

- The sum of the velocities returned by the Task Priority Algorithm is lower than $0.01 rad/s$.
- The current position of the end-effector is closer than $3cm$ from the goal position and the angle between the current and the desired orientation is lower than $0.15 rad$ during 5 iterations of the Task Priority Algorithm.
- In the grasping phase, if the current used by the manipulator surpass $4A$.

And the decision of continuing or cancelling depends on:

- At the end of the first phase, in order to continue, the distance between the current position of the end-effector and the desired position must be lower than $7cm$.
- In the second phase, the maximum allowed error is $3cm$ regarding the position and $0.15 rad$ regarding the orientation.
- The success of the third phase depends on whether the object has been grasped or not.
- At the end of the last phase, the I-AUV always goes back to the home position.

In the Fig. 5.9 the position and orientation traced by the end-effector with respect to the Grasping Pose is showed. Fig.5.10 represents the vehicle position and orientation during the experiment. Fig 5.11 the joint values. And Fig. 5.12 the current consumed by the arm.

At the begging of this experiment, the gripper is already open, thus, the first phase of the methodology starts directly.

During the first phase, the end-effector is guided to the Pre-grasping Pose. As was previously said, in order to conclude each phase, it is needed to reach a concrete pose, but also to keep in range during a some iterations of the Task Priority Algorithm. It is possible to see in the Fig. 5.2, how the end-effector oscillate converging to a desired pose, until it achieves to stabilize during a period of time. Once it happens, the next phase can start.

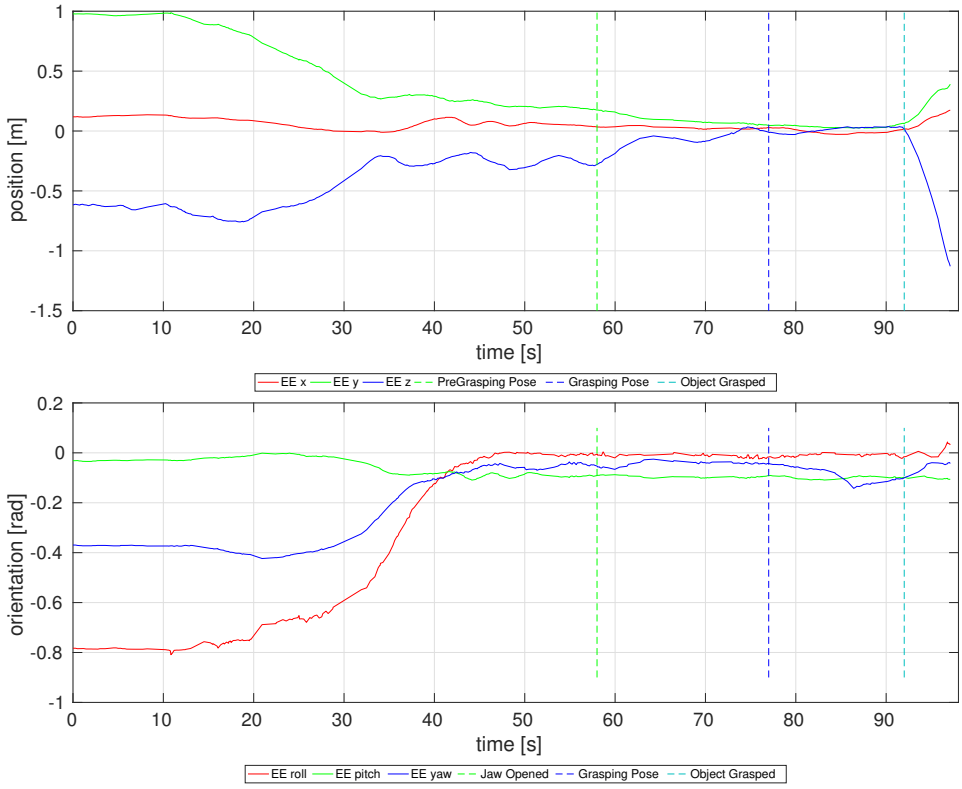


Figure 5.9: Position and Orientation of the end-effector with respect to the grasping pose during the Third Experiment.

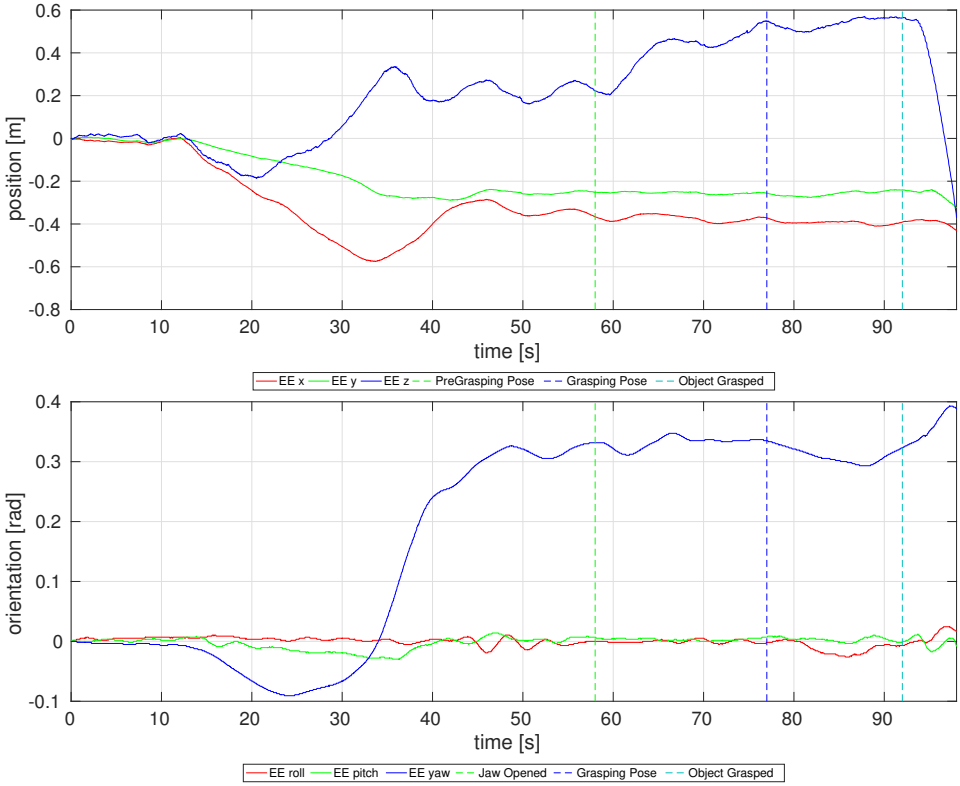


Figure 5.10: Position and Orientation of the vehicle during the Third Experiment.

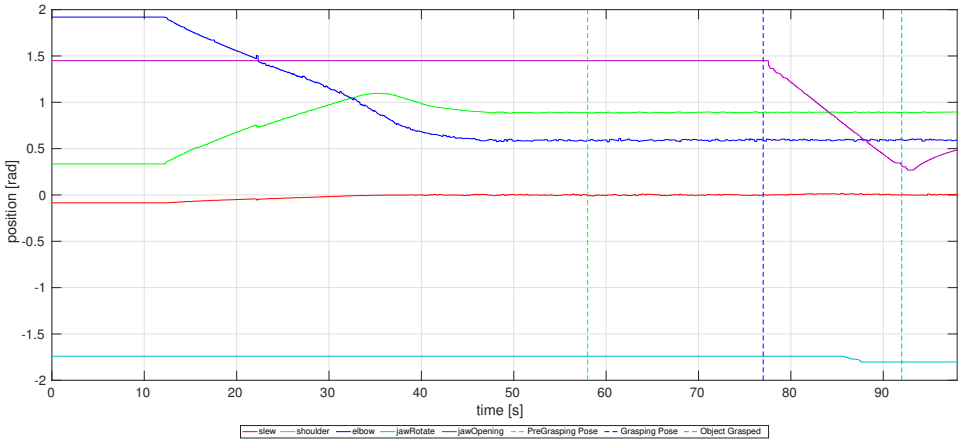


Figure 5.11: Arm Joint values during the Third Experiment.

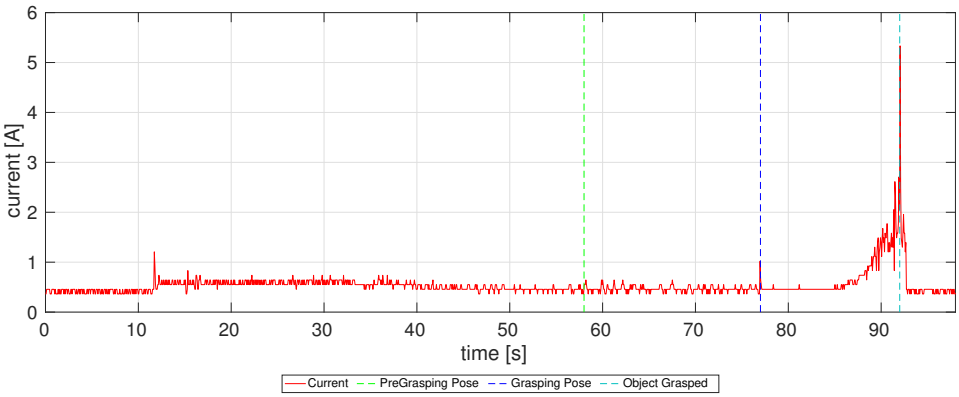


Figure 5.12: Current used by the Manipulator during the Third Experiment.

In the second phase, the end-effector must reach the Grasping Pose. In this phase, the end-effector spends less time stabilizing. This is due to the goal and the current pose are closer. Thus the velocities returned by the Task Priority Algorithm are lower. Once the end-effector is in the allowed range during time enough, the current phase finishes.

In the third phase, the gripper is closing at the same time that the robot is trying to keep the end-effector as close as possible to the Grasping Pose. When the current consumed by the manipulator reach the defined threshold, the current phase finishes.

Last, the robot must separate the amphora $1m$ from the sea floor, then the intervention concludes.

It is also remarkable how the tasks in charge of keeping a favourable configuration have actuated in this experiment. Fig. 5.11 shows the *slew* joint, which did not start parallel to the center of masses of the vehicle, it is at its 0 value, has been positioned at this value and has been maintained there during the whole experiment. This has been possible thanks to the degrees of freedom of the vehicle which allow to perform the defined tasks without having to use the *slew* joint.

5.4 Conclusions

In this chapter, a methodology based on the Task Priority Algorithm for grasping an object taking as input its grasping pose has been presented.

This methodology has been divided in four phases. This phases allows the end-effector to reach the pose defined as Grasping Pose, following the approximation direction that this pose defines. And, grasping the object detecting when it has been grasped with enough force and separating the

object from the floor before the robot can go back to its home position with the object grasped.

Three experiments have proven the suitability of the methodology. In the first two experiments an amphora and a black box mock up have been respectively grasped using only arm movements. The last experiment shows the results of the MERBOTS Project, where the methodology was used to recover an amphora from a harbour using both a manipulator and a vehicle.

Chapter 6

Toolbox for Improving Reliability

6.1 Introduction

This chapter presents two methodologies for detecting problems during an intervention and correct them.

First, an algorithm for controlling the position of the end-effector during the intervention has been described. A camera attached to the base of the manipulator records the scene, and each time a marker placed at the gripper appears at the scene its pose is estimated. This pose is used to calculate the position and orientation of the end effector in each moment. After that, the Inverse Kinematics Algorithm gives the real state of the arm joints. In case the internal value of the joints differs from the real ones, they are updated.

Secondly, a methodology that detects collisions during the approaching phase and correct the trajectory in case they appear is detailed. For this methodology, a force-torque sensor has been attached at the wrist of the gripper. During the approaching phase, the values returned by the sensor are read. If a contact is detected, according to the current position of the end-effector and the direction of the force detected, the trajectory is modified to continue with the intervention.

Both methodologies have been successfully tested in real experiments.

Moreover, the first algorithm have been published in an JCR journal. Concretely in the Annual Reviews in Control Journal [Peñalver et al., 2015b].

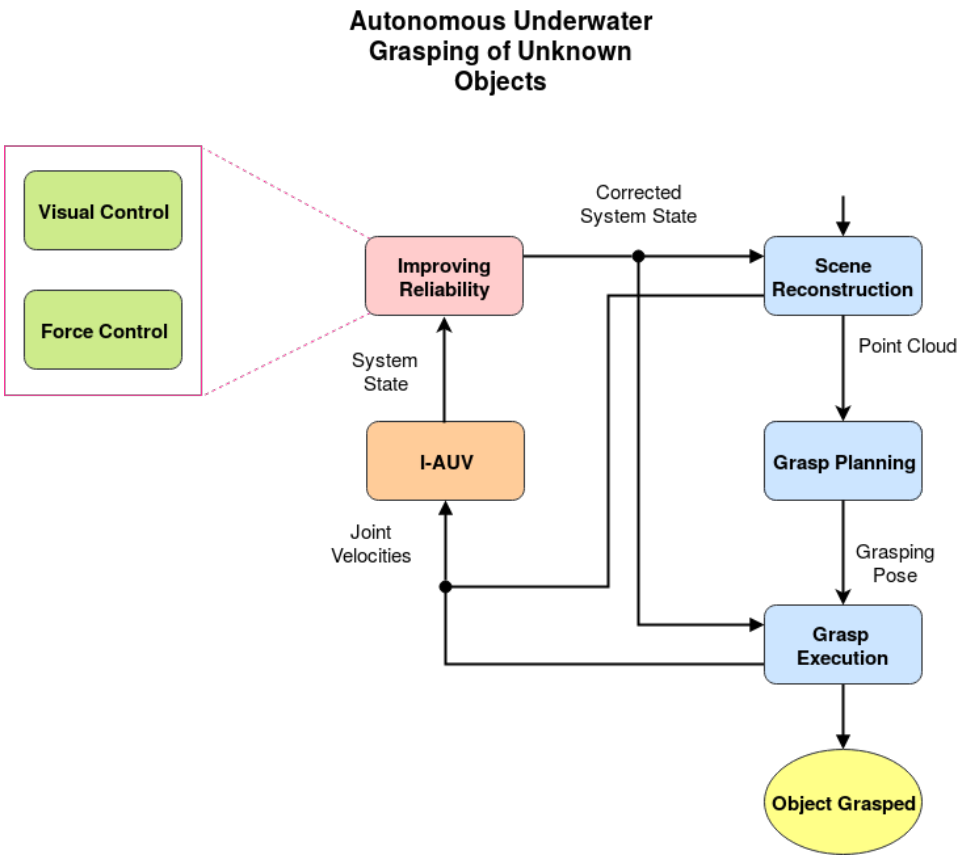


Figure 6.1: Flowchart of an autonomous underwater grasping intervention. Whilst the I-AUV is moving, two tasks are in charge of controlling that the movements are appropriate and correct them in case it is needed (green). The first task visually monitors the trajectory of the manipulator and the second one detects collisions using a force/torque sensor.

6.2 Motivation

It is not unusual for kinematics errors to occur in robotic arms due to bad initializations or miscalibration of the joints.

The initialization of the Light-Weight ARM5E robotic arm [Fernández et al., 2013] consists on guiding the joints to their limits at the same time that the consumed current is monitored. When a current surpasses a defined value, the arms stops and the current position of the joint is saved. The saved values define the initial position of each of the joints. The problem is that these positions are not always exactly the same, producing uncertainty obtaining the position and orientation of the end-effector.

Moreover, the arm sometimes suffer from joint miscalibration, specially when they are moving close to the joint limits.

The errors produced by the bad initialization are quite smalls compared with the ones produced by the joint miscalibration. But even an small error in one of the first joints, can mean a big error at the end-effector position.

For that reason, if an intervention requires accurate movements of the manipulator, the use of methodologies to correct the possible errors that could appear, is crucial.

6.3 Visually Kinematic Controller

A new algorithm has been developed to control, in an autonomous way, that the movements performed by the Light-Weight ARM5E robotic arm and correct them during an intervention.

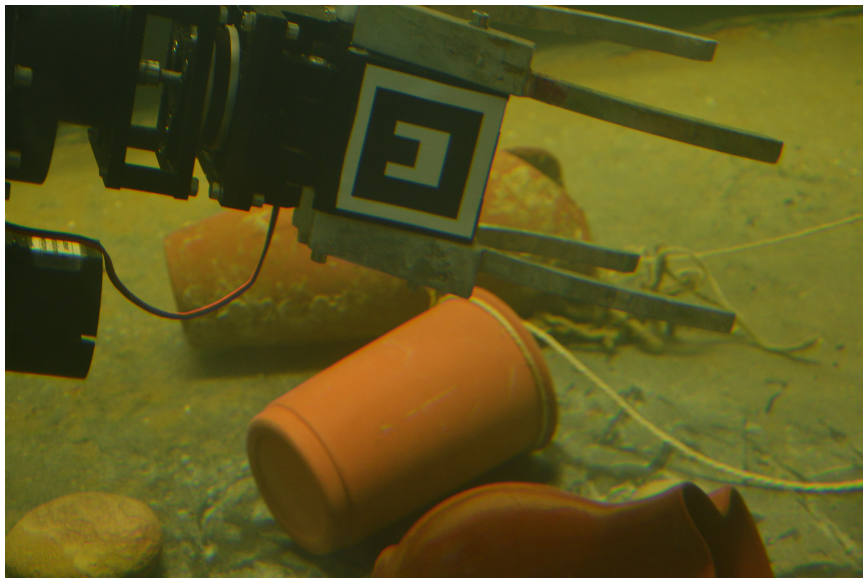
The algorithm calculates the position and orientation of the end-effector with respect to the base of the arm by estimating the pose of a known marker placed at the gripper of the manipulator.

This algorithm is used to automatically calculate the extrinsic parameters of the camera with respect to the base of the arm at the beginning of an intervention. And to update the values of each joint of the robotic arm.

6.3.1 Marker Detection

An AR marker is a white figure with a known size drawn over a black square (see Fig. 6.2). As have been said before, this marker is attached to known place of the robotic arm, in this case at the gripper. This marker is detected using the ARToolkit library [Kato and Billinghurst, 1999]. Despite ARToolKit is a software library for building AR applications, the library provides multiple methods for detecting and localizing the position and orientation of the marker.

Figure 6.2: AR marker attached at the gripper of the ARM5E manipulator.



6.3.2 Transformation between the Camera and the End-Effector

Once the marker has been detected, the previous algorithms provide its pose with respect to the camera (cMm, which is the homogeneous matrix that represents the relationship between two frames, in this cases the camera c and the marker m). Therefore, the next step consists in obtaining the transformation between the camera and the end-effector of the arm (cMe). For this, it just needed to multiply the homogeneous matrix between the camera and the marker (cMm) by the relationship between the marker and the end-effector (mMe), which must be known:

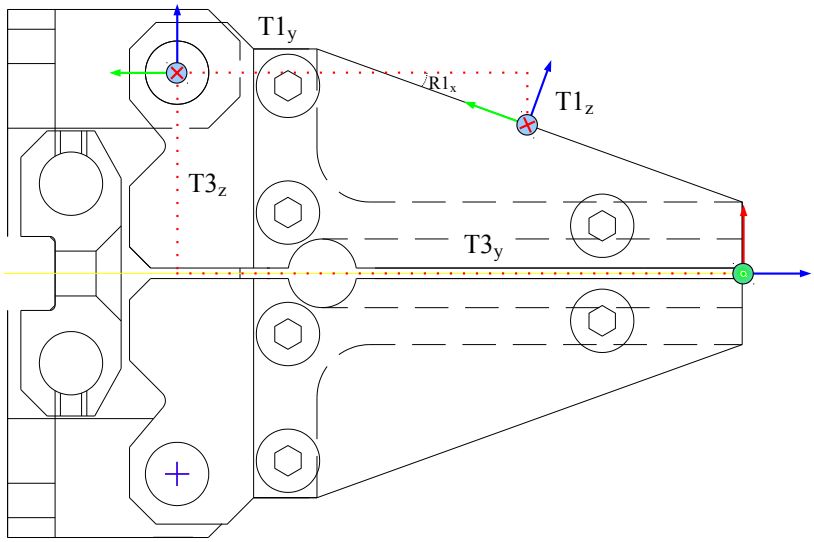
$$\mathrm{cMe} = \mathrm{cMm} * \mathrm{mMe}$$

For this experiment, the markers has been placed on the top of the gripper and the end-effector of this arm is in the tip of this gripper. Thus, the transformation between the markers and the end-effector, depends, in addition to the static measures (see Fig. 6.3), on the opening of the gripper at each moment.

The relationship between the marker and the end-effector for this gripper, is detailed as follows:

- The first transformation is a rotation around the x axis (R1x) in order to place the marker frame parallel to the palm of the gripper followed by a translation about the y (T1y) and z (T1z) axis in order to place

Figure 6.3: Static transformations between the marker and the end-effector.



the frame in the joint of the wrist of the gripper which allows the arm open and close the gripper (mMw):

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\mathrm{R1}_x) & -\sin(\mathrm{R1}_x) & \mathrm{T1}_y \cos(\mathrm{R1}_x) - \mathrm{T1}_z \sin(\mathrm{R1}_x) \\ 0 & \sin(\mathrm{R1}_x) & \cos(\mathrm{R1}_x) & \mathrm{T1}_z \cos(\mathrm{R1}_x) + \mathrm{T1}_y \sin(\mathrm{R1}_x) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- The second transformation is a rotation around the x axis, which depends on the opening α of the hand in each detection of the marker:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- The next transformation is a translation about the y (T3y) and z

(T3z) axis in order to place the frame in the position of the end-effector (wMe):

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \mathrm{T3}_y \\ 0 & 0 & 1 & \mathrm{T3}_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Finally, two rotations around the x (R4x) and z (R4z) axis are needed to orientate the frame like the end-effector frame:

$$\begin{pmatrix} \cos(\mathrm{R4}_z) & -\sin(\mathrm{R4}_z) & 0 & 0 \\ \cos(\mathrm{R4}_x)\ \sin(\mathrm{R4}_z) & \cos(\mathrm{R4}_x)\ \cos(\mathrm{R4}_z) & -\sin(\mathrm{R4}_x) & 0 \\ \sin(\mathrm{R4}_x)\ \sin(\mathrm{R4}_z) & \cos(\mathrm{R4}_z)\ \sin(\mathrm{R4}_x) & \cos(\mathrm{R4}_x) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

As a result, the transformation between the marker and the end_effector (mMe) is:

$$\begin{pmatrix} \cos(\mathrm{R4}_z) & -\sin(\mathrm{R4}_z) & 0 & 0 \\ \sigma_2\ \sin(\mathrm{R4}_z) & \cos(\mathrm{R4}_z)\ \sigma_2 & -\sigma_1 & \sigma_3 \\ \sin(\mathrm{R4}_z)\ \sigma_1 & \cos(\mathrm{R4}_z)\ \sigma_1 & \sigma_2 & \sigma_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where:

$$\begin{aligned} \sigma_1 &= \sin(\alpha + \mathrm{R1}_x + \mathrm{R4}_x) \\ \sigma_2 &= \cos(\alpha + \mathrm{R1}_x + \mathrm{R4}_x) \\ \sigma_3 &= \mathrm{T3}_y\ \cos(\alpha + \mathrm{R1}_x) - \mathrm{T3}_z\ \sin(\alpha + \mathrm{R1}_x) + \\ &\quad \mathrm{T1}_y\ \cos(\mathrm{R1}_x) - \mathrm{T1}_z\ \sin(\mathrm{R1}_x) \\ \sigma_4 &= \mathrm{T3}_z\ \cos(\alpha + \mathrm{R1}_x) + \mathrm{T3}_y\ \sin(\alpha + \mathrm{R1}_x) + \\ &\quad \mathrm{T1}_z\ \cos(\mathrm{R1}_x) + \mathrm{T1}_y\ \sin(\mathrm{R1}_x) \end{aligned}$$

6.3.3 Transformation between the base of the arm and the camera

In order to obtain the relationship between the base of the arm and the camera (bMc), the arm is moved to a predefined configuration that allows

the camera to see clearly the marker. At this moment, the arm waits until the camera estimates the position and orientation of the marker and the detailed algorithm calculates the transformation between the camera and the end-effector (cMe) at this moment.

On the other hand, the relationship between the base of the arm and the end-effector (bMe) is calculated by means of the direct kinematics of the arm at this moment. Once these two matrices (cMe and bMe) have been obtained, in order to calculate the transformation between the base of the arm and the camera (bMc), just a product operation is needed:

$$\mathrm{bMc} = \mathrm{bMe} * (\mathrm{cMe})^{-1}$$

This part of the algorithm must be done preferably in a moment when the user is sure that the arm is well calibrated, for example just after the initialization, because the matrix obtained will be used in the next steps and this matrix depends directly on the values of the joints.

6.3.4 Updating the joints

Once the process of initialization of the algorithm has been done, the camera will be used to continuously detects the marker. For each detection, the following steps are performed in order to obtain the real value of each joint and update them:

1. Calculate the relationship between the base of the arm and the end-effector (bMe) at this moment, using the detection of the marker (cMe) and the values calculated in the initialization of the algorithm (bMc):

$$\mathrm{bMe} = \mathrm{bMc} * \mathrm{cMe}$$

2. Obtain the real value of the joints (q), using the inverse kinematic (IK) of the arm for the frame (bMe) calculated in the previous step:

$$\mathrm{q} = \mathrm{IK}(\mathrm{bMe})$$

3. Update the internal values of the arm with the values obtained in the previous step.

6.3.5 Kinematic control of the arm

Due to the fact that the algorithm updates the internal values of the joints, the user does not need to be careful about whether the marker is detected or not. If during a period of time the camera cannot detect the marker,

the values of the joints are updated depending of its movement, whereupon some errors due to miscalibration can be added, but in the moment that the camera can detect the marker again, these errors are cancelled. Despite the possible errors during the no detection time, these errors will always be equal or smaller than the errors produced without the algorithm. This is because the errors are produced when the arm is moving, thus the movement executed by the arm from the last position without errors is always either equal or smaller.

6.3.6 Experiment

During the experiments performed in the Sant Feliu harbour (Girona, Spain), related to the MERBOTS project, an amphora was recovered in autonomous way. This kind of interventions require a high precision since if the end-effector does not follow the planned trajectory, the fingers of the gripper could collide with the object of interest or even the floor, being able to damage any of the components.

The previously detailed algorithm was used to control the real position and orientation of the end-effector during the intervention and to correct the trajectory in case it is not the desired.

First of all and just after the arm initialization, the arm is guided to a predefined position where the marker is clearly visible. Then the camera-arm system is initialized and the intervention can start.

After the initialization of the components, and once the object of interest has been properly detected and the grasping pose determined, the end-effector is conducted to this pose. Fig. 6.4 shows the arm joint values returned by the system during the intervention, together with the joints values fixed using the algorithm.

The differences between the corrected and the non-corrected joints are quite evident, specially in the *elbow* joint. This joint started the intervention in a position close to its limit, thus probably it suffered a miscalibration previous to start the intervention. But, apart from this joint, the rest are also continually being corrected according to the marker detections.

In order to realize how these joint errors affect to the position and orientation of the end-effector, the Direct Kinematics algorithm has been applied to the corrected and non-corrected joints, obtaining the trajectories shown in the Fig. 6.5.

As a result, the algorithm has corrected positioning errors higher than $5cm$ and orientation errors around $0.3rad$. These errors could produce that the gripper is not properly positioned when it is closed, and then, that the object is not grasped correctly.

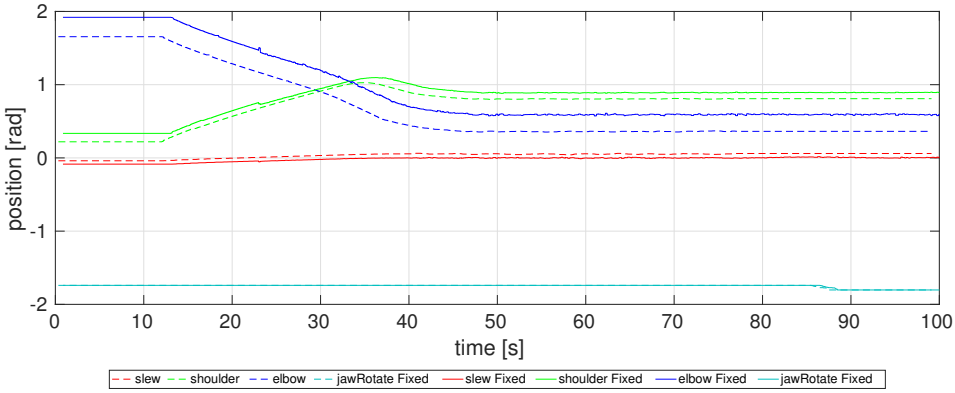


Figure 6.4: Joint values returned by the arm and after being corrected using The Visually Kinematic Controller, during the MERBOTS trial experiment.

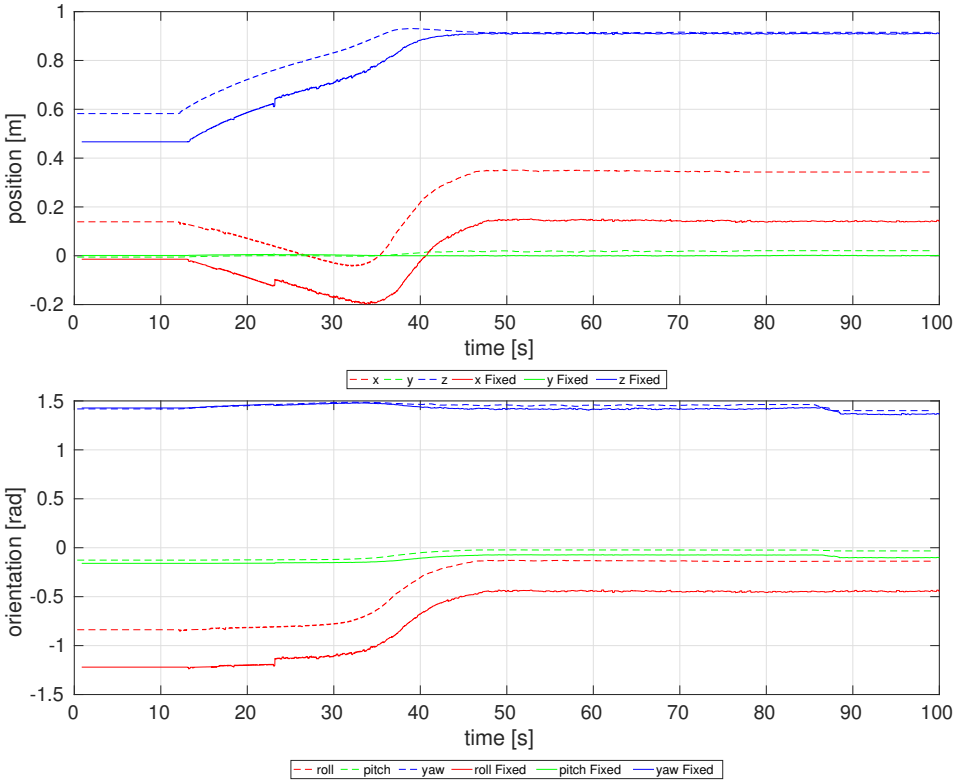


Figure 6.5: Position and Orientation returned by the system and fixed, during the MERBOTS trial experiment.

Figure 6.6: Force-Torque Sensor with 6 degrees of freedom built by the Optofocer company.

6.4 Contact Detection and Reaction

A methodology that reacts to possible collisions, during the approaching phase, modifying the planned trajectory has been developed.

A force-torque sensor has been attached to the wrist of the gripper. Each time a value higher than a threshold is read, the current state of the system is studied. Depending on this state, the methodology guess if the gripper has touched the object of interest or the floor. This assumption is used to correct the trajectory that the end-effector must follow, allowing the system to complete the intervention satisfactorily.

6.4.1 Force-Torque Sensor

As was said before, a force-torque sensor has been placed at the wrist of the gripper. In concrete, the sensor is the Optoforce HEX-70-CE-2000N [Optoforce,].

This sensor provides 6 degrees of freedom force and torque measurements. Its dimensions are 20 x 70 mm and its weight 165 g (see Fig. 6.6). It supports forces of $2000N$ compression, $800N$ tension and $300N$ in x and y. On the other hand, the torque measured can reach to $15Nm$ in x and y, and $10Nm$ in z. Having an accuracy of 3% of the full scale of each component.

The sensor has a maximum sampling frequency of $1000Hz$, and can be run in Windows and Linux, having a ROS (Robot Operating System) package available for controlling it.

6.4.2 Contact Detection

The previously described sensor is used to detect contacts between the gripper and the environment.

When the manipulator is ready to start the approaching phase, the sensor is reset and hereinafter, the values returned read.

During this phase, the gripper is supposed to be oriented with the fingers pointing, more or less, to the object of interest, thus, they would be the first in contacting with the object or even the sea floor, in case something was wrong.

Due to the sensor is placed at the wrist of the arm, when one of the fingers contact with something, the force applied will try to turn the sensor. For that reason, in this application, the torque is more useful for detecting contacts. Moreover, when the manipulator is moving, the weight of the gripper can affect the values returned by the x and y components of the torque. Thus, the z component is more suitable for controlling the possible collisions of the gripper.

Then, when the norm of the value returned by the component z of the torque surpass a predefined threshold, the arm stops.

6.4.3 Trajectory Modification

When a contact is detected, this must be studied and some decision have to be taken to continue with the intervention.

First step is to recognize if either the object of interest or the sea floor are in contact with the gripper. Knowing the shape of the gripper and obtaining the pose of the end-effector using the Direct Kinematics, it is easy to calculate the current position of each of the four extremes of the fingers. If one of them has surpass more than 75% of the theoretical size of the object, it is supposed that the sea floor has been touched. Otherwise, the object of interest is supposed to be in contact with one finger.

In case the supposition is the sea floor, the trajectory is changed in order to move the end-effector $5cm$ in the direction of the normal of the sea floor, and then, the approaching phase finishes, the sensor detection is stopped and the gripper starts closing to grasp the object.

On the other hand, if the object has been touched, it is necessary to calculate with which finger. To do so, the current pose of the end-effector with respect to the object of interest is studied. Depending on the inclination of the z component of the end-effector with respect to the maximum moment of inertia of the object, it is possible to discard two of the fingers, because in the same side of the gripper there is another finger closer to the object. Then, depending either the z component of the torque is positive or negative, the correct finger can be obtained.

Once the finger is found, the trajectory is modified to continue with the intervention avoiding this contact. First, the end-effector is moved $5cm$ against the approaching direction, then $5cm$ in y or $-y$ direction of the

current end-effector pose, to separate the finger from the object. This last translation is also done to the final grasping pose. Then, the approaching phase restarts, continuing reading the sensor values.

6.4.4 Experiments

Two experiments have been done in a water tank with the arm ARM5E attached to a fixed platform. In both, an amphora is tried to be grasped but, the grasping poses have been modified to test if the reactions to the collisions are appropriate. In the first experiment, the final pose has been translated to achieve that the fingers contact with the floor. And, in the second one, the final pose has been displaced to see how the arm reacts when a finger touches the object during the approaching phase.

Floor Contact

Fig. 6.7 shows the values of the four joints of the arm plus the joint that opens and closes the gripper, during the approaching phase of this experiment. In Fig. 6.8 can be seen the Cartesian position (x, y and z) followed by the end-effector with respect to the Grasping pose. And Fig. 6.9 presents the force and torque values returned by the sensor.

In the first experiment, the arm is moved to the Pre-grasping pose. Then, the sensor values are zeroed and the approaching phase starts. During this phase, the end-effector is guided to the Grasping pose, it is to the position (0,0,0).

But, after the second 25, the norm of the z component of the torque surpasses the $50Nm$ which is the predefined threshold. At that moment the arm stops and the contact is studied.

As the value of the used torque is negative, the right part of the gripper is in contact with something, and this lower finger is at $3cm$ of the theoretical floor, surpassing more than 85% of the theoretical object. For that reason, the floor is supposed to be touched.

Then, the arm gripper is moved $5cm$ to the normal of the plane direction, which approximately the same that the z component of the Grasping pose.

When the gripper is moved to the new Grasping Pose, the Approaching phase finishes and the JawOpening joint starts closing the gripper.

It is worth noting how, as was said before, the z component of the torque is not affected by the movement of the end effector, unlike the x and y components.

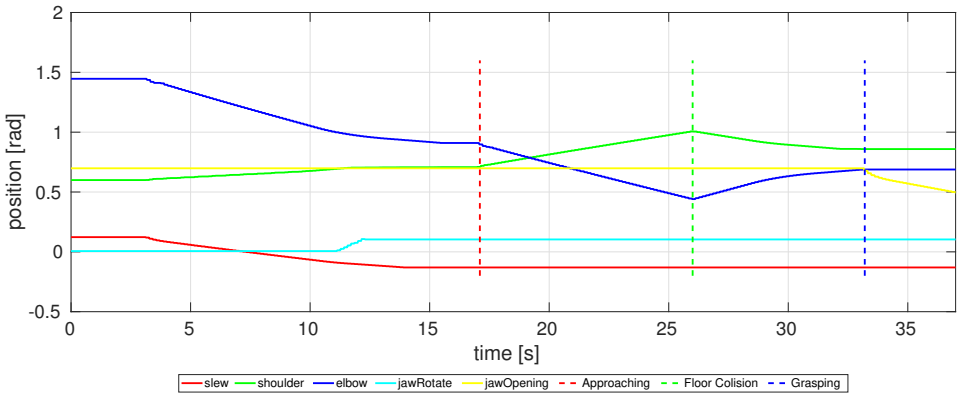


Figure 6.7: Joint values returned by the arm during the approaching phase of the experiment modified to achieve that the gripper contact with the tank floor.

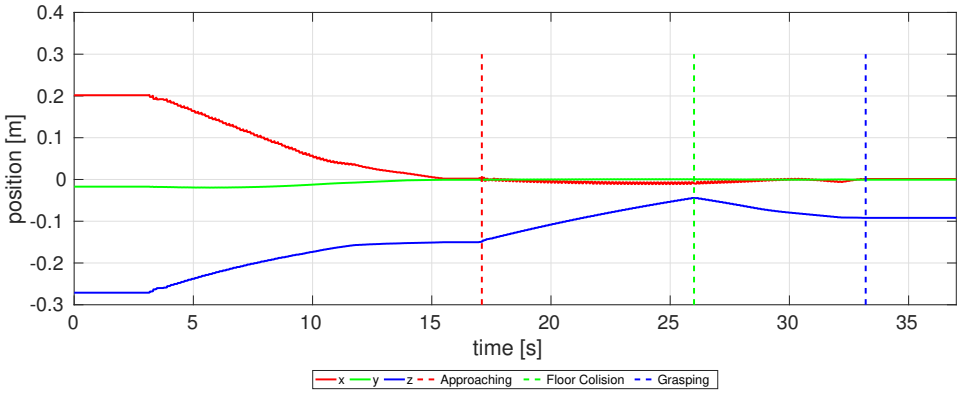


Figure 6.8: Cartesian position followed by the end-effector with respect to the Grasping Pose, during the approaching phase of the experiment modified to achieve that the gripper contact with the tank floor.

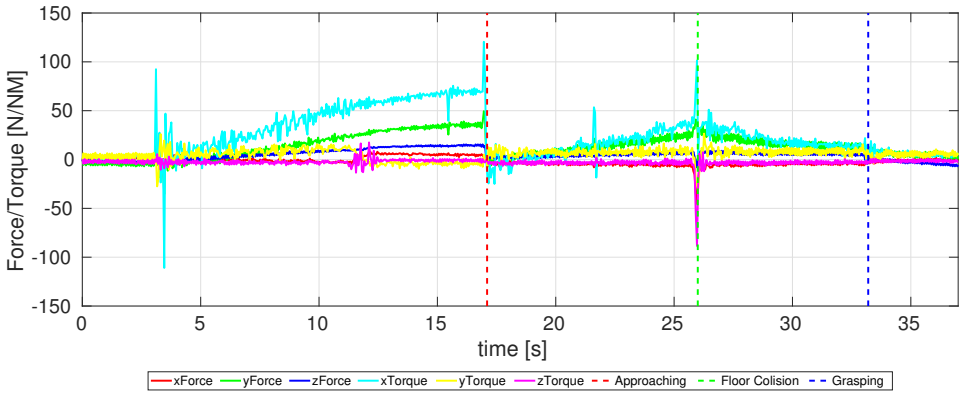


Figure 6.9: Values returned by the force-torque sensor during the approaching phase of the experiment modified to achieve that the gripper contact with the tank floor.

Object Contact

At this second experiment, the Grasping Pose has been displaced $10cm$ in the y direction of the real Grasping Pose.

Fig. 6.10 shows the values of the four joints of the arm plus the joint that opens and closes the gripper, during the approaching phase of this second experiment. In Fig. 6.11 the Cartesian position (x, y and z) followed by the end-effector with respect to the Grasping can be seen. And Fig. 6.12 presents the force and torque values returned by the sensor.

As in the last experiment, the arm is moved to the Pre-grasping pose. Then, the sensor values are zeroed and the Approaching phase starts.

After that, the arm is guided to the Grasping pose reading at the same time the z torque returned by the sensor.

Previous to the second 20, the norm of the z torque surpasses the $50Nm$, which means that a contact exists. At that moment the arm stops and the contact is studied.

The value returned is negative, thus the lowest finger of the right part of the gripper is in contact with something. The extreme of the finger is not lower enough to consider that the contact is with the floor. Thus, it is supposed that the object of interest has been touched.

In order to avoid contacting with the object, the end-effector is moved $5cm$ in the z direction of the Grasping pose. Then, as the right finger is the one involved, the end-effector is moved $5cm$ in the y component of the Grasping pose direction (see Fig. 6.11). The Grasping pose is also moved $5cm$ in its y direction, to modify the trajectory that the end-effector is going to follow, and avoiding contacting with the object during this phase.

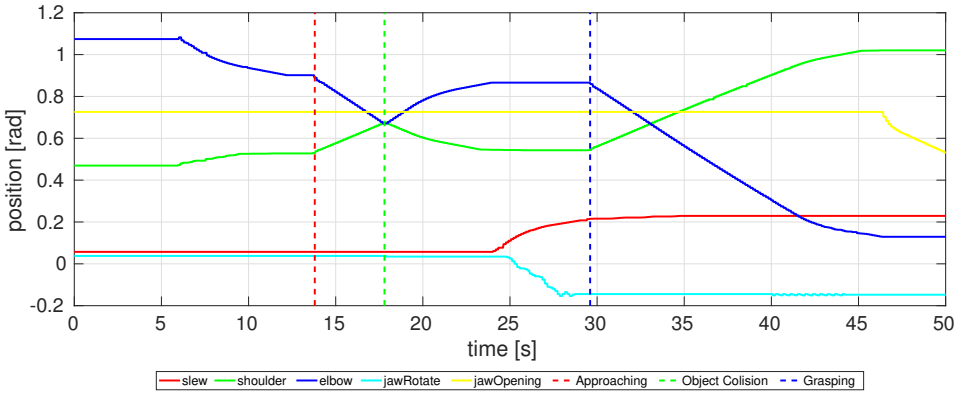


Figure 6.10: Joint values returned by the arm during the approaching phase of the experiment modified to achieve that the gripper contact with the object of interest.

After doing that, the end-effector can finally reach the new Grasping pose, and then the Approaching phase finishes.

6.5 Conclusions

This chapter describes two methodologies for controlling some of the errors that can appear during an intervention and try to fix them.

The first methodology controls the position and orientation reached by the arm end-effector during an intervention. This is based on the detection of a marker, with a known size, placed on the gripper. This detection is done using a camera, which is recording during the whole intervention, placed on the base of the arm.

Each time the camera detects the marker, its pose is calculated. This pose can be transformed to obtain the current pose of the end-effector. Using the Inverse Kinematics algorithm the current joint values can be obtained. If the joints values returned by the arm differ from the ones returned by the algorithm, the first ones are updated.

The algorithm have been successfully tested during the MERBOTS Project experiments. Where it was used to correct the trajectory followed by the end-effector whereas the I-AUV is recovering an amphora in an autonomous way.

Secondly, a methodology for detecting contacts between the gripper and the environment and correct the trajectory for avoiding them has been presented.

This methodology is based on the installation of a force-torque sensor

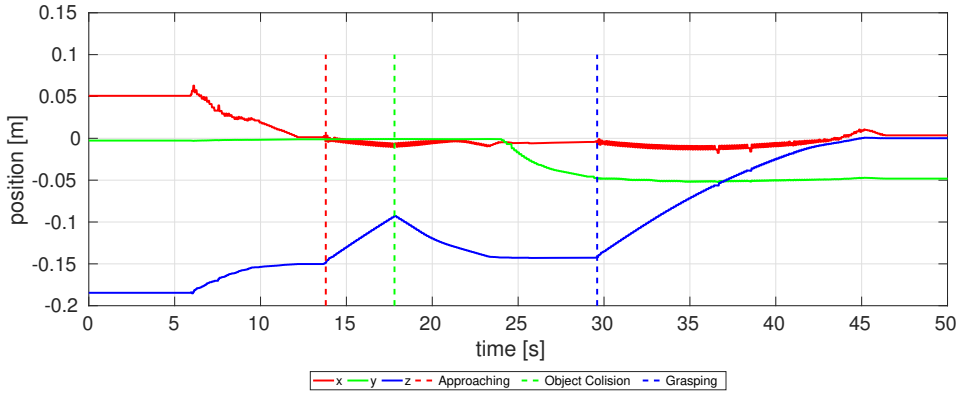


Figure 6.11: Cartesian position followed by the end-effector with respect to the Grasping Pose, during the approaching phase of the experiment modified to achieve that the gripper contact with the object of interest.

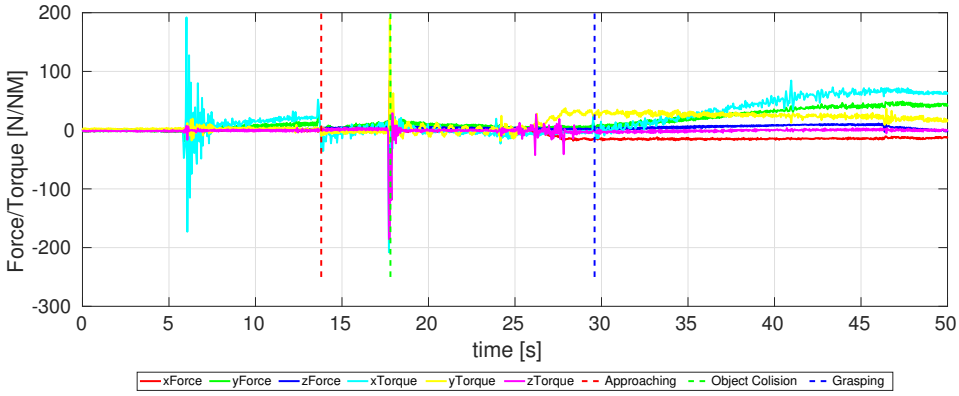


Figure 6.12: Values returned by the force-torque sensor during the approaching phase of the experiment modified to achieve that the gripper contact with the object of interest.

at the wrist of the manipulator. During the approaching phase, the values returned by the sensor are read. If a collision is detected, the current position and orientation of the end-effector is studied to detect if the gripper is in contact with the object of interest or the floor.

Then, the trajectory is changed to avoid this collision and to continue with the grasping in a satisfactory way.

Chapter 7

Conclusions

This chapter concludes the work presented throughout this document. It first summarizes the thesis by reviewing the contents described in each chapter. Then, the research contributions extracted from the proposal and the experiments are pointed out. Additionally, some interesting future research issues are commented in the future work section. Finally, the publication related to this work are listed.

7.1 Summary

Introducing autonomous systems to perform underwater intervention tasks has supposed the appearance of new problems that need to be researched. In this thesis, the problem of autonomously grasping an unknown object in an underwater scenario when the I-AUV is close to the object of interest, has been studied. This problem encompasses the reconstruction of the scene for obtaining the needed information about the object of interest, the calculation of the grasping pose with the highest probability of success, and the proper guidance of the manipulator gripper to the obtained grasping pose.

Chapter 2 introduced an approach to autonomously reconstruct the scene where the object is located. This makes possible to scan the scene from multiples views and closer to the object of interest, obtaining a more accurate and complete reconstructions compared with usual approaches. The comparison of the reconstruction of four objects using some of these approaches has also been presented.

In addition, Chapter 2 described an algorithm for optimizing the process of reconstruction. This algorithm reduces the time spent in detecting the laser peaks on the images.

Chapter 3 presented a kinematic control framework for redundant robots. It integrates, in a unified framework, the treatment of multiple tasks, mul-

tiple kinematic chains, different joints priorities and hard constraints. Two simulated experiments have demonstrated the capabilities of the framework.

An alternative that takes as input the point cloud of an scene and plans a grasping has been presented in the Chapter 4. The algorithm detects the object of interest from the point cloud of the scene, and looks for the grasping pose that better fulfils a hierarchy of tasks and strictly accomplishes a set of constraints. Three experiments have proven the adequacy of the algorithms.

A methodology that properly guides the end-effector of a manipulator to a grasping pose taken as input, has been detailed in the Chapter 5. Three grasping interventions have been performed using the methodology to evidence its suitability.

Finally, the Chapter 6 described two methodologies for detecting problems during an intervention and correct them. The first one, controls the position and orientation reached by the end-effector using a camera. The second methodology, detects contacts between the gripper and the environment using a force-torque sensor, and correct the trajectory of the end-effector for avoiding them.

7.2 Contributions

This main goal of the thesis has been fulfilled developing a set of methodologies that, all together, can solve the main goal of the thesis which consists on autonomously grasping an unknown object in an underwater scenario. Furthermore, the algorithms are simple to use and easily adaptable to manipulators with different morphology. In the development of this goal, various research contributions were achieved. these contributions are listed below:

- A new approach for 3D reconstructing an underwater scene from multiple points of views. Placing a laser stripe emitter and a camera at the forearm of a robotic arm, while the arm is moving, the laser scans the scene and the camera records it. The fact of positioning the camera at the arm makes possible to see the scene from different points of view and from positions closers to the object, obtaining a more accurate and complete reconstruction.

- An extension of the Reverse Priority framework for kinematic control of redundant robots. It integrates, in an unified framework, the treatment of multiple tasks, multiple kinematic chains that can share several joints, different joint priorities and hard constraints in the joint and Cartesian space.

- A method for segmenting a point cloud of a scene with objects placed on a flat surface and estimate the object sizes.

- A framework for obtaining the grasping pose that better fulfil a hierarchy of tasks and that accomplishes a set of constraints, using the Multi-Task Priority Algorithm.
- A methodology for updating the arm joint values by detecting a marker placed on the gripper using a camera.
- An algorithm able to correct wrong trajectory of the end-effector during the approaching phase, by detecting collisions using a force-torque sensor.

7.3 Future lines

The work that has been presented throughout this document leads to a large number of interesting issues that should be addressed in the future. Among them, we specially remark the following:

Scene Reconstruction

Although the proposed methodology have shown promising results, further work needs to be done to improve the quality of the reconstructions. The results obtained reconstructing illuminated scenes are not good enough, thus developing new methods for detecting the laser peaks in this kind of scenarios would be interesting.

The trajectory followed by the end-effector during the scan has been defined previous to the intervention. This means that the distance between the camera and the objects during the reconstruction depends on the scene. If the system were able to dynamically modify the trajectory taking into account the distance between the camera and the objects, the accuracy of the reconstruction would increase.

Once the reconstruction has finished, an algorithm that studies the point cloud obtained to calculate a new end-effector trajectory that makes possible to reconstruct parts of the object hidden for the camera in the first trajectory, would produce a more complete point cloud.

Grasp Planning

The described methodology requires to have the objects placed over a flat surface. Improving the segmentation algorithm for all kinds of surfaces would be able to solve more realistic problems.

Concerning the estimation of the shape of the different objects, the proposed solution looks for the minimum bounding box that includes the object. This is a valid option for simple grippers that wrap the object for grasping

it. More dexterous gripper requires better shape estimations. Deep learning algorithms could help to obtain the complete shape of an object from its partial reconstruction.

Continuing with the shape estimation, it is quite common that objects are partially buried in underwater scenarios. Thus, methodologies for detecting symmetries or convexity could help to approximate real shapes.

The presented method for estimating the object frame always try to position it in the center of masses of the object. Even though this is a good solution for the used gripper, other positions could be better options in some object using when dexterous grippers are used, i.e. an object with a handle.

Grasp Execution

The Multi-Task Priority algorithm has been tested in simulated and real scenarios. But a comparison with other kinematic control algorithms would be also interesting.

Experiments using the described algorithm with more complex systems like humanoids or robots with hyperredundant kinematic chains, would give the algorithm a higher credibility.

A study about the computational time spent in each iteration depending on the robot and the number of violated hard constraints, could specify the limitations of the algorithm when it is going to be used in real time.

Finally, the algorithm could be extended by giving the option of deactivate tasks at certain moments, or changing dynamically the values of the hard constraints. This would be useful for obstacle avoidance tasks or when the goal is to keep an object in the field of view.

Improving Reliability

The more number of sensors used during an intervention, the higher probability of detecting errors and being able to correct them. An example could be the use of tactile sensors at the fingers of the gripper, they could help to detect if a grasp is robust enough.

The proposed methodologies can also be extended. The marker used in the Visually Kinematic Controller, cannot be detected in dark scenarios. Thus, a luminous patter could be useful in these kind of scenes.

The force-torque sensor is only used for detecting collisions, but it could be also profitable for detecting if the object grasped is too heavy for being lifted or if for any reason it has been released during its transport.

7.4 Publications

Parts of this thesis have previously been published in the following journal and conference papers:

International Journals

1. Palomeras, N., Peñalver, A., Massot-Campos, M., Negre, P., Fernández, J. J., Ridao, P., Sanz, P. J., Oliver, G. (2016). I-AUV docking and panel intervention at Sea. *SENSORS*, 16, pp. 1-18.

2. Peñalver, A., Pérez, J., Fernández, J. J., Fornas, Sales, J., Sanz, P. J., García, J. C., Fornas, D., Marín, R. (2015). Visually-guided manipulation techniques for robotic autonomous underwater panel interventions. *Annual Reviews in control*, vol.40.

3. Pérez, J., Sales, J., Peñalver, A., Fornas, D., Fernández, J. J., García, J. C., Sanz, P. J., Marín, R., and Prats, M. (2015). Exploring 3-d reconstruction techniques: A benchmarking tool for underwater robotics. *IEEE Robotics & Automation Magazine*, 22(3):85-95.

4. Fornas, D., Sales, J., Peñalver, A., Perez, J., Fernández, J. J., Marín, R., Sanz, P. J. (2015). Fitting primitive shapes in point clouds: a practical approach to improve autonomous underwater grasp specification of unknown objects. *Journal of Experimental & Theoretical Artificial Intelligence*.

Boock Chapters

1. Pérez, J., Sales, J., Peñalver, A., Fernández, J. J., Sanz, P. J., García, J. C., Martí, J. V., Marín, R., and Fornas, D. (2015). Robotic manipulation within the underwater mission planning context.*Motion and Operation Planning of Robotic Systems*, pages 495-522. Springer.

International Conferences

1. Peñalver, A., Fernández, J. J., Soriano, A., Sanz, P. J. (2018). A multi-task priority framework for redundant robots with multiple kinematic chains under hard joint and Cartesian constraints. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2018), Madrid (Spain)*.

2. Peñalver, A., Fernández, J. J., Sanz, P. J. (2017). Autonomous underwater grasping using multi-view laser reconstruction. *Oceans 2017 MTS/IEEE Aberdeen*.

3. Peñalver, A., Fernández, J. J., Sales, J. Sanz, P. J. (2015). Multi-view underwater 3D reconstruction using stripe laser light and an eye-in-hand camera. *Oceans 2015 MTS/IEEE Genoa.*

4. Sanz, P., Pérez, J., Sales, J., Peñalver, A., Fernández, J. J., Fornas, D., Marín, R., and García, J. (2015). A benchmarking perspective of underwater intervention systems. *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles, Girona (Spain)*, 48(2):8-13.

5. Sanz, P. J., Peñalver, A., Sales, J., Fernández, J. J., Pérez, J. Fornas, D., García, J. C., Marín, R. (2015). Multipurpose underwater manipulation for archaeological intervention. *6th International Workshop on Marine Technology (MARTECH) Cartagena.*

6. Palomeras, N., Peñalver, A., Massot-Campos, M., Villacrosa, G. Negre, P., Fernández, J. J., Ridao, P., Sanz, P. J., Oliver, G., Palomer, A. (2014). I-AUV docking and intervention in a subsea panel. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), Chicago.*

7. Peñalver, A., Pérez, J., Fernández, J. J., Sales, J., Sanz, P. J., García, J. C., Fornas, D., Marín, R. (2014). Autonomous intervention on an underwater panel mockup by using visually-guided manipulation techniques. *19t IFAC World Congress, Cape Town.*

8. Sanz, P., Peñalver, A., Sales, J., Fornas, D., Fernández, J. J., Pérez, J., Bernabe, J. A. (2013). GRASPER: a multisensory based manipulation system for underwater operations. *IEEE International Conference on Systems, Man and Cybernetics, Manchester.*

9. Peñalver, A., Prats, M., Fernández, J. J., Sales, J. (2013). Semi-autonomous grasping approach of unknown objects in underwater environments combining structured light and a virtual simulation environment. *5th MARTECH International Workshop On Marine Technology, Girona.*

Bibliography

[Ala et al., 2015] Ala, R., Kim, D. H., Shin, S. Y., Kim, C., and Park, S.-K. (2015). A 3d-grasp synthesis algorithm to grasp unknown objects based on graspable boundary and convex segments. *Information Sciences*, 295:91–106. (Cited on page 56.)

[Antonelli, 2014] Antonelli, G. (2014). *Underwater Robots*. Springer, 3rd edition. (Cited on page 34.)

[Aykin and Negahdaripour, 2013] Aykin, M. D. and Negahdaripour, S. (2013). Forward-look 2-d sonar image formation and 3-d reconstruction. In *Oceans-San Diego, 2013*, pages 1–10. IEEE. (Cited on page 14.)

[Baumgartl and Henrich, 2012] Baumgartl, J. and Henrich, D. (2012). Fast vision-based grasp and delivery planning for unknown objects. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pages 1–5. VDE. (Cited on page 56.)

[Bohg et al., 2011] Bohg, J., Johnson-Roberson, M., León, B., Felip, J., Gratal, X., Bergström, N., Kragic, D., and Morales, A. (2011). Mind the gap-robotic grasping under incomplete observation. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 686–693. IEEE. (Cited on page 56.)

[Bohg et al., 2014] Bohg, J., Morales, A., Asfour, T., and Kragic, D. (2014). Data-driven grasp synthesis–a survey. *IEEE Transactions on Robotics*, 30(2):289–309. (Cited on page 55.)

[Bone et al., 2008] Bone, G. M., Lambert, A., and Edwards, M. (2008). Automated modeling and robotic grasping of unknown three-dimensional objects. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 292–298. IEEE. (Cited on page 55.)

[Bowtech,] Bowtech. Miniature high resolution underwater ccd camera. http://www.teledynemarine.com/divecam-720m-al/?BrandID=5. (Cited on page 27.)

[Brandou et al., 2007] Brandou, V., Allais, A.-G., Perrier, M., Malis, E., Rives, P., Sarrazin, J., and Sarradin, P.-M. (2007). 3d reconstruction of natural underwater scenes using the stereovision system iris. In *OCEANS 2007-Europe*, pages 1–6. Ieee. (Cited on page 16.)

[Bruno et al., 2013] Bruno, F., Gallo, A., Filippo, F. D., Muzzupappa, M., Petriaggi, B. D., and Caputo, P. (2013). 3d documentation and monitoring of the experimental cleaning operations in the underwater archaeological site of baia (italy). In *2013 Digital Heritage International Congress (DigitalHeritage)*, volume 1, pages 105–112. (Cited on page 15.)

[Caccia, 2006] Caccia, M. (2006). Laser-triangulation optical-correlation sensor for rov slow motion estimation. *IEEE Journal of Oceanic Engineering*, 31(3):711–727. (Cited on page 17.)

[Calli et al., 2011] Calli, B., Wisse, M., and Jonker, P. (2011). Grasping of unknown objects via curvature maximization using active vision. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 995–1001. IEEE. (Cited on page 57.)

[Centelles et al., 2015] Centelles, D., Rubino, E., Soler, M., Martí, J., Sales, J., Marin, R., and Sanz, P. (2015). Underwater radio frequency based localization and image transmission system, including specific compression techniques, for autonomous manipulation. In *OCEANS 2015-Genova*, pages 1–5. IEEE. (Cited on page 5.)

[Chan and Dubey, 1995] Chan, T. F. and Dubey, R. V. (1995). A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Transactions on Robotics and Automation*, 11(2):286–292. (Cited on page 36.)

[Chaumette and Marchand, 2000] Chaumette, F. and Marchand, E. (2000). A new redundancy-based iterative scheme for avoiding joint limits. application to visual servoing. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 1720–1725. IEEE. (Cited on page 35.)

[Chiaverini, 1997] Chiaverini, S. (1997). Singularity-robust task-priority redundancy resolution for realtime kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410. (Cited on page 34.)

[Chiaverini et al., 1994] Chiaverini, S., Siciliano, B., and O, E. (1994). Review of the damped least-squares methods for inverse kinematics with experiments on an industrial robot manipulator. *IEEE Transactions on Control Systems Technology*, 2(2):123–134. (Cited on page 34.)

[Choi et al., 1994] Choi, S., Takashige, G. Y., and Yuh, J. (1994). Experimental study on an underwater robotic vehicle: ODIN. In *Autonomous Underwater Vehicle Technology, 1994. AUV '94., Proceedings of the 1994 Symposium on*, pages 79–84. (Cited on page 3.)

[Cocito et al., 2003] Cocito, S., Sgorbini, S., Peirano, A., and Valle, M. (2003). 3-d reconstruction of biological objects using underwater video technique and image processing. *Journal of Experimental Marine Biology and Ecology*, 297(1):57–70. (Cited on page 15.)

[Costa et al., 2006] Costa, C., Loy, A., Cataudella, S., Davis, D., and Scardi, M. (2006). Extracting fish size using dual underwater cameras. *Aquacultural Engineering*, 35(3):218–227. (Cited on page 16.)

[Dune et al., 2008] Dune, C., Marchand, E., Collowet, C., and Leroux, C. (2008). Active rough shape estimation of unknown objects. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3622–3627. IEEE. (Cited on page 55.)

[Egeland, 1987] Egeland, O. (1987). Task-space tracking with redundant manipulators. *IEEE Journal on Robotics and Automation*, 3(5):471–475. (Cited on page 34.)

[Escande et al., 2014] Escande, A., Mansard, N., and Wieber, P.-B. (2014). Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028. (Cited on page 35.)

[Evans et al., 2001] Evans, J., Keller, K., Smith, J., Marty, P., and Rigaud, O. (2001). Docking techniques and evaluation trials of the SWIMMERs AUV: an autonomous deployment. In *AUV for work-class ROVs. OCEANS, 2001*, pages 520–528. (Cited on page 3.)

[Evans et al., 2003] Evans, J., Redmond, P., Plakas, C., Hamilton, K., and Lane, D. (2003). Autonomous docking for Intervention-AUVs using sonar and video-based real-time 3D pose estimation. In *OCEANS 2003. Proceedings*, volume 4, pages 2201–2210. (Cited on page 3.)

[Fernández et al., 2015] Fernández, J., Pérez, J., Peñalver, A., Sales, J., Fornas, D., and Sanz, P. (2015). Benchmarking using uwsim, simurv and ros: An autonomous free floating dredging intervention case study. In *OCEANS 2015-Genova*, pages 1–7. IEEE. (Cited on page 5.)

[Fernández et al., 2013] Fernández, J. J., Prats, M., Sanz, P. J., García, J. C., Marin, R., Robinson, M., Ribas, D., and Ridao, P. (2013). Grasp-

ing for the seabed: Developing a new underwater robot arm for shallow-water intervention. *IEEE Robotics & Automation Magazine*, 20(4):121–130. (Cited on pages 25, 47, 64, 100, and 115.)

[Fischler and Bolles, 1987] Fischler, M. A. and Bolles, R. C. (1987). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in computer vision*, pages 726–740. Elsevier. (Cited on page 59.)

[Flacco and De Luca, 2014] Flacco, F. and De Luca, A. (2014). A reverse priority approach to multi-task control of redundant robots. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 22014)*, pages 2421–2427, Chicago, IL, USA. (Cited on pages 34, 35, 37, and 39.)

[Flacco and De Luca, 2015] Flacco, F. and De Luca, A. (2015). Unilateral constraints in the reverse priority redundancy resolution method. In *2015 IEE/RSJ International Conference on Intelligent Robots and Systems (IROS 22015)*, pages 2564–2571, Hamburg, Germany. (Cited on page 35.)

[Forest et al., 2004] Forest, J., Salvi, J., Cabruja, E., and Pous, C. (2004). Laser stripe peak detector for 3D scanners. a FIR filter approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 646–649 Vol.3. (Cited on page 20.)

[Fornas et al., 2016] Fornas, D., Sales, J., Peñalver, A., Pérez, J., Fernández, J. J., Marín, R., and Sanz, P. J. (2016). Fitting primitive shapes in point clouds: a practical approach to improve autonomous underwater grasp specification of unknown objects. *Journal of Experimental & Theoretical Artificial Intelligence*, 28(1-2):369–384. (Cited on page 5.)

[Garcia et al., 2010] Garcia, J., Prats, M., Sanz, P., Marin, R., and Belmonte, O. (2010). Exploring multimodal interfaces for underwater intervention systems. In *Proceedings of the IEEE ICRA 2010 Workshop on Multimodal Human-Robot Interfaces*. (Cited on page 5.)

[Garcia et al., 2015] Garcia, J. C., Patrao, B., Almeida, L., Perez, J., Menezes, P., Dias, J., and Sanz, P. (2015). Design and evaluation of a natural interface for remote operation of underwater robots. *IEEE computer graphics and applications*. (Cited on page 5.)

[Guo, 2013] Guo, Y. (2013). 3d underwater topography rebuilding based on single beam sonar. In *Signal Processing, Communication and Computing (ICSPCC), 2013 IEEE International Conference on*, pages 1–5. IEEE. (Cited on page 14.)

[Hildebrandt et al., 2008] Hildebrandt, M., Kerdels, J., Albiez, J., and Kirchner, F. (2008). A practical underwater 3d-laserscanner. In *OCEANS 2008*, pages 1–5. IEEE. (Cited on page 16.)

[Hsiao et al., 2010] Hsiao, K., Chitta, S., Ciocarlie, M., and Jones, E. G. (2010). Contact-reactive grasping of objects with partial shape information. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1228–1235. IEEE. (Cited on page 56.)

[Inglis et al., 2012] Inglis, G., Smart, C., Vaughn, I., and Roman, C. (2012). A pipeline for structured light bathymetric mapping. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4425–4432. (Cited on page 19.)

[Intelmann, 2006] Intelmann, S. S. (2006). Comments on hydrographic and topographic lidar acquisition and merging with multibeam sounding data acquired in the olympic coast national marine sanctuary. (Cited on page 13.)

[Jordt-Sedlazeck and Koch, 2013] Jordt-Sedlazeck, A. and Koch, R. (2013). Refractive structure-from-motion on underwater images. In *Proceedings of the IEEE international Conference on Computer Vision*, pages 57–64. (Cited on page 15.)

[Kanoun et al., 2011] Kanoun, O., Lamiraux, F., and Wieber, P.-B. (2011). Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task. *IEEE Transactions on Robotics*, 27(4):785–792. (Cited on page 35.)

[Kato and Billinghurst, 1999] Kato, H. and Billinghurst, M. (1999). Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proc. 2nd IEEE and ACM Int. Workshop on Augmented Reality (IWAR '99)*, pages 85–94. (Cited on pages 19 and 115.)

[Klingbeil et al., 2011] Klingbeil, E., Rao, D., Carpenter, B., Ganapathi, V., Ng, A. Y., and Khatib, O. (2011). Grasping with application to an autonomous checkout robot. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2837–2844. IEEE. (Cited on page 57.)

[Kondo et al., 2004] Kondo, H., Maki, T., Ura, T., Nose, Y., Sakamaki, T., and Inaishi, M. (2004). Structure tracing with a ranging system using a sheet laser beam. In *Underwater Technology, 2004. UT'04. 2004 International Symposium on*, pages 83–88. IEEE. (Cited on page 17.)

[Kumar and Kumar, 2011] Kumar, N. S. and Kumar, R. (2011). Design & development of autonomous system to build 3d model for underwater objects using stereo vision technique. In *India Conference (INDICON), 2011 Annual IEEE*, pages 1–4. IEEE. (Cited on page 15.)

[Lane et al., 1997] Lane, D. M., Davies, J. B. C., Casalino, G., Bartolini, G., Cannata, G., Veruggio, G., Canals, M., Smith, C., O'Brien, D. J., Pickett, M., Robinson, G., Jones, D., Scott, E., Ferrara, A., Angelleti, D., Coccoli, M., Bono, R., Virgili, P., Pallas, R., and Gracia, E. (1997). Amadeus: advanced manipulation for deep underwater sampling. *IEEE Robotics Automation Magazine*, 4(4):34–45. (Cited on page 3.)

[Lane et al., 2012] Lane, D. M., Maurelli, F., Larkworthy, T., Caldwell, D., Salvi, J., Fox, M., and Kyriakopoulos, K. (2012). PANDORA: Persistent autonomy through learning, adaptation, observation and re-planning. In *Proceedings of the 3rd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*, pages 367–372, Porto, Portugal. (Cited on page 4.)

[Lee et al., 2003] Lee, H.-K., Kim, M.-H., and Lee, S.-R. (2003). 3d optimal determination of grasping points with whole geometrical modeling for unknown objects. *Sensors and Actuators A: Physical*, 107(2):146–151. (Cited on page 56.)

[Lei et al., 2017] Lei, Q., Meijer, J., and Wisse, M. (2017). A survey of unknown object grasping and our fast grasping algorithm-c shape grasping. In *Control, Automation and Robotics (ICCAR), 2017 3rd International Conference on*, pages 150–157. IEEE. (Cited on page 55.)

[Lenz et al., 2015] Lenz, I., Lee, H., and Saxena, A. (2015). Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724. (Cited on page 57.)

[Liegeois, 1977] Liegeois, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE transactions on systems, man, and cybernetics*, 7(12):868–871. (Cited on page 36.)

[Lippiello et al., 2013] Lippiello, V., Ruggiero, F., Siciliano, B., and Villani, L. (2013). Visual grasp planning for unknown objects using a multifingered robotic hand. *IEEE/ASME Transactions on Mechatronics*, 18(3):1050–1059. (Cited on page 56.)

[Lundblad et al., 2006] Lundblad, E. R., Wright, D. J., Miller, J., Larkin, E. M., Rinehart, R., Naar, D. F., Donahue, B. T., Anderson, S. M., and Battista, T. (2006). A benthic terrain classification scheme for american samoa. *Marine Geodesy*, 29(2):89–111. (Cited on page 13.)

[Maciejewski and Klein, 1985] Maciejewski, A. A. and Klein, C. A. (1985). Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *The International Journal of Robotics Research*, 4(3):109–117. (Cited on page 34.)

[Maldonado et al., 2010] Maldonado, A., Klank, U., and Beetz, M. (2010). Robotic grasping of unmodeled objects using time-of-flight range data and finger torque information. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2586–2591. IEEE. (Cited on page 57.)

[Mansard et al., 2009] Mansard, N., Khatib, O., and Kheddar, A. (2009). A unified approach to integrate constraints in the stack of tasks. *IEEE Transactions on Robotics*, 25(3):670–685. (Cited on page 35.)

[Marani et al., 2009] Marani, G., Choi, S. K., and Yuh, J. (2009). Underwater autonomous manipulation for intervention missions AUVs. *Ocean Engineering*, 36(1):15–23. (Cited on page 4.)

[Marey and Chaumette, 2010] Marey, M. and Chaumette, F. (2010). New strategies for avoiding robot joint limits: Application to visual servoing using a large projection operator. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 6222–6227. IEEE. (Cited on page 36.)

[Massot-Campos and Oliver-Codina, 2015] Massot-Campos, M. and Oliver-Codina, G. (2015). Optical sensors and methods for underwater 3d reconstruction. *Sensors*, 15(12):31525–31557. (Cited on page 13.)

[Massot-Campos et al., 2015] Massot-Campos, M., Oliver-Codina, G., Kemal, H., Petillot, Y., and Bonin-Font, F. (2015). Structured light and stereo vision for underwater 3d reconstruction. In *OCEANS 2015-Genova*, pages 1–6. IEEE. (Cited on page 17.)

[McKinnon et al., 2011] McKinnon, D., He, H., Upcroft, B., and Smith, R. N. (2011). Towards automated and in-situ, near-real time 3-d reconstruction of coral reef environments. In *OCEANS 2011*, pages 1–10. IEEE. (Cited on page 14.)

[Negahdaripour et al., 2009] Negahdaripour, S., Sekkati, H., and Pirsiavash, H. (2009). Opti-acoustic stereo imaging: On system calibration and 3-d target reconstruction. *IEEE Transactions on image processing*, 18(6):1203–1214. (Cited on page 14.)

[Nicosevici et al., 2009] Nicosevici, T., Gracias, N., Negahdaripour, S., and Garcia, R. (2009). Efficient three-dimensional scene modeling and mosaicing. *Journal of Field Robotics*, 26(10):759–788. (Cited on page 15.)

[Optoforce,] Optoforce. Optoforce hex-70-ce-2000n. https://optoforce.com/. (Cited on page 122.)

[PCL-Segmenation,] PCL-Segmenation. Euclidean cluster extraction. http://www.pointclouds.org/documentation/tutorials/cluster_extraction.php. (Cited on page 59.)

[Peñalver et al., 2015a] Peñalver, A., Fernández, J. J., Sales, J., and Sanz, P. J. (2015a). Multi-view underwater 3d reconstruction using a stripe laser light and an eye-in-hand camera. In *OCEANS 2015-Genova*, pages 1–6. IEEE. (Cited on page 11.)

[Peñalver et al., 2017] Peñalver, A., Fernández, J. J., and Sanz, P. J. (2017). Autonomous underwater grasping using multi-view laser reconstruction. In *OCEANS 2017 - Aberdeen*, pages 1–5. (Cited on pages 11 and 53.)

[Peñalver et al., 2018] Peñalver, A., Fernández, J. J., Soriano, A., and Sanz, P. J. (2018). A multi-task priority framework for redundant robots with multiple kinematic chains under hard joint and cartesian constraints. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2018)*, Madrid, Spain. (Cited on page 33.)

[Peñalver et al., 2015b] Peñalver, A., Pérez, J., Fernández, J. J., Sales, J., Sanz, P. J., García, J. C., Fornas, D., and Marín, R. (2015b). Visually-guided manipulation techniques for robotic autonomous underwater panel interventions. *Annual Reviews in Control*, 40:201 – 211. (Cited on page 113.)

[Pérez et al., 2017] Pérez, J., Sales, J., Peñalver, A., Fernández, J., Fornas, D., García, J., Marín, R., and Sanz, P. (2017). Benchmarking water turbidity effect on tracking algorithms **this work was partly supported by spanish ministry of economy and competitiveness under grant dpi2014-57746-c3 (merbots project), by universitat jaume i grant pid2010-12 and phd grants predoc/2012/47 and predoc/2013/46, by generalitat valenciana phd grant acif/2014/298 and prometeo/2016/066 grant. *IFAC-PapersOnLine*, 50(1):11191 – 11196. 20th IFAC World Congress. (Cited on page 5.)

[Prats et al., 2012a] Prats, M., Fernández, J., and Sanz, P. (2012a). Combining template tracking and laser peak detection for 3D reconstruction

and grasping in underwater environments. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 106–112. (Cited on pages 18 and 20.)

[Prats et al., 2012b] Prats, M., Fernández, J. J., and Sanz, P. J. (2012b). An approach for semi-autonomous recovery of unknown objects in underwater environments. In *Optimization of Electrical and Electronic Equipment (OPTIM), 2012 13th International Conference on*, pages 1452–1457. IEEE. (Cited on page 17.)

[Prats et al., 2012c] Prats, M., Pérez, J., Fernández, J. J., and Sanz, P. J. (2012c). An open source tool for simulation and supervision of underwater intervention missions. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2577–2582. IEEE. (Cited on pages 5 and 25.)

[Prats et al., 2012d] Prats, M., Pérez, J., Fernández, J. J., and Sanz, P. J. (2012d). An open source tool for simulation and supervision of underwater intervention missions. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2577–2582. IEEE. (Cited on page 46.)

[Prats et al., 2012e] Prats, M., Ribas, D., Palomeras, N., García, J. C., Nannen, V., Wirth, S., Fernández, J. J., Beltrán, J. P., Campos, R., Ridao, P., et al. (2012e). Reconfigurable auv for intervention missions: a case study on underwater object recovery. *Intelligent Service Robotics*, 5(1):19–31. (Cited on page 5.)

[Ribas et al., 2012] Ribas, D., Palomeras, N., Ridao, P., Carreras, M., and Mallios, A. (2012). Girona 500 auv: From survey to intervention. *IEEE/ASME Transactions on Mechatronics*, 17(1):46–53. (Cited on pages 47, 64, and 100.)

[Ribas et al., 2015] Ribas, D., Ridao, P., Turetta, A., Melchiorri, C., Palli, G., FernÃąndez, J. J., and Sanz, P. J. (2015). I-auv mechatronics integration for the trident fp7 project. *IEEE/ASME Transactions on Mechatronics*, 20(5):2583–2592. (Cited on page 61.)

[Richtsfeld and Vincze, 2008] Richtsfeld, M. and Vincze, M. (2008). Grasping of unknown objects from a table top. In *Workshop on Vision in Action: Efficient strategies for cognitive agents in complex environments*. (Cited on page 57.)

[Rigaud et al., 1998] Rigaud, V., Coste-Maniere, E., Aldon, M. J., Probert, P., Perrier, M., Rives, P., Simon, D., Lang, D., Kiener, J., Casal, A., Amar,

J., Dauchez, P., and Chantler, M. (1998). Union: underwater intelligent operation and navigation. *IEEE Robotics Automation Magazine*, 5(1):25–35. (Cited on page 3.)

[Rosenblum and Kamgar-Parsi, 1992] Rosenblum, L. and Kamgar-Parsi, B. (1992). 3d reconstruction of small underwater objects using high-resolution sonar data. In *Autonomous Underwater Vehicle Technology, 1992. AUV'92., Proceedings of the 1992 Symposium on*, pages 228–235. IEEE. (Cited on page 14.)

[Rubino et al., 2017] Rubino, E. M., Centelles, D., Sales, J., Martí, J. V., Marín, R., Sanz, P. J., and Alvares, A. J. (2017). Progressive image compression and transmission with region of interest in underwater robotics. In *OCEANS 2017 - Aberdeen*, Aberdeen (UK). IEEE, IEEE. (Cited on page 5.)

[Sanz et al., 2013] Sanz, P. J., Ridao, P., Oliver, G., Casalino, G., Petillot, Y., Silvestre, C., Melchiorri, C., and Turetta, A. (2013). Trident an european project targeted to increase the autonomy levels for underwater intervention missions. In *2013 OCEANS - San Diego*, pages 1–10. (Cited on pages 4 and 5.)

[Schmidt and Rzhanov, 2012] Schmidt, V. E. and Rzhanov, Y. (2012). Measurement of micro-bathymetry with a gopro underwater stereo camera pair. In *Oceans, 2012*, pages 1–6. IEEE. (Cited on page 16.)

[Servos et al., 2013] Servos, J., Smart, M., and Waslander, S. L. (2013). Underwater stereo slam with refraction correction. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3350–3355. IEEE. (Cited on page 16.)

[Siciliano and Khatib, 2008] Siciliano, B. and Khatib, O., editors (2008). *Springer Handbook of Robotics*. Springer Berlin Heidelberg. (Cited on page 34.)

[Siciliano and Slotine, 1991] Siciliano, B. and Slotine, J. J. E. (1991). A general framework for managing multiple tasks in highly redundant robotic systems. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 1211–1216 vol.2. (Cited on page 37.)

[Simetti et al., 2014] Simetti, E., Casalino, G., Torelli, S., Sperinde, A., and Turetta, A. (2014). Floating underwater manipulation: Developed control methodology and experimental validation within the trident project. *Journal of Field Robotics*, 31(3):364–385. (Cited on page 35.)

[Speth et al., 2008] Speth, J., Morales, A., and Sanz, P. J. (2008). Vision-based grasp planning of 3d objects by extending 2d contour based algorithms. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2240–2245. (Cited on page 58.)

[Stückler et al., 2011] Stückler, J., Steffens, R., Holz, D., and Behnke, S. (2011). Real-time 3d perception and efficient grasp planning for everyday manipulation tasks. In *ECMR*, pages 177–182. (Cited on page 57.)

[Suzuki and Oka, 2016] Suzuki, T. and Oka, T. (2016). Grasping of unknown objects on a planar surface using a single depth image. In *Advanced Intelligent Mechatronics (AIM), 2016 IEEE International Conference on*, pages 572–577. IEEE. (Cited on page 57.)

[Svoboda et al., 2005] Svoboda, T., Martinec, D., and Pajdla, T. (2005). A convenient multi-camera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 14(4):407–422. (Cited on page 19.)

[Tritech,] Tritech. Seastripe laser line projector. http://www.tritech.co.uk/media/support/manuals/seastripe-laser-line-projector-mk2-operator-installation-manual.pdf. (Cited on page 27.)

[Videre,] Videre. Low-power stereo head with a fixed baseline. http://users.rcn.com/mclaughl.dnai/support.htm. (Cited on page 27.)

[Vogt and Smoot, 1984] Vogt, P. R. and Smoot, N. C. (1984). The geisha guyots: Multibeam bathymetry and morphometric interpretation. *Journal of Geophysical Research: Solid Earth*, 89(B13):11085–11107. (Cited on page 13.)

[Wang et al., 2000] Wang, C.-C., Shyue, S.-W., and Cheng, S.-H. (2000). Underwater structure inspection with laser light stripes. In *Proceedings of the 2000 International Symposium on Underwater Technology (Cat. No.00EX418)*, pages 201–205. (Cited on page 14.)

[Wang et al., 1995] Wang, H., Rock, S., and Lee, M. (1995). Experiments in automatic retrieval of underwater objects with an AUV. In *OCEANS '95. MTS/IEEE. Challenges of Our Changing Global Environment. Conference Proceedings.*, volume 1, pages 366–373 vol.1. (Cited on page 3.)

[Whitney, 1969] Whitney, D. E. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, 10(2):47–53. (Cited on page 34.)

[Yuh and West, 2001] Yuh, J. and West, M. (2001). Underwater robotics. *Advanced Robotics*, 15(5):609–639. (Cited on page 1.)

[Zerr and Stage, 1996] Zerr, B. and Stage, B. (1996). Three-dimensional reconstruction of underwater objects from a sequence of sonar images. In *Image Processing, 1996. Proceedings., International Conference on*, volume 3, pages 927–930. IEEE. (Cited on page 14.)