ROBOT LEARNING APPLIED TO AUTONOMOUS UNDERWATER VEHICLES FOR INTERVENTION TASKS

Arnau Carrera Viñas

Doctoral Thesis

Robot Learning applied to Autonomous Underwater Vehicles for intervention tasks

Arnau Carrera Viñas

2017

Doctoral Thesis

Robot Learning applied to Autonomous Underwater Vehicles for intervention tasks

Arnau Carrera Viñas

2017

Doctoral Program in Technology

Supervised by:

Marc Carreras, Narcís Palomeras and Petar Kormushev

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

Doctor of Philosophy

Certificate of Thesis Direction

Dr. Marc Carreras and Dr. Narcís Palomeras from the Universitat de Girona, and Dr. Petar Kormushev from Imperial College London.

Declare:

That the work entitled *Robot Learning applied to Autonomous Underwater Vehicles for intervention tasks* presented by Arnau Carrera Viñas to obtain his degree in Doctor of Philosophy has been developed under our supervision and fulfils the requirements to obtain an International Mention.
In order to certify the aforesaid statement, we sign this document.

Girona, 31 May 2017

Dr. Marc Carreras

Dr. Narcís Palomeras

Dr. Petar Kormushev

Acknowledgments

Along these lines, I would like to express my gratitude to all the persons that helped me in countless ways, throughout the entire thesis.

M'agradaria començar agraint el suport de la meva família i especialment als meus pares. Sempre han estat al meu costat i m'han ajudat a fer aquest camí més fàcil. Sense el seu suport no hauria set possible arribar fins aquí.

Vull donar les gràcies als meus supervisors. Primer de tot en Marc, ja que va ser ell el que em va oferir la possibilitat de començar aquest doctorat, i durant aquests anys m'ha guiat amb el seu optimisme, sempre arribant un pas més endavant de l'esperat. A en Narcís vull agrair-li la seva supervisió més pròxima, amb els petits passos del dia a dia, i la seva opinió més crítica. Finally to Petar, for introducing me the topic of this thesis and for encouraging me since then.

També vull agrair a tots els membres del CIRS, els bons moments i la seva companyia durant els anys del doctorat. Sense ells el doctorat hauria set només un període cansat i ple de frustracions. Vull destacar a la Tali que ha set com una altra supervisora, en David que ha fet possible tenir els end-effectors amb uns dissenys tan elaborats, i finalment en Lluís i en Carles que sempre m'han ajudat a poder tenir un robot funcionant després de tots els desastres i accidents que he causat. Evidentment, també vull estendre aquest agraïment a la gent del P4 i a tot el grup VICOROB en general i especialment a la Joseta, la Mireia, i l'Anna. Ha set un plaer fer el doctorat envoltat per tota la gent de VICOROB.

I am grateful to all the people of the PANDORA project who has worked with me during these years. Especially to the people who has shared hours of work with me in the water tank at the CIRS. Those are George C. Karras and Bechlioulis Charalampos, from NTUA; Reza Ahmadzadeh, Nawiid Jamali, and Matteo Leonetti from IIT; and Michael Cashmore from KCL.

I am also grateful to the people of the CMRE. Especially to Pilar Caamaño, Alberto Tremori and Robert Been, to see my sleep face and not so good mood during some months. I want to extend the gratitude to all the people who during this last year have insisted questioning me if I had already finished it or not yet. I also want to thank them for understanding all the times I had rejected plans for the weekends to work on this PhD.

Moreover, I would like to thank Başak for her support, encouragement, and comprehension during the final part of this work, içten teşekkürler. Although she does not consider it important.

Last but not the least, I want to express also my gratitude to the anonymous reviewers that contributed with valuable comments to improve the work and the publications derived from this PhD. Likewise I would like to thank the reviewers of this thesis manuscript for their time and their comments, and Judy Fenwick for proofreading it. Finally I would like to thank the agencies that partially funded this work: the EU project FP7-ICT-2011-7-288273 PANDORA and the national project DPI2011-27977-C03-02 COMAROB.

List of Publications

Publications derived from this thesis

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

Journals publications:

- **[PR-15] A. Carrera**, N. Palomeras, N. Hurtós, P. Kormushev, and M. Carreras "Cognitive system for autonomous underwater intervention", *Pattern Recognition Letters*, North-Holland, 2015.

- **[AR-15]** N. Palomeras, **A. Carrera**, N. Hurtós, G. Karras, C. Bechlioulis, M. Cashmore, D. Magazzeni, D. Long, K.J. Kyriakopoulos, P. Kormushev, J. Salvi, and M. Carreras, "Toward persistent autonomous intervention in a subsea panel", *Autonomous Robots*, 1-28, Springer, 2015.

Conferences publications:

- **[NGCUV-15] A. Carrera**, N. Palomeras, N. Hurtós, and M. Carreras "Free-floating panel intervention by means of Learning by Demonstration", *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*, Girona, Spain, 2015.

- **[OCEANS-15] A. Carrera**, N. Palomeras, N. Hurtós, P. Kormushev, and M. Carreras "Learning multiple strategies to perform a valve turning with underwater currents using an I-AUV" *MTS/IEEE OCEANS - Genova*, Genova, Italy, 2015.

- **[CCIA-14] A. Carrera**, N. Palomeras, N. Hurtós, P. Kormushev, and M. Carreras, "Learning by demonstration applied to underwater intervention," in *17th International Conference of the Catalan Association of Artificial Intelligence*, Barcelona, 2014.

- **[WS-ICRA-14] A. Carrera**, G. Karras, C. Bechlioulis, N. Palomeras, N. Hurtós, K. Kyriakopoulos, P. Kormushev, and M. Carreras, "Improving a learning by demonstration framework for intervention auvs by means of an UVMS controller," in *ICRA Workshop on Persistent Autonomy for Marine Robotics*, Hong Kong, 2014.

- [**OCEANS-14**] **A. Carrera**, N. Palomeras, D. Ribas, P. Kormushev, and M. Carreras "An Intervention-AUV learns how to perform an underwater valve turning" *MTS/IEEE OCEANS - Taipei*, Taipei, Taiwan, 2014.
- [**OCEANS-13**] **A. Carrera**, M. Carreras, P. Kormushev, N. Palomeras, and S. Nagappa "Towards valve turning with an AUV using Learning by Demonstration" *MTS/IEEE OCEANS - Bergen*, Bergen, Norway, 2013.
- [**CIT-12**] **A. Carrera**, S.R. Ahmadzadeh, A. Ajoudani, P. Kormushev, M. Carreras, and D.G. Caldwell "Towards Autonomous Robotic Valve Turning" *Cybernetics and Information Technologies*, 2012.

Other publications

Parallel work at the time of this thesis led to the following publications:

- [**ICRA-15**] N. Jamali, P. Kormushev, **A. Carrera**, M. Carreras, and D.G. Caldwell "Underwater robot-object contact perception using machine learning on force/torque sensor feedback",*IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, Washington, 2015.
- [**SAS-15**] N. Hurtós, N. Palomeras, **A. Carrera**, and M. Carreras, "Adaptive frequency filtering for forward-looking sonar imagery spectral registration", *Sensors Applications Symposium (SAS)*, 2015.
- [**ICAPS-15**] M. Cashmore, M. Fox, D. Long, D. Magazzeni, B. Ridder, **A. Carrera**, N. Palomeras, N. Hurtós, and M. Carreras, "ROSPlan: Planning in the Robot Operating System", *Twenty-Fifth International Conference on Automated Planning and Scheduling (ICAPS)*,Jerusalem, Israel, 2015.
- [**EUROCAST-15**] M. Carreras, **A. Carrera**, N. Palomeras, D. Ribas, N. Hurtós, J. Salvi, and P.Ridao, "Intervention Payload for Valve Turning with an AUV", *International Conference on Computer Aided Systems Theory*, 2015.
- [**IROS-14**] N. Hurtós, N. Palomeras, M. Carreras, **A. Carrera**, C. Bechlioulis, G. Karras, S. Heshmati-alamdari, and K. Kyriakopoulos, "Sonar-based chain following using an autonomous underwater vehicle," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS)*, Chicago, Illinois, 2014.
- [**ICRA-14**] S.R. Ahmadzadeh, M. Leonetti, **A. Carrera**, M. Carreras, P. Kormushev, and D.G. Caldwell "Online discovery of AUV control policies to overcome thruster failures" in *IEEE International Conference on Robotics and Automation (ICRA)*, Hong kong, China, 2014.

Acronyms

3D three-dimensional

AHRS Attitude and Heading Reference System

ASC Autonomous Surface Craft

ASV Autonomous Surface Vehicle

AUV Autonomous Underwater Vehicle

BEA Bureau of Investigation and Analysis

BRIEF binary robust independent elementary features

CIRS Underwater Robotics Research Center or *Centre d'Investigació en Robòtica Submarina*

CP Control Policy

CMRE Centre for Maritime Research and Experimentation

CST Constructing Skill Trees

D-H Denavit-Hartenberg

DMP Dynamic Movement Primitive

DS Dynamical System

DP Dynamic Position

DOF Degree of Freedom

DVL Doppler Velocity Log

EM Expectation-Maximization

EKF Extended Kalman Filter

FAST features from an accelerated segment test

FK Forward Kinematics

FOV Field of View

F/T Force/Torque

GPS Global Positioning System

GPU Graphics Processing Unit

GMM Gaussian Mixture Model

GMR Gaussian Mixture Regression

GUI Graphical User Interface

HAUV Hovering Autonomous Underwater Vehicle

HROV Hybrid Remotely Operated Vehicle

HMM Hidden Markov Model

HRI Human-Robot Interaction

I-AUV Intervention Autonomous Underwater Vehicle

IGMM Infinite Gaussian Mixture Model

IIT Istituto Italiano di Tecnologia

IK Inverse Kinematics

IL Imitation Learning

ILC Iterative Learning Control

IMU Inertial Measurement Unit

KLR Kernel Logistic Regression

LbD Learning by Demonstration

LCS Longest Common Subsequence

LIBS Laser Induced Breakdown Spectroscopy

LR Logistic Regression

LWR Locally Weighted Regression

MP Movement Primitive

ORB oriented FAST and rotated BRIEF

PANDORA Persistent Autonomy through Learning, Adaptation, Observation and Replanning

PbD Programming by Demonstration

PID Proportional-Integral-Derivative

PD Proportional-Derivative

p-DMP Parametric Dynamic Movement Primitive

p-GMM Parametric Gaussian Mixture Model

p-HMM Parametric Hidden Markov Model

RL Reinforcement Learning

RANSAC RANdom SAmple Consensus

RFDM Reactive Fuzzy Decision Maker

ROS Robot Operative System

ROV Remotely Operated Vehicle

rpm revolutions per minute

SAUC-E Student Autonomous Underwater Challenge - Europe

SIFT Scale-Invariant Feature Transform

SoG Sum of Gaussian

SSV Seismic Support Vessel

SURF Speeded-Up Robust Features

SVM Support Vector Machine

SVS Sound Velocity Sensor

SLAM Simultaneous Localization And Mapping

TMS Tether Management System

TP Transition Point

UPC Universitat Politènica de Barcelona

USBL Ultra Short BaseLine

UVL Underwater Vision Lab

UVMS Underwater Vehicle-Manipulator System

WLSR Weighted Least Square Regression

List of Figures

List of Tables

List of Algorithms

Contents

Abstract

The interest in performing underwater tasks using AUVs has been growing over the past few decades. The initial focus was on the exploration of underwater areas, performing survey trajectories harvesting data to build bathymetric maps or photo-mosaics. Later the focus of research switched to performing inspections of underwater structures or elements (e.g., pipelines) which required more complex behaviours and localization methods. Advances in these fields have awakened the interest of the community in using AUVs to perform intervention tasks, replacing ROVs and manned submersibles. This replacement offers the possibility of automatizing tasks, increasing the efficiency and repetitively, as well as reducing cost, time, and the equipment needed to perform an underwater intervention. I-AUVs open the door for more companies and scientists to increase the frequency of some periodic tasks and the development of new tasks, limited until now by the cost.

In this thesis, a flexible framework for underwater interventions using a LbD algorithm as a core has been developed. This algorithm allows the robot's user to transfer a skill or knowledge to the I-AUV using a natural and intuitive form, which is the demonstration. The facility and speed in programming a new task gives the robot the ability to perform multiple tasks with minimal effort.

The developed framework for interventions has been tailored to the GIRONA 500 AUV in order to enable it to perform an underwater valve turning task under different conditions. The GIRONA 500 has been equipped with a 4 DOF Manipulator and a sensorized passive end-effector, which includes a camera and an F/T sensor. The system is composed of several modules in order to successfully perform the intervention task. The more relevant modules are: a tightly coordinated control of the vehicle and the end-effector positions; a visual system to detect and localize the valve panel, and also to extract the state of the valves; an appropriate LbD in order to learn and perform the task learned; an appropriate human robot interface to perform the demonstrations; the use of the F/T sensor to guarantee a correct manipulation of the valve; and the possibility of interacting with other algorithms to adjust the behaviour of the AUV during the task.

Throughout this thesis, the experiment developed has been carried out in a mock-up scenario of a sub-sea installation with a valve panel. This scenario is equipped with a

recreation of a sub-sea valve panel and manifold perturbation systems. This panel can be moved to several positions to simulate different panels, the movement of the valves can be limited, and moreover the resistance of the valve to be turned can be gradually changed until the point at which the valve is blocked. The aim of the perturbation system is to recreate any problem that can be found in a real environment, for example an unexpected occlusion of the panel or underwater currents around the sub-sea panel. The difficulty of the task has been increased gradually in order to test the new improvements in the proposed framework. The most relevant result has been obtained in a long-term experiment in which the GIRONA 500 I-AUV has been used in an autonomous mission of approximately 3 hours and has turned more than 30 valves.

Resum

Durant les últimes dècades ha augmentat l'interès en la utilització de Vehicles Autònoms Submarins (AUVs) per realitzar tasques submarines. Una de les primeres tasques desenvolupades va ésser la de recollir dades mentre el vehicle segueix unes trajectòries predefinides per realitzar un estudi posterior de l'àrea coberta, un exemple podria ser la construcció de mapes batimètrics o fotogràfics del fons marí. Més endavant es van utilitzar AUVs per realitzar tasques d'inspecció en estructures submarines o altres elements amb l'objectiu d'inspeccionar el seu estat, un exemple podria ser el seguiment de emissaris submarins. Tots aquests avanços han despertat l'interès de la comunitat per utilitzar els AUVs en intervencions submarines substituint així els Vehicles Teleoperats (ROVs) o els submergibles tripulats. Aquesta substitució ofereix la possibilitat d'automatitzar les tasques, incrementant l'eficiència i la repetibilitat; per altra banda també redueix el cost, temps i equipament necessari per realitzar les tasques. Aquests factors obren la porta a més empreses i científics a portar a terme més tasques fins ara limitades pel cost.

En aquesta tesis s'ha desenvolupat un marc de treball (framework) per a realitzar intervencions submarines amb AUVs basat en un algorisme d'Aprenentatge per Demostració (LbD). Aquest algorisme permet a l'usuari del robot transferir el seu coneixement al vehicle d'intervenció (I-AUV) d'una forma natural i intuïtiva, realitzant demostracions al robot. La rapidesa i facilitat de la programació dóna al robot la capacitat de ser utilitzat en múltiples tasques amb un mínim esforç addicional.

El framework desenvolupat s'ha ajustat a les característiques del GIRONA 500 AUV, amb l'objectiu de que pugui girar vàlvules submarines en diverses condicions. El GIRONA 500 s'ha equipat amb un braç robòtic de 4 graus de llibertat (DOF) i un element terminal passiu, el qual està sensoritzat amb una càmera i un sensor de força i torsió (F/T). El framework inclou mòduls amb les següents funcionalitats: un control coordinat del vehicle i de l'element terminal; un sistema visual per detectar el panell i extreure l'estat de les vàlvules; una interfície robot-humà adequada per poder realitzar les demostracions còmodament; l'algorisme de LbD per aprendre i realitzar les tasques; l'ús del sensor de força i torsió per garantir la correcta manipulació de la vàlvula; i la possibilitat d'interactuar amb altres algorismes, com ara ajustar el comportament aprés segons un algorisme que

analitza l'entorn.

Al llarg de tota la tesis s'ha utilitzat com entorn de desenvolupament un tanc d'aigua amb una recreació d'un escenari d'intervenció subaquàtic on s'han de girar determinades vàlvules d'un panell. Aquest escenari disposa d'una recreació d'un panell de vàlvules submarines i de diversos tipus de sistemes de pertorbacions. El panell pot estar situat en múltiples posicions, simulant panells diferents; per altra banda les vàlvules tenen un moviment limitat i la resistència al gir pot ser modificada fins a bloquejar la vàlvula. El objectiu dels sistemes de pertorbacions és simular dificultats de l'entorn real, com ara oclusions del panell per elements estranys o corrents d'aigua. El grau de dificultat de la tasca s'ha incrementat de forma gradual, per tal de poder provar noves millores en el framework. Els resultats més destacats en tota la tesis s'han obtingut en un experiment de llarga durada, on s'ha utilitzat el mètode proposat durant més de 3 hores de forma autònoma i s'han girat més 30 vàlvules.

Resumen

Durante las últimas décadas ha aumentado el interés en la utilización de AUVs para realizar tareas submarinas. Una de las primeras tareas realizadas consistía en llevar a cabo un conjunto de trayectorias recogiendo datos para estudiar a posteriori esa zona, un ejemplo podría ser la construcción de mapas batimétricos o fotográficos del fondo marino. Más adelante se utilizaron AUVs para realizar tareas de inspección en estructuras submarinas u otros elementos con el objetivo de inspeccionar su estado, por ejemplo el seguimiento de tuberías. Todos estos avances han despertado el interés de la comunidad para utilizar los AUVs en intervenciones submarinas, substituyendo a los ROVs o sumergibles tripulados. Esta substitución ofrece la posibilidad de automatizar tareas, incrementando la eficiencia y la repetitividad; por otro lado reduce el coste, tiempo y equipamiento necesario para realizar las tareas. Estos factores abren la puerta a más empresas y científicos a realizar tareas hasta ahora limitadas por el coste.

En está tesis se ha desarrollado un marco de trabajo (framework) para intervenciones submarinas basado en un algoritmo de aprendizaje por demostración (LbD). Este algoritmo permite al usuario del robot transferir su conocimiento al vehículo de intervención (I-AUV) de una forma natural e intuitiva, mediante demostraciones. La rapidez y sencillez de la programación otorga al robot la capacidad de ser utilizado en múltiples tareas con un mínimo esfuerzo adicional.

El framework desarrollado se ha ajustado a las características del GIRONA 500 AUV, para realizar la manipulación de válvulas submarinas en diversas condiciones. El GIRONA 500 ha sido equipado con un brazo robótico de 4 grados de libertad (DOF) y un elemento terminal pasivo, el cual está sensorizado con una cámara y un sensor de fuerza y torsión (F/T). El framework incluye módulos con las siguientes funciones: un control coordinado del vehículo y del elemento terminal; un sistema visual para detectar el panel y extraer el estado de las válvulas, una interfaz humano robot apropiada para poder realizar las demostraciones cómodamente; el algoritmo de LbD para aprender y realizar las tareas, el uso del sensor de fuerza y torsión para garantizar una correcta manipulación de la válvula; y la posibilidad de interactuar con otros algoritmos, por ejemplo para ajustar el comportamiento aprendido según un algoritmo que analiza el entorno.

A lo largo de toda la tesis se ha utilizado como entorno de desarrollo un tanque de agua con una recreación de un escenario de intervención submarina donde se tiene que girar determinadas válvulas de un panel. Este escenario dispone de una recreación de un panel submarino con diferentes válvulas y de varios tipos de sistemas de perturbaciones. El panel puede estar situado en múltiples posiciones, simulando distintos paneles, las válvulas tienen un movimiento limitado y la resistencia al giro puede ser modificada hasta bloquear la válvula. El objetivo de los sistemas de perturbaciones es recrear dificultades del entorno real, como por ejemplo oclusiones del panel por elementos extraños o corrientes de agua. El grado de dificultad de la tarea se ha ido incrementando de forma gradual, para poder probar las nuevas mejoras en el framework. El resultado más destacado en la tesis ha sido obtenido en un experimento de larga duración, donde se ha utilizado el método propuesto durante más de 3 horas de forma autónoma y se han girado más 30 válvulas.

1 Introduction

This chapter introduces the aim and motivations of this thesis in the development of a complete system in order to adapt an Autonomous Underwater Vehicle (AUV) to perform underwater interventions by means of Learning by Demonstration (LbD). Section 1.1 relates the motivations to replace the extensively used Remotely Operated Vehicles (ROVs) for AUVs and the reason for selecting the LbD over other possibilities. Next, Section 1.2 presents the goal and divides it into different sub-goals. Section 1.3 introduces the context in which this thesis has been carried out. Finally, section 1.4 presents the organization of the document.

1.1 Motivation

The aim of robotics and artificial intelligence is to simplify a humans work in performing difficult, dangerous, and repetitive tasks. This situation becomes more relevant in a hostile environment like underwater. With this purpose in mind humans have developed different systems to overcome their own limitations, and be able to explore, inspect and intervene in the submarine world.

Over the past few decades, the first underwater robotic approach, called ROVs, have improved in such a way that it has been possible to reduce the need of manned submersibles or divers. Nowadays, these vehicles are used on a daily basis on offshore industries bases, as well as in scientific applications. However, to operate a ROV an extensive and expensive support is required, formed by a vessel equipped with a crane to launch and recover the vehicle, a Tether Management System (TMS) to control the umbilical cable, and a Dynamic Position (DP) system in order to follow the position of the ROV appropriately (Figure 1.1 shows a picture of an operational vessel and a ROV). Apart from these elements a dedicated crew is also needed to operate the vessel and the vehicle. All these factors have stimulated research towards the development of AUVs. They are stand-alone platforms that can perform underwater tasks (e.g., gather data close to the sea-floor) without human supervision which avoids the constant supervision of the operation vessel, and the risks associated with the umbilical cable .

To reach a point where AUVs can start replacing ROVs, decades of research in areas like vehicle control, path planning and mission planning have been necessary. These advances have allowed AUVs to perform successful underwater applications such as in marine geology [Yoerger et al.1998, Kelley et al.2005, German et al.2008, Paduan et al.2009], marine biology, [Armstrong et al.2006, Williams et al.2010], underwater archaeology [Foley et al.2009, Bingham et al.2010], fisheries management [Clarke et al.2009], under-ice exploration [Kunz et al.2009], and disaster response [Camilli et al.2010]. However, intervention applications, where the system manipulates the environment, are still mainly performed by ROVs due to the complexity of the task.

(a) Vessel of the DEEPOCEAN company, equipped with a TMS and a DP system working in the off-shore industry.

(b) MAGNUM ROV of OCEANEERING being launched from the launch and recovery cage.

Figure 1.1: These figures show (a) an example of the kind of vessel able to operate ROVs and (b) a ROV used to perform underwater interventions in the oil & gas industry.

The more relevant scenarios for underwater intervention tasks are marine rescue, deep water archaeology, marine science, and the oil & gas offshore industries. In these scenarios, most of the intervention operations are being undertaken by manned submersibles or ROVs, both equipped with robotic manipulators. Below some commonly developed tasks in each field are detailed, with examples shown in Figure 1.2.

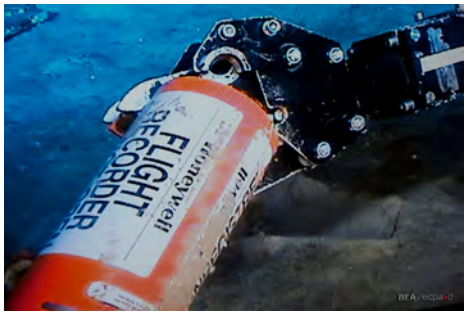
Marine rescue: the search and recovery of a flight data recorder in the black-box of a crashed planed is a good example of a marine rescue task where Intervention Autonomous Underwater Vehicles (I-AUVs) could replace ROVs. I-AUVs can be easily mobilized due to their small dimensions and weight. Also, they can be launched from inexpensive vessels not dedicated to intervention or oceanographic missions. These features reduce the time and cost of this kind of operation.

Deep water archaeology: two main archaeological interest topics remain hidden in the depths of the sea. First, human settlements, from 6000 to 16000 years ago, submerged in the sea after the last Ice Age which hide clues of our historical heritage. Second, shipwrecks, in particular deep-sea wrecks, which are also an important source of historical information. Archaeological institutions are interested in documenting these settlements with sea-floor maps and recovering some parts for study. However, most archaeological institutions are small and can not cover the cost of ROV deployment and operations. I-AUVs can reduce this cost making more excavations possible.

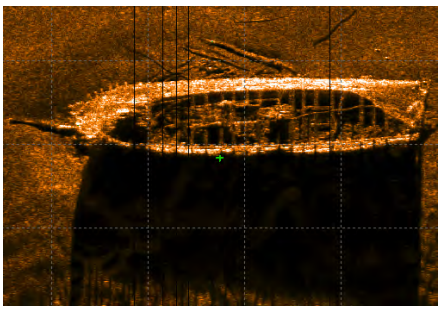
Marine science: to study ocean and sea behaviours and status over long periods of time, scientists have developed artificial infrastructures, which are installed in the seabed and use instruments and sensors to harvest data about the environment. These observatories need maintenance in terms of reconfiguration (connecting/disconnecting instruments), installation of new equipment, replacing batteries, instrumentation de-fouling, and, for isolated observatories, recovery of data. Again, the introduction of I-AUVs can facilitate these operations by reducing the costs associated.

Oil & gas industry: this industry is one of the principal users of underwater robotics technology as some oil & gas reservoirs can be found under the sea-bed. At this moment, they use work-class intervention ROVs to routinely inspect and repair submerged infrastructures. In order to reduce the cost and allow more frequent inspections, they have started to consider the use of AUVs for inspection of oil wells, chains, risers, etc. [Gilmour2012]. On the other hand, the use of I-AUVs to perform simple tasks like opening valves or plugging a hot stab is also under study. The use of I-AUVs reduces the cost of operations and, in this case, can eliminate human errors, usually provoked by fatigue and stress during the interventions.

As can be appreciated, I-AUVs will proportionate time and cost benefits to the final users and can be applied to several different operations. However, the current state of the art

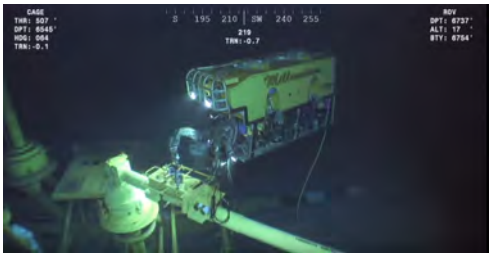


(a) Image recorded during the blackbox recovery of Air France Flight 447. The BEA used a REMORA 6000 ROV during the recovery mission.



(b) Side-scan sonar image of shipwreck Aid in Estonia.

(c) OBSEA is an expandable sea-floor observatory, developed and maintained by the UPC located on the seabed at Vilanova i la Geltrú (Spain).



(d) MILLENION ROV of OCEANEENGINERING performing a maintenance intervention on an oil & gas installations.

Figure 1.2: Four sub-sea images representing the main intervention tasks.

is still too underdeveloped for these environments and tasks, unlike in other industries (e.g., automotive industry, food industry, and electronic industry) where robots are extensively used in handling operations, welding, assembly, dispensing, and processing. The main difference between the two worlds is the fact that these industries have an almost absolute control of the environment, which is completely known by the robot. This simplifies the intelligence of the robot and a simple behaviour can perform complex and repetitive tasks without difficulty. Instead, underwater intervention tasks take place in an uncontrolled environment which can not be changed or adapted to the needs of robots, so the robots need to be flexible and intelligent to adapt themselves to the changing environment.

In recent years, the robotics community and the industry sector have focused their efforts in order to move robots from a controlled and closed environment to an open work space where humans and robots can work together. This has been possible with the appearance of compliant robotics systems which allow the possibility of this shared workspace with minimal risk for a human being. Some examples are Baxter of the Rethink Robotics [Rethink-Robotics2016], YuMI of ABB [ABB2013], LBR-4 of KUKA Robotics [KUKA-Robotics2016], and UR5 of Universal Robots [Universal-Robots2017] (see Figure 1.3). All of these are light compliant robots, and can be easily installed in new

environments to perform a new task.

(a) Batxer cooperative robot of Rethink Robotics.

(b) YuMI cooperative robot of ABB.

(c) LBR-4 cooperative robot of KUKA

(d) URL-5 cooperative robot of Universal Robots

Figure 1.3: Four examples of cooperative robots designed to work in environments with humans.

The advantage of these robots is in their flexibility, as they can be easy moved and re-programmed by the factory operator without the need of any robotics knowledge. However, these advantages came at a price. The loss of speed, force and with a precision similar to that of a human operator. The most typical teaching is done as follows; the Robot manipulator is in gravity compensation mode, where the arm can be easily moved by the operator to a set of positions that will compose the task. The robot records these positions and, by using a motion planing algorithm [Hwang et al.2003, Rosell et al.1999, Sacks2003, Latombe1991], can reproduce the trajectory.

Furthermore, alternative and more innovative techniques with a similar propose are being studied, which is to transfer knowledge between a human and a robot in a more natural form. These methods are known as LbD, Programming by Demonstration (PbD) or Imitation Learning (IL) [Billard et al.2008]. In this case, the whole task or trajectory is learned and can be more precise than a motion planing algorithm, since it extracts a generalized model of the whole task instead of only some points. Also, the learned model is more adaptable and robust than motion planing. Moreover, the operator doesn't need to indicate any position, he only has to perform the task a few times.

Back in the underwater environment, ROVs are vehicles which are usually used to develop a certain task for a specific period of time. Hence, to offer a real alternative to a ROV, the I-AUV should proportionate a flexible and easy method to program the vehicle for the desired task. Following the steps used by other robot industries and research, we

believe the I-AUV should be programmed by the final user without the need of an expert robotic programmer operator.

In this thesis, we propose to use the LbD instead of motion planing for the following reasons. The LbD offers a more complete and robust solution since it learns the whole task instead of a few points, also it is difficult to define these points correctly in an unstable underwater environment. Going through a number of demonstrations offers more information about the task, and allows for small errors over the whole trajectory since they are compensated for other demonstrations. On the other hand, both methods have the common difficulty of performing a demonstration of the task in an underwater environment. To perform these demonstrations several options can be studied: performing the first intervention using the I-AUV as if it was a ROV; using a mock-up environment to teach the task to the I-AUV; using a simulator of the I-AUV and the task environment to record the demonstrations.

1.2 Objectives

Once the motivation has been set, we state the overall goal of this thesis as follows:

Development of a complete intervention framework for an I-AUV, based on an LbD and tailored to perform an underwater valve turning task under different conditions.

LbD Method: Explore through the literature a LbD method with the proper characteristics to be used in the complex underwater environment considering the limitations of an I-AUV to perform a free-floating underwater intervention.

Integration of the LbD in an I-AUV: Definition of a complete framework for intervention using the LbD as a core method. This framework integrates the AUV control architecture with the tailored LbD method. Some examples of the required integration are: the definition and format of the data shared, the development of specific modules to control the vehicle and the manipulator, and the implementation of dedicated perception systems to understand the state of the environment.

Collaboration in the development of the GIRONA 500 I-AUV: Development of the software (low- and high-level) necessary to control a commercial manipulator and integrate it in an AUV. Collaboration in the development process participating in the design of the end-effector, the necessary sensors in the end-effector, development a vision based system to detect the manipulator position, and design of the mock-up sub-sea panel.

Participate in the European Persistent Autonomy through Learning, Adaptation, Observation and Re-planning (PANDORA) Project: Participate actively in the PANDORA Project in the scenario of the valve turning task and collaborate if required in other tasks. This work includes the collaboration with the other partners in

order to integrate their algorithm in the I-AUV, development of individual and collective experiments, attending to workshops, divulgation of the project, and elaboration of reports and deliverables for the project.

Experimental validation: The validation of the several methods proposed during the thesis, following an incremental procedure. Experiment will start from a simulated environment, and end in a water tank with the I-AUV under external perturbations. The water tank conditions will be controlled but throughout the experiments the difficulty will increase gradually adding several external perturbations systems.

1.3 Context

The research for this thesis has been conducted at the Underwater Robotics Research Center or *Centre d'Investigació en Robòtica Submarina* (CIRS) of the Computer Vision and Robotics (ViCOROB) Institute of the Universitat de Girona. Research in underwater robotics has been ongoing there since 1992, supported by several Spanish and European research programs. The group has developed several AUV prototypes: GARBI [Amat et al.1999], a vehicle originally conceived as a ROV that was restyled as an AUV; URIS [Batlle et al.2005], a lightweight AUV; Ictineu [Ribas et al.2007], which won the first Student Autonomous Underwater Challenge - Europe (SAUC-E) competition in 2006; SPARUS, which championed SAUC-E in 2010 ; GIRONA 500 [Ribas et al.2012], a reconfigurable AUV that has been used in the experimental part of this thesis; SPARUS II a restyled version of the SPARUS presented as a commercial platform and winner of several tasks in the multi-domain competition EURATHLON 2014 and 2015 [Carreras et al.2013].

Research at CIRS revolves around AUV applications, and has focused on control architectures [Ridao et al.2002] [Palomeras et al.2012], model identification [Carreras et al.2003], machine learning [Carreras et al.2001, El-Fakdi and Carreras2013], mission control [Palomeras et al.2006], AUV intervention [Ribas et al.2012, Palomeras et al.2014b], Simultaneous Localization And Mapping (SLAM) [Ribas et al.2008, Mallios et al.2014b, Palomer et al.2016], single beacon navigation [Vallicrosa et al.2014], and path planning [Hernández et al.2011, Galceran Yebenes2014, Hernandez et al.2015]. Together with recent efforts in bathymetry mapping [Zandara et al.2013, Gracias et al.2013] and sonar scan matching [Mallios et al.2014a].

In conjunction with CIRS, the Underwater Vision Lab (UVL) from ViCOROB devotes its research in underwater computer vision, where most of the images are obtained using an AUV. The UVL has cover topics such as those developed with the research performed with the AUV large-scale underwater mosaicing [Ferrer et al.2007], global alignment techniques for optical mapping [Elibol et al.2011], image blending [Prados et al.2014], three-dimensional (3D) scene modelling [Nicosevici et al.2009], image-based classification [Shihavuddin et al.2013], mosaicking using forward looking sonar images [Hurtós Vilarnau

et al.2015], and recently introduced underwater omnidirectional cameras [Bosch et al.2015].

This thesis continues the efforts dedicated to machine learning in conjunction with the recent efforts in underwater intervention tasks. Moreover, the work presented herein has contributed to the following projects in which CIRS has participated:

- FP7-7 EU Project PANDORA: Persistent Autonomy through Learning, Adaptation, Observation and Re-planning (Ref FP7-ICT-2011-7-288273), funded by the European Commission.
- MINECO Project COMAROB: Robótica cooperativa Marina para el mapeo acústico y la intervención (Ref DPI2011-27977-C03-02), funded by the Spanish Ministry of Science and Innovation.

Finally this thesis has also benefited from a research stay in the Learning and Interaction lab of the Advance Robotics Department at the Istituto Italiano di Tecnologia (IIT) (Genoa, Italy). The Learning and Interaction laboratory is a science and engineering laboratory research distinguished for its innovations in Reinforcement Learning (RL) and LbD applied to humanoid robotics and manipulators [Bruno et al.2015,Kormushev et al.2013,Kormushev et al.2010, Calinon et al.2007].

1.4 Outline

The remainder of this Doctoral dissertation has been organized in five Chapters:

Chapter 2 State of the Art: this focuses on the study of the state of the art in LbD. First, it introduces machine learning techniques in order to establish the context of the LbD. After the commune elements of the LbD methods are presented, the methodologies to develop this elements are studied. Finally, four representative LbD methods are explained in detail and a list of relevant articles for this work are summarized in a table.

Chapter 3 LbD for an I-AUV: the underwater intervention framework is presented starting from the core. In this case, the LbD method called Dynamic Movement Primitive (DMP) is introduced in detail. Next, the different elements required to complete the framework are explained including the interaction between them and the DMP. Finally, the different modifications performed to the DMP in order to adapt it to the I-AUV and an underwater environment are detailed.

Chapter 4 Experimental Environment: it starts by presenting the valve turning task proposed in the Thesis and in the PANDORA Project. Afterwards, a brief look at the state of the art of the more relevant projects related to underwater intervention are presented. Moreover, it also includes all the information on the different elements

used to develop the valve turning, including hardware and software (i,e,. the GIRONA 500 AUV, the ECA ARM 5E manipulator, and the custom end-effector).

Chapter 5 Experimental Results: this presents all the experiments performed throughout this thesis. For each experiment, the results obtained and the lesson learned are explained. The experiments follow an incremental approach starting from an initial simple case, just to test the concept, and finishing with an experiment under high perturbations in the underwater tank.

Chapter 6 Conclusions: It summarizes the work completed in the whole thesis, reviewing the main contributions and presenting compelling areas for future work.

2

State of the art

A general overview about machine learning techniques with special attention to LbD methods is presented in this chapter. Section 2.1 introduces the main machine learning methodologies and Section 2.2 defines the concept of LbD and its variations. Finally, Section 2.3 presents and compares some relevant projects in the robotics domain closely related to this thesis.

2.1 Machine Learning

To understand the concept of machine learning, first it is necessary to correctly define the word *learning*. The Oxford English Dictionary defines *learning* as "The acquisition of knowledge or skills through study, experience, or being taught". Hence, in an informal manner, we can say that machine learning tries to make possible for a machine to change its structure or program in order to acquire new knowledge or a skill by means of data analysis using its own experience or by being taught. More formally, machine learning is defined as a scientific discipline dedicated to the design and development of algorithms that allow computers to evolve behaviours based on empirical data obtained through sensors or databases.

Since machine learning is a broad ranged discipline, below are listed some of its most relevant uses:

- Finding or approximating a relationship between a set of specific input/output examples.
- Discover important relationships and correlations hidden in large data sets (e.g., *Big data*) to make the data understandable and useful for other purposes.
- Learning through experience parameters or characteristics of the environment, unknown at the design phase of the algorithm. In this way, the method can be flexible to new situations.
- Capture and encode efficiently large quantities of knowledge difficult to encode by a human being.
- Adapting techniques to over time changing conditions by constant adjustments of parameters.

Machine learning algorithms can be organized based on the desired outcome of the algorithm.

Supervised Learning: generates a function that maps inputs to desired outputs or labels. For example, a classification problem.

Unsupervised learning: refers to the problem of trying to find hidden structures in unlabeled data.

Semi-supervised learning: combines both supervised and unsupervised learning to generate an appropriate function or classifier.

Transduction: or *transductive inference*, tries to predict new outputs on specific and fixed (test) cases from observed specific (training) cases.

Reinforcement Learning (RL): learns how to act, when given an observation of the world. Every action performed produces some impact on the environment and the environment provides feedback in the form of a reward that guides the learning process.

Learning by Demonstration (LbD): learns a new skill through direct training. A *teacher* demonstrates the skill to the machine or robot. To do this, the *teacher* performs the demonstration by herself or by using the robot itself.

Among all these kinds of learning algorithms, the two most suitable for the task of grasping a valve and turning it are the **RL** and the **LbD**. Both methods generate a model of the skill capable of adapting to the environment. Moreover, these two methods have the advantage of avoiding the necessity of adjusting a large number of parameters, usually by finding the most general controller.

However, **RL** algorithms present an important handicap: because this algorithm has to explore all the possible solutions to obtain feedback, it requires multiple reproductions and therefore a large amount of time to learn new tasks.

Even though simulators can be used to speed up this process they need to have high accuracy and precision, to avoid erroneous behaviours in a real environment.There are new tendencies which try to mix the exploration in simulated and real environments to try to find a trade-off between both phases [Cutler et al.2015].

On the other hand, present day underwater interventions are performed by ROVs, piloted by human operators, hence, if the **LbD** method is used, we can profit from these experts and *transfer* their current knowledge to the learning system. Additionally, LbD will simplify the learning of new intervention tasks because, once the learning system is ready, the operator should only need to perform the task to learn a few times for the system to learn it.

Is worth to mention, that in the recent years with the increase of the computational power using Graphics Processing Units (GPUs), a new machine algorithm methodology called **Deep Learning** has erupted with force. It has started boosting the development in several applications for robotics, like visual perception, object detection, control of mobile robots, speech recognition, natural language processing, etc [Lei and Ming2017]. This method can be applied in supervised, unsupervised and reinforcement learning. It uses a cascade of many layers of non-linear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input. This methodology exploits the idea of hierarchical explanatory factors where high levels, more abstract concepts, are learned from the lower levels.

2.2 Learning by Demonstration (LbD)

LbD is a machine learning technique designed to transfer knowledge from an expert to a machine, rather than analytically decomposing a problem and manually programming a desired behaviour [Dautenhahn and Nehaniv2002,Billard et al.2008]. This type of algorithm follows three sequential phases (see Figure 2.1) : First, a set of *demonstrations* of the task are recorded; second, the algorithm *learns* by generalizing all demonstrations and creating a *model*; third, the algorithm loads the model and uses it to *reproduce* the task. The model can be reused and the demonstrations can be discarded.

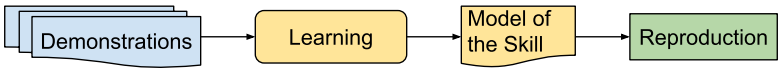


Figure 2.1: Schema of the LbD.

This structure is the most common over all LbD methods, despite the fact that there are other variations depending on the method used or its implementation. This schema will be used as the base to explain the details of the different techniques in the following subsections.

2.2.1 Skill Demonstration

To demonstrate a new skill to a robot, it is necessary to establish communication between the human expert and the machine. Human-Robot Interaction (HRI) [Goodrich and Schultz2007] [Toris et al.2012] is a multidisciplinary field focusing on the development of methodologies to allow interaction between humans and robots. HRI includes different aspects like: engineering (design and construction of the robot), computer science (enable the communication), social sciences (extracting hidden information) and humanities (make sense of the action).

Therefore, HRI provides appropriate methodologies and strategies to be followed by the *teachers* of the LbD algorithms. Figure 2.2 shows a possible classification based on how the demonstrations are executed and how they are recorded.

The main goal of the *demonstration* phase is to record the interaction in a way that the LbD algorithm can learn a model or the relationship between the *state* and the *action* variables [Argall et al.2009]. It is necessary to define correspondence between the teacher demonstrations and the robot.

In order to deal correctly with the correspondence problem, we have to consider if the *state* or the *action* variables recorded during the demonstration phase are equivalent to the *observed* variables and *executed* commands in the reproduction phase. If the *state* variables need to be converted into a reproduction of the environment, it is called *embodiment mapping*, and if the *action* variables need to be converted, it is called *operation mapping*.

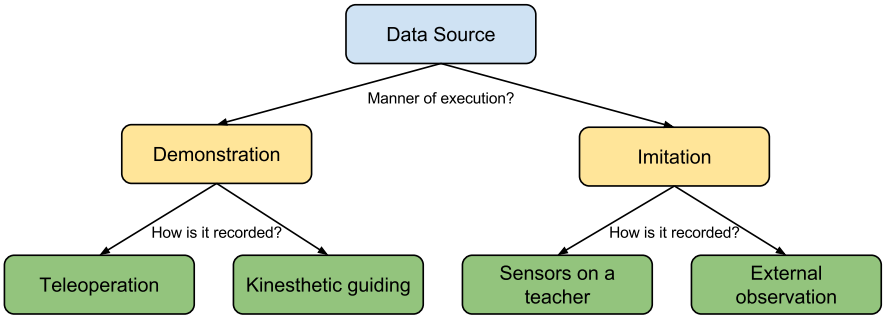


Figure 2.2: Categorization of different approaches to build the demonstration dataset.

For instance, if the goal is to learn the trajectory performed by a mobile robot, the *state* in the demonstration phase is the position and the *actions* are the motor commands and the orientation of the steering wheel. Several methods to perform a demonstration will be discussed next using this simple example to illustrate the categorization.

2.2.1.1 Demonstration

During the demonstration phase, the skill to be learned is performed by a human expert using the actual robot in a real environment. For this reason the demonstration can be recorded without need of any *embodiment mapping*. Two demonstration methods can be applied: Teleoperation and kinesthetic guiding.

Teleoperation: The expert operates the robot and it records the commands executed. Hence, there is no need for *operation mapping*. In our example, the teacher drives the robot using a remote control and the algorithm directly records the actions sent to the vehicle. Some examples can be found in the following publications [Konidaris et al.2012, Wilson and Bobick1999].

Kinesthetic guiding: The robot is in a passive mode and the teacher applies external forces to perform the task, so the commands to match or mimic the movements are inferred from the sensor data. This process will correspond to *operation mapping*. In the proposed example, the robot is in a compliant mode where the teacher will be pushing it to follow the desired trajectory. The algorithm extracts the actions from the velocity and orientation detected by its sensors. Some examples can be found in the following publications [Hersch et al.2006, Calinon et al.2013, Manschitz et al.2015, Kormushev et al.2011, Kronander and Billard2012, Krüger et al.2012, Ude et al.2010, Forte et al.2012].

2.2.1.2 Imitation

In imitation, the skill demonstration is performed using a different platform from the robot learner. For this reason, it is necessary to perform an *operation mapping*, converting the demonstrated actions into the robot's actions. In the proposed demonstration examples,

instead of using the robot to perform, a car will be used.

Sensors on Teacher: The sensors are located on the demonstration platform or the teacher's body to record the teacher demonstration. In this case, there is no need for *embodiment mapping* since the sensors used are the same as will be found on the robot. In the proposed example, the teacher drives a car performing the trajectory to be taught to the robot. Sensors have been installed on the car to record the acceleration commands and the steering wheel orientation. In this case, the action variables are mapped to match the robot's action variables. Some examples can be found in the following publications [Hovland et al.1996a, Hoffmann et al.2009, Matsubara et al.2011].

External Observation: The sensors used to record the actions are external to the demonstration platform. These sensors are not present in the robot, hence an *embodiment mapping* is needed. In our example, the car will possess of a Global Positioning System (GPS) sensor, so from the positions over time, the algorithm estimates the actions performed by the teacher. Like in the previous method, these actions have to be mapped to proper commands for the robot.Some examples can be found in the following publications [Kruger et al.2010].

2.2.2 Number of skill demonstrations

The number of demonstrations needed to teach a new skill depends on the complexity of the task and also on the desired degree of generalization that must be achieved in the learned model. What is always paramount is that the set of demonstrations cover all the possible solutions to represent the demonstrated skill.

However, the number of demonstrations also depends on how complex the task demonstration is to perform. If the skill is complex, the teacher produces small errors during the demonstrations, an elevated number of demonstrations will eliminate these errors in the learned model. However, discarding demonstrations with a large number of errors will lead to better models.

Oftentimes, a skill can have parts with multiple approaches or solutions. The teacher should decide if it is better to include all of them with different demonstrations or to focus on one particular solution, reducing the number of demonstrations.

According to the authors experience of teaching skills from 4 to 8 Degree of Freedom (DOF), the number of demonstrations needed is between 2 and 10. In general, more demonstrations does not necessarily imply that the model obtained is better. If the demonstrations are similar, they do not contribute to getting a better model.

Related to the number of demonstration there are different methodologies to incorporate in the demonstrations for the LbD algorithm:

- **Batch of Demonstrations** All the demonstrations are recorded before the learning phase. They are only used once to generate the skill model and then are ignored.

	Symbolic Learning	Trajectory Learning
Advantages	Allows learning conditional actions, loops and sequences of actions to resolve a complex task with simple actions.	Allows learning very different types of movement and gestures, unknown by the robot.
Drawbacks	Requires a set of a priori known actions as well as a system to identify and classify this knowledge. It also requires a low control system to execute these actions.	It is not suitable for learning complex high-level skills which need different reactions depending on the state.

Table 2.1: Main advantages and drawbacks between Symbolic Learning and Trajectory Learning

To improve the learning model, a whole new batch must be done, including or not including the old demonstrations, and the model is learned again.

- **Incremental Demonstrations** The demonstrations are incrementally introduced into the learning algorithm until the model reaches the desired accuracy. Therefore, the teacher can decide if it is necessary to perform more demonstrations or not. In this case, the demonstrations are used to improve the model and once used, they are discarded.

Most of the methods use a batch of demonstrations, due to the fact that they require less supervision. However, there are LbD techniques with both implementations (e.g., Gaussian Mixture Model (GMM) batch [Calinon2009] and incremental [Calinon and Billard2007] demonstrations) and also hybrid solutions using an initial batch of demonstrations that can be improved by extra demonstrations [Chernova and Veloso2009, Silver et al.2012].

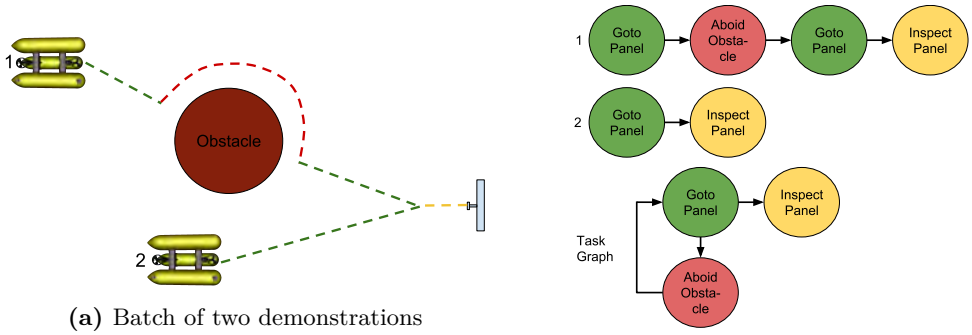
2.2.3 Skill Learning

LbD can be used to learn different levels of a skill using two approaches. The first one, named *symbolic learning*, is a high level approach in which the algorithm learns how to execute a sequence of previously known actions to perform the task at hand. In the second approach, named *trajectory learning*, the algorithm follows a low-level approach that tries to generalize the evolution of a set of variables to represent a trajectory and elements associated with the trajectory. Table 2.1 shows a comparison between the two main skill learning approaches.

2.2.3.1 Symbolic Learning

As mentioned in Table 2.1, using symbolic learning requires a prior-knowledge of actions. During the demonstrations, the method segments the skill using these actions. Each skill fragment will be classified as an action plus a set of parameters.

There are different methods to encode a skill. The classical method to represent the skill structure is using a graph [Ekvall and Kragic2006]. For the states classification, a well known approach is to use a Hidden Markov Model (HMM) [Hovland et al.1996b] [Abbas and MacDonald2011b]. Figure 2.3 shows an illustrative representation of the use of a graph to represent a skill.



(a) Batch of two demonstrations

(b) Graph Representation

Figure 2.3: Illustrative example of a skill representation using symbolic learning with a graph representation.

Not only sequential graphs are used in the literature to represent symbolic learning. In [Konidaris et al.2012], the author proposes tree graphs where the goal is the tree root and each branch represent a way to complete the task. This method is able to represent very different solutions, but it is not able to represent loop structures. Figure 2.4 shows an example of a tree representing the task used in Fig. 2.3. The loop limitations of the method can be appreciated in the example.

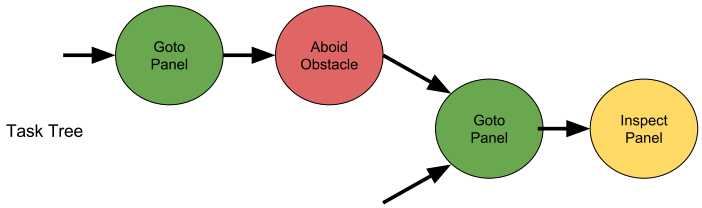


Figure 2.4: Tree representation of the demonstration presented in Figure 2.3a.

2.2.3.2 Trajectory Learning

Trajectory Learning follows a low-level approach that tries to generalize the evolution of a set of variables without any a priori knowledge. These variables usually represent a trajectory or information associated to represent a trajectory. Two different learning spaces can be used (see Fig. 2.5):

- **Joint space:** The method learns the current position or the commands sent to each motor or joint (e.g., each joint position in a manipulator).

- **Task space:** The method learns the trajectory in the frame defined by the task (e.g., the Cartesian position of an end-effector).

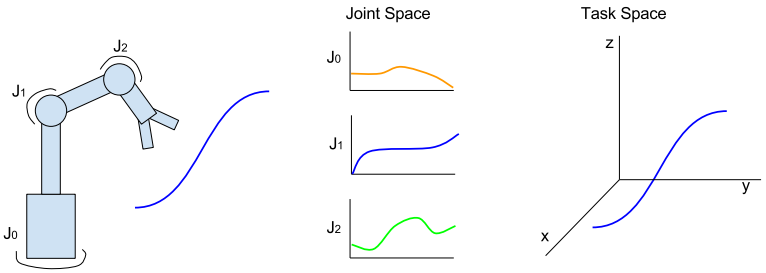


Figure 2.5: Trajectory learning diagram in which joint space and task space are represented.

Almost all learning techniques can be used in both spaces. Moreover, the learning in the *task space* can be combined with other values associated with the trajectory like Stiffness [Kronander and Billard2012] or Force [Kormushev et al.2011].

2.2.4 Skill Encoding and Reproduction

Closely related to what is learned, it is necessary to define how the skill is encoded and how it will be extracted from the learned model to reproduce it correctly. Therefore, if the learning is done in a symbolic level, a classification algorithm is used, otherwise, if the learning is done in a trajectory level, a regression or dynamic algorithm is used. In some cases, LbD methods can use a classification algorithm only to encode the skill, and a regression algorithm to reproduce it. For example, [Calinon et al.2010a] uses an HMM that is commonly used as a classification algorithm, but extracts the information with a Gaussian Mixture Regression (GMR) converting the approach to a Regression method.

2.2.4.1 Classification algorithm

To use the classification algorithm, the skill is encoded in a sequence of states, where each state is defined by a primitive action. Hence, to reproduce the skill, the algorithm classifies the current state and, according to the model, uses the primitive actions associated to it. Section 2.3.1 introduces the technique to learn a task graph through Longest Common Subsequence (LCS) [Abbas and MacDonald2011a] and Section 2.3.2 presents the technique to learn a Global task graph from a set of kinaesthetic demonstrations [Manschitz et al.2015].

2.2.4.2 Regression algorithm

Regression algorithms encode the skill associating the state with a continuous action space. Like in classification encoding, the relationship between states and actions is encoded in a model, which in the reproduction phase is extracted through a regression method.

The regression model can be executed in run-time (dynamically) or prior to the run-time (statistically). Section 2.3.3 introduces the encoding using GMM and reproduction through GMR [Calinon2009] while Section 2.3.4 presents the classical implementation of the DMP technique [Ijspeert et al.2013]. These two algorithms are closely related with the method proposed in this thesis and introduced in the next chapter.

2.3 Relevant examples of LbD algorithms

This section presents in detail four different LbD algorithms and, at the end, a general overview of methods related to this thesis.

The selected methods illustrate the two types of level of learning and encoding associated. The first two methods are examples of symbolic learning and the last two methods represent trajectory learning algorithms.

The skill demonstration is independent of the level of the learning and the encoding used. Moreover, the selection of a skill demonstration method depends on the task performed and the environmental conditions. For this reason, this section doesn't present different specific examples of the demonstration classification.

2.3.1 Task graph through Longest Common Subsequence (LCS)

The LCS algorithm proposed by [Abbas and MacDonald2011a] is one of the potential techniques to build generalized task graphs. The proposed approach uses an LCS of multiple sequences to better represent the generalization of the demonstrated task.

The technique requires symbolic demonstrations or a previous method that encodes the demonstration into a sequence of predefined symbols (representing actions primitives). This demonstration contains:

- ***Consistencies***, which represent a set of common symbols with consistent sequential relationships in all the demonstrations.
- ***Variabilities***, which represent two different types of symbols. The common elements with flexibility in the order, and the uncommon symbols that appear only under certain conditions.

The aim of this technique is to extract consistences including at least one in each path of the task graph. To understand the method, we will introduce some necessary definitions for a task graph representation:

A ***Task graph*** is a directed graph $G = (N, R)$ described by N being the set of nodes and $R \subset NxN$ being a set of direct paths between nodes where $(x, y) \in R$ implies a direct path from x to y. The *start node* is a node without a predecessor and the *end node* is a node without a successor.

A set of ***sequences*** is defined as $S = \{s|s = (s_1, s_2, \ldots, s_n)\}$, and a G task graph complimg with an S if : $x \in N \Leftrightarrow x \in s \in S$ and $(x, y) \in R \Leftrightarrow (x, y)$ is a substring of at least one sequence $s \in S$.

A *sequence* or ***path*** $p = (p_1, p_2, \ldots, p_n)$ is derived from G if : $p_i \in N \forall p_i \in p$; $(p_i, p_{i+1}) \in R$ for all substrings of p; and p_1 is a *starting node* of G and p_n is an *end node* of G. P_1 and P_2 are sets of all the derived paths from the graphs G_1 and G_2 respectively.

More general graph, G_1 is more general than G_2 if size of P_1 is greater than the size of P_2.

Simpler general graph, G_1 is simpler than G_2 if $G1$ has fewer branching nodes than $G2$.

More accurate graph, G_1 is more accurate than G_2 if P_1 has more elements than P_2 which follow the same *consistencie*.

Thus, a generalization problem is formally stated as, "Given a set of sequences S, derive the most general task graph $G = (N, R)$ from S while preserving the accuracy and simplicity". So a good generalization approach must identify the *consistencies* precisely and put them in the task structure appropriately to ensure that each path in the graph follows the *consistencies*.

The time and memory usage complexity of the LCS is $O(\prod_{i=1}^{m} n_i)$, where m is the number of sequences and n is the number of symbols in a sequence. The authors propose a limitation of $m \leq 10$ and $n_i \leq 15$, which is enough to represent task demonstrations and avoid complexities issues. The proposed algorithm follows three steps:

A. Finding common symbols and recording corresponding indices. For each common symbol x a tuple in m is created with the indices of the element in all the demonstrated sequences, i.e. $C = \{c_j = (j_1, j_2, \ldots, j_n)|s_1(j_1) = s_2(j_2) = \cdots = s_m(j_m)\}$.

B. Penalizing the elements of C. Common elements not necessarily appear in the same order in the sequences. Consequently, the C elements are penalized to distinguish the ones appearing in the same order (i.e. the *consistencies*). Hence, the LCS could be considered as the problem of finding the largest subset C_{LCS} of C so that the elements of C_{LCS} correspond to the LCS of S. To compute the penalty, two new concepts introduced below are needed

Non-intersecting two tuples $(c_i, c_j \in C)$ are said to be non-intersecting ($\updownarrow$) if and only if all the elements of c_i are either less than or greater than the corresponding elements of c_j $(c_i \updownarrow c_j \Leftrightarrow ((c_i(k) < c_j(k)) \vee (c_i(k) > c_j(k)))$.

Intersecting two tuples $(c_i, c_j \in C)$ are said to be intersecting ($\ddagger$) when they are not non-intersecting.

Penalty the magnitude of penalty (R_i) for $c_i \in C$ is the number of elements of C which intersect c_i.

C. Selecting or deleting the tuples to find LCS. The selection and deletion process consists of the following steps:

1. Pick the tuple $x \in C$ with the lowest penalty and add it to the C_{LCS}
2. If R_x is zero, jump to step 1, otherwise jump step 3.
3. Find all intersections (y) with x in C and delete them.
4. For each deleted tuple (y) in step 3, reduce the penalty magnitude of each tuple intersected with y and jump to step 1.

These steps are repeated until C becomes empty and C_{LCS} gets the maximum number of non-intersecting tuples.

Finally, using the C_{LCS} nodes as branching nodes, the task graph can be constructed. Figure 2.6 represents the whole process to extract the LCS from three demonstrations and obtain the task graph representation with the extracted *consistencies*.

2.3.2 Global task graph for Movement Primitive (MP) from kinesthetic Demonstration

The approach proposed by [Manschitz et al.2015] presents an interesting technique because it performs symbolic and trajectory learning at the same time. The division of the task into small parts reduces the complexity of the trajectory and lets the algorithm focus on the symbolic level of the task. The main contribution is to take into consideration the history of classifying and detecting the changes between different MPs as well as the influence of the previous MP on the next MP. The demonstrations are done at trajectory level and divided in the symbolic level by the teacher.

The method is composed of three stages. First, the *goal parameters*, which define the target position of each MP, are learned from the demonstrations. Second, the task graph representation is created according to the demonstrations. Third, the transition behaviours are learned as a multi-class classification problem. The transition behaviours decide, according to the current MP, state variable and history which MP will be active. Figure 2.7 shows the three stages.

A. Learning MP Parameters. The parameters of the MP are defined by the goal/end-position of the corresponding segment of the demonstration. Extracting the goal from the different MPs is not straight forward because the teacher has not been able to reach the goal position in all the demonstrations. For this reason, assuming the goal position as the average of all the end points is not sufficient.

All the segments of the demonstrated trajectories are approximated by a linear function and the goal for each lies on its future path. The goal for the MP is found by arbitrating between all the expected goals and finding the best compromise.

Several methods can be used to learn the MP. The authors propose using Weighted Least Square Regression (WLSR) to learn the parameters (a, b) to model the linear function $f(t_i) = at_i + b$. The WLSR allows weighting the data in a non-uniform way in order to

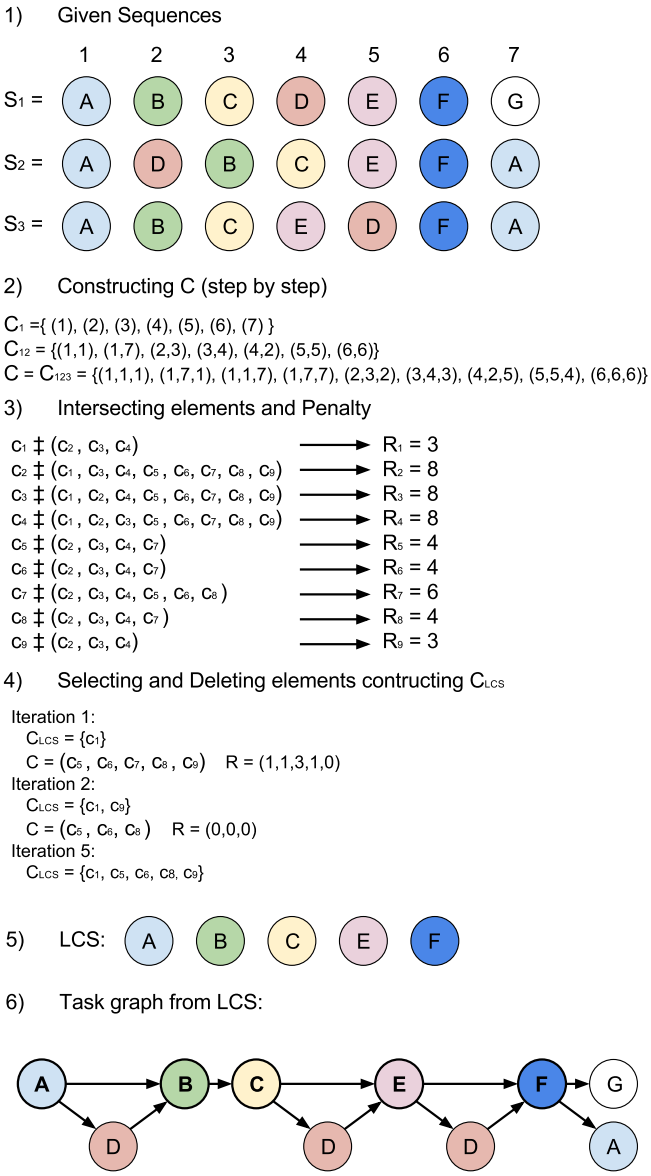


Figure 2.6: Example of a task graph generation using the LCS algorithm to extract the *consistencies*. The demonstrations have been divided into segments according to common features.

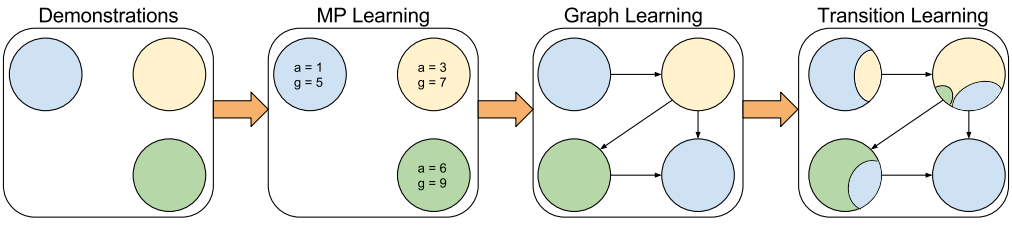


Figure 2.7: Illustrative example of the different steps to learn a skill using a global task graph for MP from kinesthetic demonstrations.

adjust the model. In this case, the weight is stronger in the final part than in the initial part, hence, the influence of the preceding movement is reduced.

B. Learning Graph Representation. The method can use a *local* or a *global* sequence graph, but the most advantageous is the *global* method that is explained below.

First, a sequence graph is a direct graph in which each node n_i is linked to an MP. Transition $t_{k,l}$ connects node n_k to node n_l if a Transition Point (TP) exists. T is defined as the transition matrix composed of all the $t_{k,l}$ and has the size of $m \times m$, where m is the number of nodes in the graph.

One essential characteristic of the *global* sequence graph is that an MP can be linked to more than one graph state. Thence, the same MP can have different TPs depending on the situation.

To create a global sequence graph, three steps have to be performed:

1. Create an acyclic graph T_j for each demonstration.

2. Replace repetitions of the MP with cyclic transitions. The algorithm used to perform this step is called *folding* and can be seen in Algorithm 2.1.

3. Combine the updated graphs into one global representation T of the Skill. The algorithm used to perform this step is called *merging* and is shown in Algorithm 2.2.

```
repetition = findRepetition(T);
while repetition.found :
  R=0;
  repetition.l = repetition.end-repetition.start+1;
  for i = repetition.start to repetition.end :
    mergeNodes(T(i+repetition.l),T(i));
    R.add(T(i));
  repetition = findRepetition(T);
  if !repetition.found :
    tail = findTail(T,repetition.end+1);
    if tail.found : //Found incomplete cycle
      for i = tail.start to tail.end
        mergeNodes(T(i+r),T(i));
        R.add(T(i));
  T.remove(R);
```

Algorithm 2.1: Graph Folding

```
repetition = findRepetition(T);
U_A = getUniquePaths(T_A);
U_B = getUniquePaths(T_B);
for all u_B of U_B : //Iterate over paths
  c_max = 0;
```

```
    for all u_A of U_A :
      c = compare(u_A, u_B); // Num matching nodes
      if c > c_max :
        c_max = c;
        n_{B,max} = u_B(1, ..., c); //First c nodes
        n_{A,max} = u_A(1, ..., c);
for all n_A of T_A, n_B of T_B : //Iterate over all nodes
  if n_B is in n_{B,max} : //Nodes match
    mergeNodes(n_A, n_B);
  else : // Nodes of graph B has to be added to A
    addNode(T_A, n_B)
```

Algorithm 2.2: Graph Merging

C. Learning the Transition Behaviour With the graph already built, the next step is to train the classifiers (one for each state in the graph). Each classifier has its domain restricted to the active node and its neighbours. Restricting the number of classes increases the accuracy of the system and reduces the complexity. Each transition in the acyclic graph is linked to one TP and then the *merging* multiple TP can be assigned to a transition. Hence, the TPs are assigned to the MP active at the moment. Any classifier can be easily applied to this method, for example, Support Vector Machine (SVM) [Cortes and Vapnik1995], Logistic Regression (LR) [Bishop2006], Kernel Logistic Regression (KLR) [Cortes and Vapnik1995], or GMM [Bishop2006].

2.3.3 Gaussian Mixture Model (GMM) with Gaussian Mixture Regression (GMR)

The GMM with GMR proposed by [Calinon2009] combines both techniques to obtain a regression algorithm designed to learn in the trajectory level. Below is a brief definition of the two methods.

A GMM is a probabilistic model that assumes that all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. Therefore, the GMMs in LbD are used to encode the temporal and/or spatial components of a continuous trajectory.

The GMR is based on the principle that all statistical information can be represented by a density function. Hence, it extracts the density function from a GMM. It also acts as a generalization process that computes conditional probability with respect to partially observed data. In LbD it is used to retrieve a smooth generalized trajectory with an associated covariance matrix describing its variance.

After introducing the GMM and GMR terms, how to encode and reproduce a skill (trajectory) with the GMM-GMR approach is explained below.

Let us assume that we have observed n demonstrations of a skill, consisting of n

observations of sensory data (Cartesian positions or joint angle) changing through time. We assume each demonstration can be rescaled to a fixed number of observations T. Hence, the total number of observations is defined by $N = nT$. A dataset ($\xi = \{\xi_j\}_j^N$) is formed by a set of data-point $\xi_j \in \mathbb{R}^D$ of D dimensions. These dimensions will be divided into two groups, ξ^I defining the state values and ξ^O defining the actions. The probability density function of K components is defined as follows:

$$p(\xi_j) = \sum_{k=1}^{K} p(k)(\xi_j|k) \tag{2.1}$$

where $p(k)$ is a prior and $p(\xi_j|k)$ is a conditional probability density function. For a Gaussian mixture, the parameters are defined by:

$$\begin{aligned} p(k) &= \pi_k, \\ p(\xi|k) &= \mathcal{N}(\xi_j; \mu_k, \Sigma_k) \\ &= \frac{1}{\sqrt{(2\pi)^D|\Sigma_k|}} \exp^{-\frac{1}{2}((\xi_j-\mu_k)^T\Sigma_k^{-1}(\xi_j-\mu_k))}. \end{aligned} \tag{2.2}$$

Thus, the GMM parameters are described by $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$, representing *prior* probabilities, *mean* vector and *covariance* matrices respectively. The π_k satisfies $\pi_k \in [0, 1]$ and $\Sigma_{k=1}^K \pi_k = 1$. Figure 2.8 shows a possible encoding of a dataset through GMM.

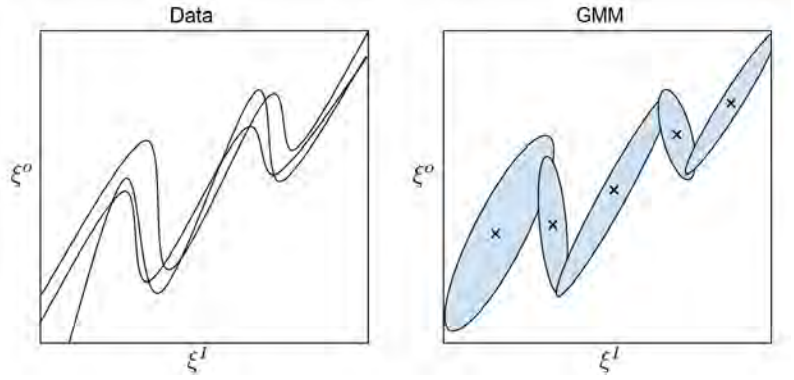


Figure 2.8: Illustration of a data set of D=2 encoded in a GMM of $K = 5$. The data set consists of $n = 4$ demonstrations of a trajectory.

Learning the GMM parameters is usually performed in batch mode by an Expectation-Maximization (EM) algorithm [Dempster et al.1977], which is a simple local search technique that guarantees a monotone increase of the likelihood during optimization.

Using equations (2.1) and (2.2), $p_{k,j}$ is defined as the posterior probability of $p(k|\xi_j)$, computed using the *Bayes* theorem $p(k|\xi_j) = \frac{p(k)p(\xi_j|k)}{\sum_{i=1}^K p(i)p(\xi_j ß)}$, and E_k as the sum of the posterior probabilities. Hence, the parameters of the GMM to be estimated are $\{\pi_k, \mu_k, \Sigma_k, E_k\}$.

The EM algorithm alternates between an *expectation* step, which computes the expectation of the log-likelihood using the current estimation of the parameters; and the *maximization* step, which estimates new parameters maximizing the expected log-likelihood found in the expectation step. These parameters are used in the next expectation step. The method converges when the log-likelihood between the previous parameters and the new parameters becomes negligible.

To accelerate the estimation of the GMM parameters, before using the EM method the parameters are approximated using a k-means segmentation method [MacQueen1967]. The two steps of the EM method are formulated as follows:

$$
\begin{aligned}
&\text{E-Step:}\\
&P_{k,j}^{t+1} = \frac{\pi_k^{(t)} \mathcal{N}(\xi_j; \mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_{i=1}^{K} \pi_i^{(t)} \mathcal{N}(\xi_j, \mu_i^{(t)}, \Sigma_i^{t})}\\
&E_k^{(t+1)} = \sum_{j=1}^{N} P_{K,j}^{(t+1)}\\
&\text{M-Step:}\\
&\pi_k^{(t+1)} = \frac{E_k^{(t+1)}}{N}\\
&\mu_k^{(t+1)} = \frac{\sum_{j=1}^{N} p_{k,j}^{(t+1)} \xi_j}{E_k^{(t+1)}}\\
&\Sigma_k^{(t+1)} = \frac{\sum_{j=1}^{N} P_{k,j}^{(t+1)} (\xi_j - \mu_k^{(t+1)})(\xi_j - mu_k^{(t+1)})^T}{E_k^{(t+1)}}
\end{aligned}
\tag{2.3}
$$

The iteration stops when the increase of the log-likelihood ($\mathscr{L}$) at each iteration becomes smaller than a threshold ($C_1 > \frac{\mathscr{L}^{(t+1)}}{\mathscr{L}^{(t)}}$). Figure 2.9 shows an example of the learning process for a GMM.

The likelihood in a GMM gives a numeric value to the similarity between the observed data and the model of the GMM. Therefore, the log-likelihood of a GMM model Θ is defined as follows:

$$
\mathscr{L}(\xi, \Theta) = \sum_{j=1}^{N} \log(p(\xi_j|\Theta)) \tag{2.4}
$$

where $p(\xi_j|\Theta)$ is the probability that ξ_j is generated by model Θ.

Next, the reproduction of a GMM model through a GMR is detailed. The goal of using a regression is to obtain an approximation of the conditional expectation between a set of state/predictor variables ξ^I and actions/responses ξ^O. The advantage of this approach is that it provides a fast and analytic solution to producing smooth and reliable trajectories,

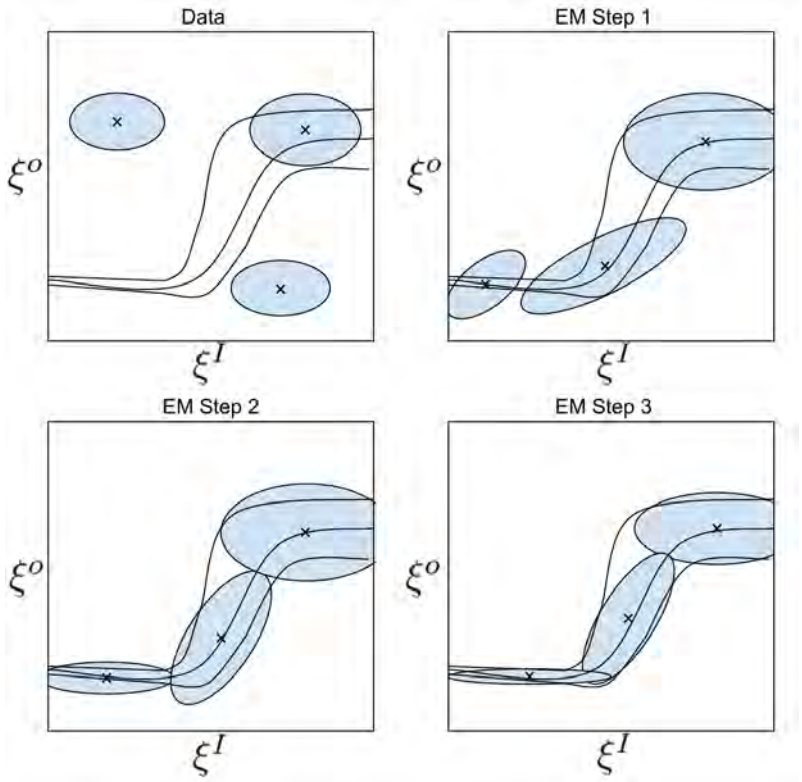


Figure 2.9: Illustrative example of how the EM algorithm adjusts the parameters of a GMM.

which can be used to control a robot efficiently. By computing the conditional expectation of ξ^O at each step of ξ^I, the whole expected trajectory is retrieved.

In this approach, the regression function is not approximated directly. Instead, the joint density of the set of trajectories is first estimated by a model from which the regression function is derived. It has the advantage of providing a density distribution of the responses instead of a single value. Hence, it generates a mean response estimation $\hat{\xi}$ and a covariance response estimation $\hat{\Sigma}$ at the same time.

The state and action values of the Gaussian component k are represented separately as

$$\mu_k = \begin{bmatrix} \mu_k^I \\ \mu_k^O \end{bmatrix}, \qquad \mu_k = \begin{bmatrix} \Sigma_k^{II} & \Sigma_K^{IO} \\ \Sigma_k^{OI} & \Sigma_K^{OO} \end{bmatrix}. \tag{2.5}$$

Considering the complete GMM, the expected distribution of ξ_k^O, when ξ^I is known, is defined by the conditional probability:

$$\begin{aligned} p(\xi_k^O|\xi_k^I) &= \mathcal{N}(\xi_k^O; \hat{\xi}_k^O, \hat{\Sigma}_k^{OO}), \\ \hat{\xi}_k^O &= \mu_k^O + \Sigma_k^{OI}(\Sigma_k^{II})^{-1}(\xi^I - \mu_k^I), \\ \hat{\Sigma}_K^{OO} &= \Sigma_k^{OO} - \Sigma_k^{OI}(\Sigma_k^{II})^{-1}\Sigma_k^{IO}. \end{aligned} \tag{2.6}$$

Then, the K Gaussian distributions $\mathcal{N}(\hat{\xi}_k^O, \hat{\Sigma}_k^{OO})$ mixed according to prior h_k:

$$p(\xi^O|\xi^I) = \sum_{k=1}^{K} h_k \mathcal{N}(\xi^O; \hat{\xi}_k^O, \hat{\Sigma}_k^{OO}), \tag{2.7}$$

where $h_k = p(k|\xi^I)$ is the probability that the Gaussian distribution k is responsible for ξ^I, i.e.,

$$h_k = \frac{p(k)p(\xi^I|k)}{\sum_{i=1}^{K} p(i)p(\xi^I|i)} = \frac{\pi_k \mathcal{N}(\xi^I; \mu_k^I, \Sigma_k^{II})}{\sum_{i=1}^{K} \mathcal{N}(\xi^I; \mu_i^I, \Sigma_i^{II})} \tag{2.8}$$

According to the linear combination of Gaussian properties, the conditional expectation of ξ^O, given that ξ^I can be approximated by a single Gaussian distribution $\mathcal{N}(\hat{\xi}^O, \hat{\Sigma}^{OO})$ with parameters:

$$\hat{\xi}^O = \sum_{k=1}^{K} h_k \hat{\xi}_k^O \qquad \hat{\Sigma}^{OO} = \sum_{k=1}^{K} h_k^2 \hat{\Sigma}_k^{OO}. \tag{2.9}$$

Hence, by evaluating $\{\hat{\xi}^O, \hat{\Sigma}^{OO}\}$ at different steps ξ^I, a generalized form of the motions $\hat{\xi} = \{\xi^I, \xi^O\}$ and associated covariance Σ^{OO} describing the constraints is obtained. Figure 2.10 illustrates the proposed GMR.

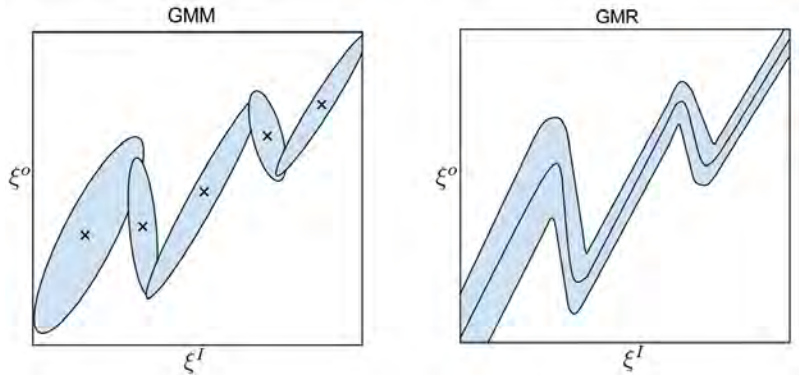


Figure 2.10: Illustrative example of a GMR process used to compute the conditional function density of $p(\xi^O|\xi^I)$. The figure on the left shows the GMM and, on the right, the GMR obtained.

Moreover, the performance of the GMR obtained will vary depending on the number of Gaussians used in the GMM. This variation represents the compromise between accuracy and a smooth response. This compromise is known as the *bias-variance trade-off*. The *bias-variance* has to be defined to fulfill two objectives: First, the bias between the estimated and real values has to be low; second, the variance on the estimate should also be low.

2.3.4 Dynamic Movement Primitive (DMP)

The DMP proposed by [Ijspeert et al.2013] is a regression algorithm used to learn a skill in a trajectory level. It proposes a generic modelling approach to generate multidimensional

systems of weak non-linear differential equations to capture an observed behaviour in an attractor landscape. The basic idea is to use an analytically well-understood dynamical system with convenient stability properties and modulate it with non-linear terms. DMP can model discrete movement (using point attractor models) and rhythmic movements (using limit cycle models). The goal is to propose a model that is an autonomous system, without explicit time dependency, and scale-invariant.

One of the simplest possible models to implement the DMP is the damped spring model:

$$\tau \ddot{z} = \alpha_z(\beta_z(g - y) - \dot{y}) + f, \tag{2.10}$$

which can be written in first-order notation,

$$\tau \dot{z} = \alpha_z(\beta_z(g - y) - z) + f \qquad \tau \dot{y} = z, \tag{2.11}$$

where τ is a time constant and α_z and β_z are positive constants. The f represents the forcing term which adds the non-linear values. Hence, if $f = 0$, these equations represent a stable second-order linear system with $(z, y) = (0, g)$ as the unique point attractor. In order for y to monotonically converge toward g, the α_z and β_z values have to be adjusted properly. The variables y,$\dot{y}$, and $\ddot{y}$ will be interpreted as a desired position, velocity and acceleration for a control system.

In order to achieve more versatile point attractor dynamics, the forcing term f in equation (2.11) can be defined as:

$$f(t) = \frac{\sum_{i=1}^{N} \Psi_i(t) w_i}{\sum_{i=1}^{N} \Psi_i(t)}, \tag{2.12}$$

where Ψ_i are fixed basis functions and w_i are adjustable weights. The function proposed has an explicit time dependence. Thus, a replacement of time is introduced by means of a first-order linear dynamics in x,

$$\tau \dot{x} = -\alpha_x x, \tag{2.13}$$

where α_x is a constant. Starting from some arbitrary state such as $x_0 = 1$, the state converges monotonically to zero. Equation (2.13) is called a *canonical system* because it models the generic behaviour.

With equation (2.13), we can reformulate our forcing term (eq. (2.12)) :

$$f(x) = \frac{\sum_{i=1}^{N} \Psi_i(x) w_i}{\sum_{i=1}^{N} \Psi_i(x)} x(g - y_0) \tag{2.14}$$

with N exponential basis functions $\Psi(x)$ Equation (2.15) and y_0 representing the initial state.

$$\Psi(x) = exp(-\frac{1}{2\sigma_i^2}(x - c_i)^2), \tag{2.15}$$

where σ_i and c_i are constants that determine the width and centres of the basis functions.

Equation (2.14) is modulated by two elements x and $(g - y_0)$. The modulation by x eliminates the forcing term when goal g is reached, providing the stability of the attractor equations. The modulation by $(g - y_0)$ incorporates the scaling properties into the model under a change of the movement amplitude.

The complete system is designed to have a unique equilibrium point at $(z, y, x) = (0, g, 0)$. Parameters w_i can be adjusted using learning algorithms to produce the desired complex trajectories before reaching g.

We assume that a desired behaviour is given by one or multiple desired trajectories in terms of position, velocity and acceleration $(y_{\text{demo}}(t), \dot{y}_{\text{demo}}(t), \ddot{y}_{\text{demo}}(t))$, where $t \in [1, ..., P]$[6]. The learning is performed in two phases, first determining the high-level parameters (g, y_0, and τ) and then learning the parameters w_i.

Parameter g is simply the position at the end of movement $g = y_{\text{demo}}(t = P)$ and, analogously $y_0 = y_{\text{demo}}(t = 0)$. Parameter τ must be adjusted according to the duration of the demonstration.

The Learning of the w_i is accomplished with Locally Weighted Regression (LWR) [Schaal and Atkeson1997]. For formulating a funciton approximation problem, we rearrange equation (2.11)

$$\tau\dot{z} - \alpha_z(\beta_z(g - y) - z) = f. \tag{2.16}$$

Inserting the information from the demonstrated trajecotry in the left-hand side of the equation, we obtain

$$f_{\text{target}} = \tau^2\ddot{y}_{\text{demo}} - \alpha_z(\beta_z(g - y_{\text{demo}}) - \tau\dot{y}_{\text{demo}}). \tag{2.17}$$

Hence, we have obtained a function approximation problem where the parameters of f are to be adjusted so that f is as close as possible to f_{target}. LWR finds for each kernel function Ψ_i in f the corresponding w_i, which minimizes the locally weighted quadratic error criterion,

$$J_i = \sum_{t=1}^{P} \Sigma_i(t)(f_{\text{target}}(t) - w_i\xi(t))^2, \tag{2.18}$$

where $\xi(t) = x(t)(g - y_0)$. This weighted linear regression problem has the solution

$$w_i = \frac{s^T T_i f_{\text{target}}}{s^T T_i s}, \tag{2.19}$$

where

$$s = \begin{pmatrix} \xi(1) \\ \xi(2) \\ ... \\ \xi(P) \end{pmatrix} T_i = \begin{pmatrix} \Psi_i(1) & & & 0 \\ & \Psi_i(2) & & \\ & & ... & \\ 0 & & & \Psi_i(P) \end{pmatrix} f_{\text{target}} = \begin{pmatrix} f_{\text{target}}(1) \\ f_{\text{target}}(2) \\ ... \\ f_{\text{target}}(P) \end{pmatrix}. \tag{2.20}$$

These equations are easy to implement. Multiple demonstratons of the trajectory are used, even at different scales and timing, and can be averaged together in the above equations. This is possible due to the invariance properties of the method. The regression performed with equation (2.19) corresponds to a batch regression. However, we can use a forgetting factor in the recursive least squares regression to determine the w_i in an incremental form.

2.3.5 Significant Methods for Robotics

In this section, the authors highlight some LbD algorithms focusing on the robotics area related mostly to manipulation and trajectory learning. All these methods are listed in Tables 2.2, 2.3, 2.4 and 2.5. They include the necessary information to classify each method and to point out the most relevant contribution and its application. The different tables are organized as follows: Table 2.2 presents methods which, in the authors opinion, represent the base for LbD, Table 2.3 presents relevant symbolic methods focusing on the structure to represent the skill, Table 2.4 presents a parametric version of the methods in Table 2.2, and Table 2.5 presents interesting modifications of the trajectory methods.

Reference Article	Method Name	Demonstration Method	Skill Learning/ Reproduction	Brief Description	Used application
[Hovland et al.1996a]	Hidden Markov Model (HMM)	Imitation/ Sensor on a teacher	Symbolic/ Classification	A task is represented by a hybrid dynamic system where a discrete event controller, represented by an HMM, modules the skill at the task level.	It has been used to insert a peg in a hole using a manipulator to demonstrate the acquisition of an assembly skill.
[Hersch et al.2006]	Gaussian Mixture Model (GMM) and Gaussian Mixture Regression (GMR)	Demonstration/ Kinesthetic guiding	Trajectory/ Regression	The algorithm extracts the constant features of the movement using a GMM and then uses the GMR and a dynamic system to reproduce the movement.	It has been tested using a humanoid robot. The task performed has consisted of a packing application, where an object is put inside a box.
[Hoffmann et al.2009]	Dynamic Movement Primitive (DMP)	Imitation/ Sensors on a teacher	Trajectory/ Regression	Improves the DMP framework avoiding singularities and large accelerations. An additional term is added to steer around obstacles.	It has been tested moving a cup between two points using a manipulator. During the reproduction, the arm is able to avoiding moving obstacles.
[Calinon et al.2010a]	Hidden Markov Model (HMM) and Gaussian Mixture Regression (GMR)	Demonstration/ Kinesthetic guiding	Trajectory/ Regression	Uses HMM to encode the task model and GMR to reproduce it.	It has been tested using two different manipulators. One to hit a ping pong ball and the other to feed a doll.

Table 2.2: First summary table of some LbD algorithms, which in the authors opinion represent the base of LbD methods on manipulation and trajectory learning.

Reference Article	Method Name	Demonstration Method	Skill Learning/ Reproduction	Brief Description	Used application
[Ekvall and Kragic2006]	Sequential Graph and Planning Algorithm	Not specified	Symbolic/ Classification	The two main contributions are the state segmentation and the constraint identification.	The proposed method has been tested in a virtual environment to define how to *set a table* and in a laboratory environment using a manipulator to stack a pyramid of blocks in a certain order according to the demonstrations.
[Abbas and MacDonald2011a]	Longest Common Subsequence (LCS)	Not specified	Symbolic/ Classification	It creates the LCS using multiple sequences of the demonstrations.	It has been tested using abstract path planning problems, where a robot has to go from a *start* position to the *end*.
[Manschitz et al.2014]	Gaussian Mixture Model (GMM) or Support Vector Machine (SVM)	Demonstration/ Kinesthetic guiding	Symbolic/ Classification	The task is represented as a sequential graph of primitive actions, the transitions are done using GMM and SVM.	The method has been proved with an industrial manipulator to unscrew a light bulb.
[Konidaris et al.2012]	Constructing Skill Trees (CST)	Demonstration/ Teleoperation	Symbolic/ Classification	The CST segements the demonstrations into a chain of skills, where each skill has a goal and an abstract action associated. The different chains are merged creating a skill tree to represent the task.	The method has been tested in three scenarios using a mobile robot with two manipulators. The robot has been taught to navigate in different scenarios and interact with the environment.

Table 2.3: Second summary table of the some LbD algorithms focusing on symbolic methods for robotics application related to manipulation and trajectory learning.

Reference Article	Method Name	Demonstration Method	Skill Learning/ Reproduction	Brief Description	Used application
[Wilson and Bobick1999]	Parametric Hidden Markov Model (p-HMM)	Imitation/ External observation	Symbolic/ Classification	It includes a global parametric variation in the output probabilities of the HMM states to represent spatial variation in the task.	The method has been used to recognize gestures performed by human, recognizing two similar gestures. *Pointing* at an object and indicating *movement direction.*
[Kruger et al.2010]	Parametric Hidden Markov Model (p-HMM)	Imitation/ External observation	Symbolic/ Classification	The learned task is modified by a parameter that effects all the states in the HMM model.	It has been used to move objects between positions in a controlled environment, defining the initial and final positions using the parameters.
[Calinon et al.2014]	Parametric Gaussian Mixture Model (p-GMM)	Demonstration/ Kinesthetic guiding	Trajectory/ Regression	Uses multiple frame references to learn different models of the same task and then merges them	It has been used to move a conical peg from one position to another.
[Matsubara et al.2011]	Parametric Dynamic Movement Primitive (p-DMP)	Imitation/ Sensors on the Teacher	Trajectory/ Regression	The demonstrations are provided with a style parameter, this parameter represents some differences in the environment.	It has been tested moving an object from one position to another, avoiding obstacles with different heights.

Table 2.4: Third summary table presenting a parametric version of LbD algorithms in the Table 2.2.

Reference Article	Method Name	Demonstration Method	Skill Learning/ Reproduction	Brief Description	Used application
[Kormushev et al.2011]	Dynamic Movement Primitive (DMP)	Demonstration/ Kinesthetic guiding	Trajectory/ Regression	DMP learning has been extended to include a force associated with the trajectory. The position and force primitive models share the canonical system.	It has been tested performing an ironing task and opening a door with a manipulator.
[Kronander and Billard2012]	Gaussian Mixture Model (GMM) with Gaussian Mixture Regression (GMR)	Demonstration/ Kinesthetic guiding	Trajectory/ Regression	The robot already knows the task and, during the reproduction, the teacher interacts with the robot to offer corrections of the stiffness of the trajectory.	A test of the method has been performed with two tasks; moving an object, and pouring a liquid from a bottle into a cup.
[Krüger et al.2012]	Infinite Gaussian Mixture Model (IGMM)	Demonstration/ Kinesthetic guiding	Trajectory/ Regression	The IGMM automatically finds the number of mixtures necessary to reflect the data complexity.	It has been used to perform several different movements with the arm of an iCub [Metta et al.2008] robot.
[Forte et al.2012]	Dynamic Movement Primitive (DMP)	Demonstration/ Kinesthetic guiding	Trajectory/ Regression	The method creates database of robot motions and adapts them to the need requirement using regression methods.	It has been used to perform several pick-up task using visual tracking and two different dual manipulator robots, the HOAP-3 and two KUKA LBR-4.

Table 2.5: Forth summary table of some LbD algorithms which present some interesting modifications of trajectory learning.

3

Framework for underwater skill learning

With the aim of performing autonomous underwater interventions, this chapter presents a complete framework for an Intervention Autonomous Underwater Vehicle (I-AUV) that uses Learning by Demonstration (LbD) as a bottom line. Section 3.2.1 presents the LbD technique in detail and justifies the use of this method. Section 3.3 introduces all the software elements in the I-AUV framework and all the modifications done to adapt the LbD basic algorithm in order to perform autonomous underwater interventions.

3.1 Introduction

Learning by Demonstration (LbD) is a family of machine learning algorithms which exploits the possibility of transferring knowledge from a human teacher to a machine in a natural way. Since the robot developers can not foresee all the specific needs of a robotic user, this method lets the owner teach new skills by giving demonstrations. Hence, the user of a robot using LbD is not tied to a robotic expert to teach new tasks to the robot, reducing maintenance and deployment costs while increasing the robot's applications.

A redefined propose of the well known Dynamic Movement Primitive (DMP) algorithm has been extended in this work to tailor this method to the particular needs of an AUV equipped with a manipulator (i.e., an I-AUV). Not only has the DMP algorithm been adapted to the I-AUV, but several modifications have been made in the I-AUV to adapt it to the LbD algorithm requirements.

When LbD is applied to industrial manipulators, the teaching is either performed through kinematic movements, where the teacher moves the robot to perform the demonstrations, or the teacher herself is recorded performing the demonstration and the robot learns by imitation. In an underwater environment this is not possible, so the best option is to enable the I-AUV to be operated as a ROV to record the demonstrations.

Besides the I-AUV, the environment must also be adapted since some tasks are performed under dangerous conditions. For this reason, it is interesting to create a mock-up environment in which the human operator will be able to perform the demonstrations with the same I-AUV in a controlled environment. The learned models can be tested and evaluated in this mock-up environment and be deployed in the real one later.

Therefore, one or more human operators (e.g., an operator for the manipulator and another for the vehicle) will perform the demonstrations in a controlled environment. These demonstrations will include different strategies to resolve the intervention task in order to create a flexible and generic model of the skill to learn.

This thesis has focused on the context of performing a concrete case of free-floating manipulation, where the I-AUV has to turn an underwater valve on a panel without docking. The experiments have evolved from a simple scenario to a complex scenario presenting new challenges to the methods developed in this thesis. It makes the evolution of the algorithm and its interaction with other systems in the I-AUV necessary to complete the task successfully.

The following sections present the proposed DMP algorithm to be used in underwater intervention tasks.

3.2 Learning by Demonstration (LbD) Approach

This section explains why the DMP algorithm performs better than other LbD techniques like HMM [Hovland et al.1996a] [Calinon et al.2010a], GMM [Hersch et al.2006] in the proposed scenario.

When developing an LbD algorithm, the first election to make is to decide which aspect of the skill is going to be learned. Because performing simple intervention actions (e.g., keep a position, grab an object, keep a contact position) with an I-AUV is not a trivial task, constructing an action library to learn a skill semantically would be too difficult and costly. For this reason, the authors have decided to use trajectory learning to transfer the skill to be performed.

The second decision is to choose the learning in the *trajectory level* to learn the skill. We decided to learn in the *task space* instead of the *joint space*, due to the a greatter flexibility in the configuration of the I-AUV as well as in the environment.

Several methods can be used to learn a skill in the trajectory space. As presented in the previous section, the most common approaches are GMM-GMR [Hersch et al.2006], HMM-GMR [Calinon et al.2010a], and DMP [Ijspeert et al.2013]. From all possible algorithms we decided to use as the base a method that encoded the DMP using an HMM and to generate the movement using a combination of a modified GMR [Calinon et al.2010b], which combines the different advantages of booth options, presenting the following desirable features:

- Any smooth movement can be encoded and generated using a DMP algorithm.
- The system converges to a final goal point and automatically adapts to perturbations.
- Generated movements are adapted on-line to changes in the goal position.
- The model is translation invariant and time independent.
- The speed of reproduction can be modified without affecting the shape of the trajectory.

Section 3.2.1 presents the version of DMP algorithm used while Section 3.2.2 compares the classic DMP with the approach used.

3.2.1 Dynamic Movement Primitive (DMP) using a modified HMM and GMR

The version of the DMP used in this thesis was introduced by [Calinon et al.2010b] and modified by [Kormushev et al.2011]. The learned skill is encapsulated in a superposition of basis motion fields (see Fig. 3.1). The flexibility of the proposed representation allows us to adapt the algorithm to specific requirements, as explained in Section 3.4.

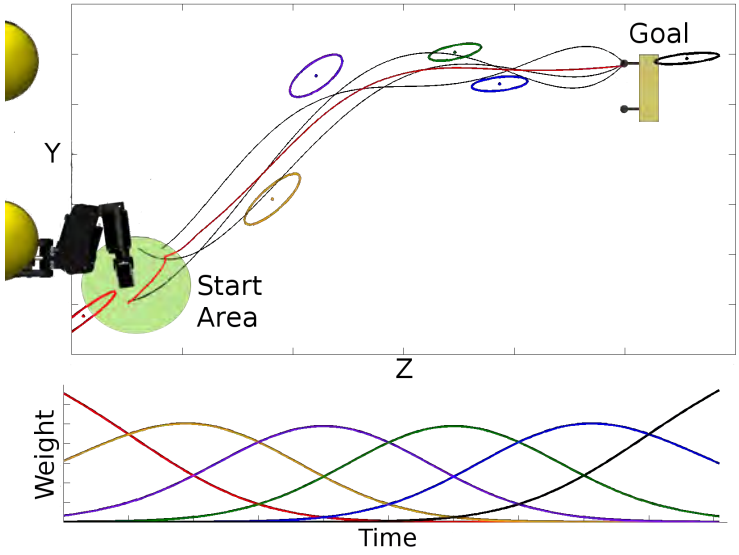


Figure 3.1: Top figure shows a set of 2D demonstrated trajectories (black) and one reproduction (red). In this case, the demonstrated trajectory has to grasp the valve aligning the fore arm of the manipulator with the valve. Below, the h function is represented. The encoding of the trajectories using a DMP algorithm has been done using 6 Gaussians adequately weighted over time.

To better understand this encoding, we can imagine a mass attached to different damped strings. These strings attract the mass, changing their forces during the time of the experiment, moving the mass along the desired trajectory.

To generate the superposition, each attractor has an associated weight that changes during the time defined by the $h_i(t)$ (see (3.1)). The weight of each attractor is represented with a Gaussian, whose centres μ_i^T are equally distributed in time, and whose variance parameters ($\Sigma_i^T = \text{total\_time}/K$) are set to a constant value inversely proportional to the number of Gaussians (K).

$$h_i(t) = \frac{\mathcal{N}(t; \mu_i^T, \Sigma_i^T)}{\sum_{k=1}^{K} \mathcal{N}(t; \mu_k^T, \Sigma_k^T)}. \tag{3.1}$$

Instead of using real time, a decay term is used to obtain a time-invariant model:

$$t = \frac{ln(s)}{\alpha}, \tag{3.2}$$
$$\text{where } s \text{ is a canonical system : } \dot{s} \quad = s - (\alpha p_{cf})s,$$

the α value is selected by the user depending on the duration of the demonstrated task, and p_{cf} is the period of the control frequency of the system.

The number of attractors is preselected by the user and represented using Gaussians, depending on the complexity of the task. The position of the attractor is the center of the Gaussian (μ_i^x) and the stiffness (matrix K_i^P) is represented by the covariance of the

Gaussian. The values are learned from the observed data through least-squares regressions. All the data from the demonstrations is concatenated in a matrix $Y = [\ddot{x}\frac{1}{K^P} + \dot{x}\frac{K^V}{K^P} + x]$, where x, $\dot{x}$ and $\ddot{x}$ are the position, velocity and acceleration recorded at each time instant of the demonstrations. Also, the weights at each time instant are concatenated to obtain matrix H. With these two matrices, the linear equation $Y = H\mu^x$ can be written. The least-square solution to estimate the attractor center is then given by $\mu^x = H^{\dagger}Y$, where $H^{\dagger} = (H^T H)^{-1}H^T$ is the pseudo-inverse of H.

The user needs to set a minimum K^P_{min} and a maximum K^P_{max} to define the limits of the stiffness and to estimate the damping as follows:

$$K^P = K^P_{min} + \frac{K^P_{max} - K^P_{min}}{2}, \tag{3.3}$$

$$K^V = 2\sqrt{K^P}. \tag{3.4}$$

To take into account variability and correlation along the movement and among the different demonstrations, the residual errors of the least-squares estimations are computed in the form of covariance matrices for each Gaussian ($i \in \{1, ...K\}$).

$$\Sigma^X_i = \frac{1}{N}\sum_{j=1}^{N}(Y'_{j,i} - \bar{Y_i}')(Y'_{j,i} - \bar{Y_i}')^T, \quad \forall_i \in \{1, ...K\}, \tag{3.5}$$

where:

$$Y'_{j,i} = H_{j,i}(Y_j - \mu^x_i). \tag{3.6}$$

In 3.5, the $\bar{Y_i}'$ is the mean of Y'_i over the N data points.

Finally, the residual terms of the regression process are used to estimate the K^P_i through the eigen components decomposition.

$$K^P_i = V_i D_i V_i^{-1}, \tag{3.7}$$

where:

$$D_i = k^P_{min} + (k^P_{max} - k^P_{min})\frac{\lambda_i - \lambda_{min}}{\lambda_{max} - \lambda_{min}}. \tag{3.8}$$

In (3.7), the λ_i and the V_i are the concatenated eigenvalues and eigenvector of the inverse covariance matrix $(\Sigma^x_i)^{-1}$. The basic idea is to determine a stiffness matrix proportional to the inverse of the observed covariance.

Therefore, the model for the task will be composed of: the K^P_i matrices and μ^x_i centres representing the Gaussians; $h_i(t)$ representing the influence of each matrix function; K^V representing the damping; and α, which is assigned according to the duration of the demonstration. Figure 3.1 shows a simple example where the data learned is represented.

Finally, to reproduce the skill learned, the desired acceleration is generated with

$$\hat{\ddot{x}} = \sum_{i=1}^{K} h_i(t)[K_i^P(\mu_i^X - x) - K^V \dot{x}], \tag{3.9}$$

where x and $\dot{x}$ are the current position and velocity.

One of the benefits of using the variability and correlations of the demonstrated task is the possibility of modifying the reproduced trajectory by external forces or other requirements of the task and still follow the proper trajectory.

3.2.2 Differences between both implementations for DMP

The DMP proposes to capture the skill/trajectory in the attractor landscape, defined by a non-linear function and a point attractor primitive. The proposed approach follows the same idea but instead of a non-linear system it uses an HMM to represent the attractor points associating a Gaussian function to each state, and a GMR to reproduce the learned model.

In the following lines the main of the DMP equations proposed by the [Ijspeert et al.2013] and [Calinon et al.2010b] methods are compared to show their similarities. Hence, lets rewrite the (2.11) in the following terms:

$$\begin{aligned} \ddot{x} &= \alpha_z(\beta_z(g - x) - \dot{x}) + f(s), \\ \ddot{x} &= f(s) + \alpha_z\beta_z(g - x) - \alpha_z\dot{x}, \\ \ddot{x} &= f(s) + K^P(g - x) - K^V\dot{x}. \end{aligned} \tag{3.10}$$

On the other hand, the equivalent equation of the proposed DMP can be re-written as follows:

$$\ddot{x} = s\left[K^P\left((\sum_{i=1}^{K} h_i\mu_i^x + x_0 - x\right)\right] + (1 - s)\left[K^P(g - x)\right] - K^V\dot{x}, \tag{3.11}$$

where x_0 defines the initial starting position and g defines the position to attain. The s is the decay term defined by the canonical system. Hence, $s\left[K^P\left((\sum_{i=1}^{K} h_i\mu_i^x + x_0 - x\right)\right]$ is the equivalent of $f(s)$; $(1 - s)\left[K^P(g - x)\right]$ is the equivalent of $\alpha_z\beta_z(g - x)$ and $K^V\dot{x}$ of $\alpha_z\dot{x}$.

Equation (3.11) allows, to the DMP, to be translational invariant and g guarantees the convergence of the system. But if the system is described in the frame of a given target, x_0 can be set to zero without losing generality. Furthermore, if the demonstrations reduce the velocity to zero at the end of the skill, the system will automatically converge to the last attractor μ_K^x, and the explicit attractor g can be removed. Under these assumptions the DMP is formalized as

$$\hat{\dot{x}} = \sum_{i=1}^{K} h_i(s)[K^P(\mu_i^X - x) - K^V \dot{x}]. \tag{3.12}$$

The K^P value has been extended to a K_i^P as mentioned in Section 3.2.1 including the co-relation and variance of each Gaussian function.

There are several advantages to using a GMR and an HMM instead of the classic approach. First, it allows the possibility of dealing with recognition and reproduction of skills in the common probabilistic approach. Second, the learning process is distinct from the retrieval process, thus it allows training and refinement of the skill providing partial demonstrations. Third, it allows spatial and temporal distortion to be handled flexibly. Fourth, this approach learns correlations across different variables and uses this information in the reproduction process. Fifth, the type of motion is generic and doesn't need to be specified as this the method uses the same system to handle discrete and periodic movements.

One drawback of this approach is that learning the weights $h_i(t)$ from examples has an impact on the system's stability in regions that have not been covered during the demonstrations.

3.3 Framework

To perform an underwater intervention task using an I-AUV by means of a LbD algorithm it is necessary to build a set of complementary functionalities. These functionalities are organized in a framework with the aim of facilitating the integration of the LbD in the control architecture of the I-AUV. Moreover, the modular design of the framework allows the replacement of different elements to adapt the algorithm to different tasks. For example, it is possible to change the kind of control for the I-AUV or change the perception sensors or elements to identify different elements. Figure 3.2 shows a diagram with the basic elements that compose the proposed framework.

The framework is composed of several modules organized in four layers. Starting from the bottom, the first layer contains all the sensors and actuators drivers. The next layer processes sensor data. It contains the localization filter that merges the information from the navigation sensors (e.g. the Doppler Velocity Log (DVL), the Inertial Measurement Unit (IMU), the Pressure, and the Compass) and perception modules that process the perception sensors (i.e., cameras, range sensors, and Force/Torque (F/T) sensor) which are used to gather information from the environment. On top of this, the control layer includes the AUV and manipulator controllers (i.e., position, velocity, and force). These controllers are in charge of following the desired commands sent by the superior layer. Finally, in the top level layer, the LbD system is in charge of acquiring data from demonstrations (phase 1), learning the model (phase 2) and reproducing the task by generating commands for

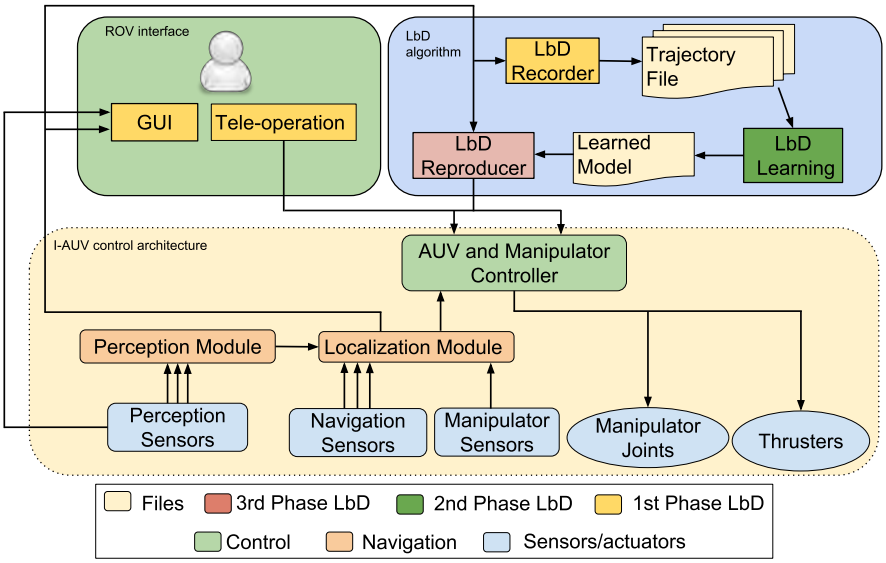


Figure 3.2: Diagram of the generic framework designed to apply LbD to an underwater intervention by means of an I-AUV

the vehicle and the manipulator (phase 3). The ROV interface module is also at this level. This module is in charge of the HRI, so an operator receives live information regarding the I-AUV state and can tele-operate it with a joystick. Therefore, the proposed intervention framework integrates perception, localization, control, interactions with the operator, and learning.

The perception module is essential for the use of the proposed LbD framework. This module is capable of identifying and localizing, the element to be manipulated, estimating its position and orientation in the Cartesian space (e.g. using a vision perception system as presented in Section 4.3.4). Hence, the trajectory performed by the AUV or the end-effector during the intervention task can be recorded using as a reference an identified element. Learning the trajectories with respect to the intervention element allows us to reuse the same model with the same element in different positions.

3.3.1 Demonstration

The demonstrations developed in the work of this thesis have been performed in the same environment as the reproductions, however, the authors are aware that this is not always possible and in some cases is not advisable. The proposed method learns the skill at the trajectory level to make it independent of the environment and other physical elements.

Throughout the development of this thesis the same I-AUV has always been used, hence, we have not been able to test the physical independence of the method. The closest test consisted of transferring a model learned in a simulated virtual environment with the real I-AUV. The transfer models have been able to perform the task but without the same

accuracy.

These problems can be attributed to the differences between the motion control of both the virtual and real systems. Beside these differences, the model learned in simulation can be used with a real I-AUV modifying only the K^P_{min} and K^P_{max} values. This suggests that applying a model learned in a mock-up environment or with a different I-AUV could be viable by just adapting the K^P_{min} and K^P_{max} to the new scenario/vehicle characteristics.

The set of demonstrations should have a minimum of 2 demonstrations and a maximum of 10 according to the authors experience (more details in Section 2.2.2). These demonstrations should explore the whole task space and cover the variabilities of the skill demonstrated. A good recommendation is to cover the boundaries of the demonstrated trajectory as well as the ideal trajectory, so that any unexpected situations can be better handled. Moreover, the quality of the reproductions depends from the quality of the demonstration. In this sense, it is better for the pilots to perform several demonstrations and discard the less accurate ones than include a huge quantity of demonstrations. On the other hand, if the intervention task can be solved using very different strategies, learning each strategy independently is recommendable since their generalization could create a model which might not represent any one of them.

3.3.2 Learning

The multivariant approach of the DMP algorithm makes the control of all the desired DOFs in the same model possible. This approach is useful to control the Cartesian trajectory of an end-effector with a fixed base. However, when the movement in each DOF has significant differences in terms of distance, velocity, force or precision this approach is not recommendable. The correlation between all the DOFs causes disturbances between them during the reproduction phase and increases the difficulty of establishing a proper minimum and maximum for the K^P values and the correct number of states to be used (K). The authors propose organizing the DOFs in different models according to the movement's characteristics. With this approach the parameters can also be tailored to these movement characteristics. All the models share the same canonical system to guarantee the same evolution over time.

On the other hand, the LbD algorithm needs to know which control scheme will be implemented in the I-AUV. Depending on the I-AUV control scheme, the LbD algorithm has to learn the trajectory of the end-effector or the trajectory of the end-effector and the vehicle base. If there are some DOFs that can not be actuated, the DMP algorithm will avoid learning them. The I-AUV control strategies are presented in Section 4.3.6, Section 4.4.7 and Section 4.6.

3.3.3 Reproduction

To reproduce the learned skill, the DMP algorithm generates the necessary accelerations or forces to follow the demonstrated trajectory with respect to the reference frame. These accelerations have to be converted into proper commands in order to be accepted by the I-AUV controller. The authors propose generating the same commands used by the teacher to demonstrate the skill. Therefore, acceleration commands are integrated to velocity commands, which are the commands used in tele-operation. To do this, accelerations in the learning frame must be transformed into velocities in the I-AUV's reference frame. Moreover, during the reproduction phase, the method can consider redundancies between DOFs and modify the generated commands accordingly.

3.4 DMP for underwater intervention

The following sections present the modification made in the DMP algorithm to solve the different challenges proposed in this thesis. The new features added to adapt the DMP to an underwater intervention domain are also detailed. Is worth mentioning that similar efforts have been made for industrial manipulators, like [Nemec et al.2013] where the algorithm increase the speed of the trajectory learned using a DMP combined with Iterative Learning Control (ILC) or RL, or [Ude et al.2014] where the DMP has been extended to learn orientation in Cartesian space.

3.4.1 Modification of the Canonical System

The canonical system has been modified to allow adjustment of the reproduction speed or even to reproduce the skill in the opposite direction. The goal of this modification is to let an evaluation system (e.g. a Fuzzy system proposed by [Reza Ahmadzadeh et al.2013]) to analyse how safe it is to continue executing the task and, according to this judgement, allow the robot to move forward, stop or move backwards along the learned trajectory.

Hence, the canonical system has been modified as follows:

$$\dot{s} = s - (\alpha p_{cf}) s e, \tag{3.13}$$

where e is an external continuous value between -1 and 1. The -1 means that the environment conditions are extremely dangerous and the action cannot be performed. The 0 value means that conditions are not safe, but the robot can keep the current position while waiting to see how the situation evolve. The 1 means that the environment situation is safe and the algorithm can advance at the same speed as in the demonstrations. This change requires another modification in the reproduction equation, to reduce the motion speed at the same time it reduces the advance in time. Equation (3.9) is modified as follows:

$$\hat{\ddot{x}} = |e| \sum_{i=1}^{K} h_i(t)[K_i^P(\mu_i^X - x) - K^V \dot{x}], \tag{3.14}$$

where $|e|$ is the absolute value of the environment evaluation. By this means, the force developed by each attractor is reduced proportionally to the safety of the environment.

3.4.2 DMP with angular normalization

The DMP can work with any kind of continuous variable and generate appropriate movement commands. However, an underwater intervention trajectory not only includes the position along the trajectory but also the orientation respect to the target. The authors propose learning the orientation of the AUV and end-effector using the Euler representation and wrap the value between π and $-\pi$. Equation (3.9) has been modified as follows:

$$\hat{\ddot{x}} = |e| \sum_{i=1}^{K} h_i(t)[K_i^P(wrap(\mu_i^X - x)) - K^V \dot{x}], \tag{3.15}$$

where the *wrap* function performs the normalization of the resulting angle between π and $-\pi$.

3.4.3 p-DMP

To model a task with a high degree of variability using an LbD algorithm, two strategies can be followed: either perform a high number of demonstrations in order to capture the variability of the task or learn one model for each situation. Both solutions present problems. The former leads to a too generic model not representative of the task while the latter forces the user to create a new model every time conditions change. The parametric LbD algorithm overcomes both problems by identifying the key parameters that differentiate each situation, consequently adapting the model.

In contrast to the previous approaches [Matsubara et al.2011] [Calinon et al.2014] [Kruger et al.2010] , we propose a parametric version of the DMP which learns different models associated to an environmental condition. In the reproduction phase, a weighted combination of these models is generated according to the current environmental situation. The motivation of this method is to represent different strategies to perform the same task depending on external factors which are not represented in the learned model.

The p-DMP associates a parametric value with each recorded demonstration. These parameters can be defined by several values not related to the number of DOFs learned. The parameter value ought to be defined by some environmental condition. The only requirement for selecting a proper parameter is that it should be possible to relate its values using a distance function.

All the demonstrations are grouped by the associated values thus conforming different

groups. For each group, a representative model will be learned using the aforementioned DMP algorithm (Section 3.2.1), with appropriate K^P_{max}, K^P_{min} and number of attractors (K).

In the reproduction phase, the influence (m) of each model is computed according to the distance between the current value of the parameter and the different models. The influence value obtained with (3.16) is normalized between 0 and 1.

$$\forall j \in J, \quad m_j = 1 - \frac{dist(p, q_j)}{\sum_{i=1}^{J} dist(p, q_i)}, \tag{3.16}$$

where $dist$ is the function that defines the distance between the current set of parameters (p) and group parameters (q) and J is the list of learned groups.

Moreover, the influence of each model is applied to coordinate the advance of the time (n) for each group, facilitating the combination of models with different durations using the time invariant model(t):

$$\begin{gathered} \forall j \in J, \quad t_j = \frac{ln(s_j)}{\alpha_j}, \\ \text{where } s \text{ is: } \dot{s_j} = s_j - \alpha_j s_j n_j, \end{gathered} \tag{3.17}$$

n_j is the proportional advance computed as:

$$\forall j \in J, \quad n_j = \frac{u_j}{\sum_{i=1}^{J}(u_i m_i)} \tag{3.18}$$

and u is the time needed to perform a regular reproduction of the learned model. Finally, the mixture of all these influences is computed as follows:

$$\hat{\ddot{x}} = \sum_{j=1}^{J} m_j n_j \left(\sum_{i=1}^{K} h_{j,i}(t_j)[K^P_{j,i}(\mu^X_{j,i} - x) - K^v_j \dot{x}] \right), \tag{3.19}$$

to obtain the desired acceleration.

3.4.4 Auto-tuning of the Learning parameters

Adjusting parameters and tuning the values to find the best approximation is always a tedious job which needs experimentation and time. In the case of LbD, this phase is usually quick due to the reduced number of parameters involved. Moreover, generic values obtain good results but not optimal solutions.

To simplify the use of LbD algorithms and obtain better results, the authors have proposed an algorithm to find and adjust the different values of the DMP algorithm proposed in this work. This approach uses a *brute-force* algorithm to search over a range of parameters, and compares the values in a simulated environment with ideal conditions

to speed-up the process.

The proposed algorithm tunes the three values (the number of states(k) and the minimum and maximum of K^P) defined by the user in the DMP algorithm. The first search iterates over all the values for k. For each value of k, the algorithm searches over all the possible combinations defined by the step and a superior and inferior limit for K^P. For each combination, the algorithm generates a model using the demonstrated trajectories. Algorithm 3.1 shows a pseudocode to illustrated the proposed method.

```
1  list_of_K = arange(min_k, max_k, setp_k)
2  list_of_kp = arange(min_kp, max_kp, step_kp)
3  ideal_traj = compute_avg_traj(demos_traj)
4  for states in list_of_K:
5    for i_k_min in range(len(list_of_kp)-1):
6      for j_k_max in range(len(list_of_kp[i_k_min:-1])):
7        kp_min = list_kp[i_k_min]
8        kp_max = list_kp[i_k_min + i_k_max + 1]
9        traj = learn_and_simulate(states, kp_min, kp_max,
             demos_traj)
10       is_better_dof = compare_traj_dof(ideal_traj, traj,
              dofs_groups)
11       for i in range(dofs_group):
12         if is_better_dof[i]:
13           best_k[i] = states
14           best_min_kp[i] = kp_min[i]
15           best_max_kp[i] = kp_max[i]
16 return [ best_k, best_min_kp, best_max_kp ]
```

Algorithm 3.1: Simplified version of the code for the auto-tuning algorithm, skipping most of the low level details of the implementation.

To define which set of values provides the best result, the average trajectory (*compute_avg_traj*()) is computed using the batch of demonstrations. The best trajectory is the one with the greatest likeness to this trajectory. The algorithm groups the trajectory's learned DOFs according to the different models learned. These groups are chosen according to the application and the user's criterion (more details in Section 3.3.2). The values obtaining the best average likeness in all the DOFs in the group are considered the best tuning (*compare_traj_dof*()). Dividing the search into different groups allows us to find different and more appropriate values for each group of DOFs.

4

Autonomous valve turning

This thesis has focused on resolving a free-floating autonomous underwater valve turning intervention. The GIRONA 500 AUV equipped with a commercial underwater manipulator of 4 DOFs and a custom end-effector has been chosen to perform this task. All the experiments have been performed in the water tank at the CIRS installations. A mock-up valve panel with T-shape valves, designed for the PANDORA project, has been used. Section 4.1 introduces the state of the art in autonomous underwater interventions. Later, the mock-up scenario is presented in Section 4.2 and the essential elements of the valve turning case are explained. The GIRONA 500 AUV is presented in Section 4.3, the manipulator in Section 4.4, and the end-effector is introduced in Section 4.5. Finally, Section 4.6 presents two different control schemes to control the GIRONA 500 AUV and the manipulator coordinately.

4.1 State of the art in underwater intervention

There are a large number of underwater tasks that go beyond survey capabilities; maintenance of underwater installations (e.g., permanent observatories, submerged oil wells, cabled sensor networks, and pipes), the deployment and recovery of benthic stations, or the search and recovery of black-boxes to name a few. In the early 90s, the scientific community started facing these tasks with pioneering AUVs like OTTER [Wang et al.1995] and ODIN [Choi et al.1994], as well as projects like AMADEUS [Lane et al.1997] (see Section 4.1.1), and UNION [Rigaud et al.1998] (see Section 4.1.2), but it was not until the 1st decade of the 21st century that field demonstrations arrived. Very successful approaches where based on hybrid ROV/AUV concepts like the one proposed by the SWIMMER project [Evans et al.2001] (see Section 4.1.3).

The first fully autonomous intervention at sea was demonstrated by the ALIVE project [Evans et al.2003] (see Section 4.1.4), where an AUV performed a valve turning task. The first object manipulation from a floating vehicle was achieved in 2009 within SAUVIM project [Marani et al.2009] (see Section 4.1.4.1). The capability of searching for an object whose position was roughly known a priori was demonstrated. The object was endowed with artificial landmarks and the robot autonomously located it and hooked it with a recovery device while hovering. The first multi-purpose object search and recovery strategy was demonstrated in the TRIDENT project in 2012 (see Section 4.1.6). An object was searched for and autonomously recovered in a water tank [Prats et al.2011a]. The experiment was also performed in a harbour environment using a 4 DOFs manipulator [Prats et al.2012a], and later on with a 7 DOFs manipulator with a 3 fingered end-effector [Sanz et al.2012] (see Section 4.1.6).

In the next subsections, the previously mentioned projects and some contemporary (see Section 4.1.8 and Section 4.1.9) and a posteriori (see Section 4.1.10, Section 4.1.11, Section 4.1.12, Section 4.1.13) project are reviewed in chronological order, with remarks on their main contribution as well as the main results achieved. Moreover, two innovative ROVs for intervention, the OceanOne (Section 4.1.14) and the CR200 (Section 4.1.15), are also included.

4.1.1 AMADEUS 1993-99

The project focused on the development of underwater intervention capabilities divided into two phases. Phase I represents the first attempt to develop a dexterous gripper suitable for underwater applications [Lane et al.1997] (see Figure 4.1). A three fingered gripper was hydraulically actuated and coordinately controlled by mimicking the motions of an artificial elephant trunk. Phase II was devoted to the coordinated control of two underwater 7 DOF electro-mechanical manipulators [Casalino et al.2001]. Each arm weighted 65 kg, measured 140 cm, and was filled with oil enabling it to reach a depth of 500 m. To the best of

the author's knowledge, AMADEUS represents the first demonstration of an underwater electrical 7 DOFs arm with similar capabilities to an industrial manipulator. The project demonstrated the coordinated motion of the two end-effectors while manipulating a rigid object inside a water tank.

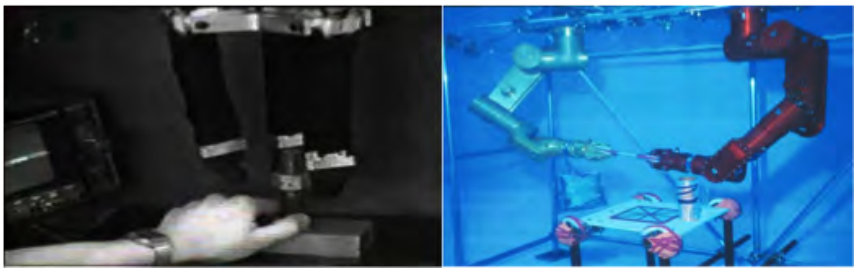


Figure 4.1: AMADEUS concept

4.1.2 UNION 1996-99

This was a pioneering project with the aim of developing methods to increase the autonomy and intelligence of ROVs [Rigaud et al.1998] (see Figure 4.2). The project focused mainly on the development of coordinated control and sensing strategies for the manipulator (PA10) and vehicle (VORTEX) systems. The joint dynamics of the vehicle-manipulator system were studied and a robust non-linear controller was proposed. To the best of the author's knowledge, UNION represents the first mechatronic assembly of a complete vehicle-manipulator system for automated manipulation. Nevertheless, the authors have failed to find published experimental manipulation results with the complete vehicle-manipulator.

Figure 4.2: UNION concept

4.1.3 SWIMMER 1999-01

This is a hybrid AUV/ROV intervention system conceived for permanent inspection and maintenance tasks on deep water oil production facilities [Evans et al.2001] (see Figure 4.3). The SWIMMER system is composed of a shuttle AUV which transports an intervention

ROV. SWIMMER is able to transit the sea-floor autonomously and to dock at a sub-sea cradle-based docking station. The cradle is cabled to the surface tele-operation post. Once docked, the transported ROV is deployed and connected to the shuttle by means of an excursion umbilical. The intervention is carried out in a conventional tele-operated way. SWIMMER's contribution consists of getting rid of the TMS and the expensive intervention vessel.

Figure 4.3: SWIMMER concept

4.1.4 ALIVE 2001-04

ALIVE is a milestone project in autonomous underwater intervention [Evans et al.2003] (see Figure 4.4). The ALIVE 3.5 Ton vehicle was equipped with two hydraulic grasps for docking and a 7 DOFs manipulation arm. It has been reported as the first AUV able to autonomously carry out a manipulation action consisting of opening/closing a valve in a sub-sea panel. The ALIVE intervention concept is based on docking at a sub-sea panel to perform a fixed-base manipulation. There is no interaction between the arm and the vehicle, hence the manipulation becomes a conventional single arm manipulation but underwater. During the final demo of the ALIVE project, the system proved its ability to navigate, dock and open a valve on an underwater panel similar to those of the oil industry autonomously.

Figure 4.4: ALIVE concept

4.1.4.1 SAUVIM 1997-09

SAUVIM is also a milestone project [Marani et al.2009] (see Figure 4.5). It was funded by the Office of Naval Research and was carried out at the Autonomous System Laboratory of the University of Hawaii. It was coexistent with ALIVE, but proposed a very different approach to autonomous intervention. The weight of the AUV (approx. 6 Ton) was much greater than that of the arm (just 65 Kg), therefore, both systems were considered as practically uncoupled. In other words, the perturbations produced in the AUV's position as result of the motion of the arm were negligible. Because of that, uncoupled controllers were implemented. SAUVIM demonstrated accurate navigation and station keeping, as well as being the first project to demonstrate the autonomous recovery of an a priori known object. It is worth mentioning that SAUVIM used the ANSALDO arm previously built in the context of the AMADEUS project.

Figure 4.5: SAUVIM concept

4.1.5 RAUVI 2009-11

RAUVI was a Spanish funded project devoted to the design and implementation of a reconfigurable AUV for intervention missions (see Figure 4.6). The major outcome of RAUVI was the development of the GIRONA 500 AUV [Ribas et al.2012]) which was equipped with an ECA/CSIP electrically driven arm with 4 DOFs [Fernandez et al.2013]. The AUV was equipped with a stereo pair of cameras and was used to demonstrate autonomous object recovery in a water tank environment [Prats et al.2011b]. To the best of the authors' knowledge, after ALIVE and SAUVIM, RAUVI is the 3rd project which demonstrated experimentally autonomous intervention capabilities, and the one using the lightest vehicle (less than 200 kg). RAUVI was the seed of a more ambitious EU project named TRIDENT.

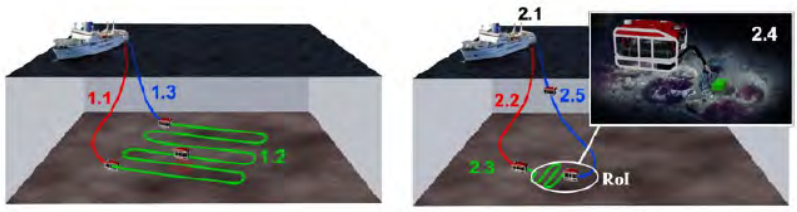


Figure 4.6: RAUVI concept

4.1.6 TRIDENT 2010-12

This project proposed a new methodology for multi-purpose underwater intervention tasks [Sanz et al.2012] (see Figure 4.7). A team of two cooperative heterogeneous robots with complementary skills, an Autonomous Surface Craft (ASC) and an I-AUV equipped with a dexterous 7 DOFs manipulator and a 3-fingered hand, were used to perform underwater manipulation tasks. The TRIDENT concept is divided into two phases: Survey and Intervention. During the survey, the team of vehicles map the sea-floor. Next, the I-AUV is recovered and a sea-floor map is built. With the help of the map, the user selects an object and a desired intervention task. Then, the team of vehicles is deployed again to search for the target using the existing map as a reference. Once this is accomplished, the I-AUV performs a multisensory-based intervention through free-floating manipulation to recover the object. The TRIDENT concept has been demonstrated in a harbour environment in an uncoupled way: 1) The capability of both vehicles working in tandem during mapping and 2) the capability of the I-AUV to intervene with the target. As an evolution of RAUVI, TRIDENT, together with ALIVE and SAUVIM, became a milestone project in autonomous underwater manipulation providing, for the first time, field results in multipurpose object recovery.

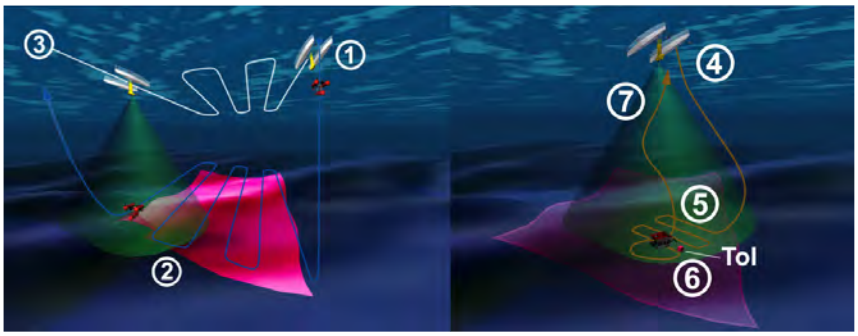


Figure 4.7: TRIDENT concept

4.1.7 TRITON 2012-14

TRITON is a Spanish funded project which demonstrated autonomous docking at a customized sub-sea panel, fixed-based manipulation for valve turning and hot stab connections. These are common applications in the maintenance of permanent submerged observatories (see Figure 4.8). Project demonstrations were carried out in a water tank [Palomeras et al.2013] as well as in a harbour environment [Palomeras et al.2014a].

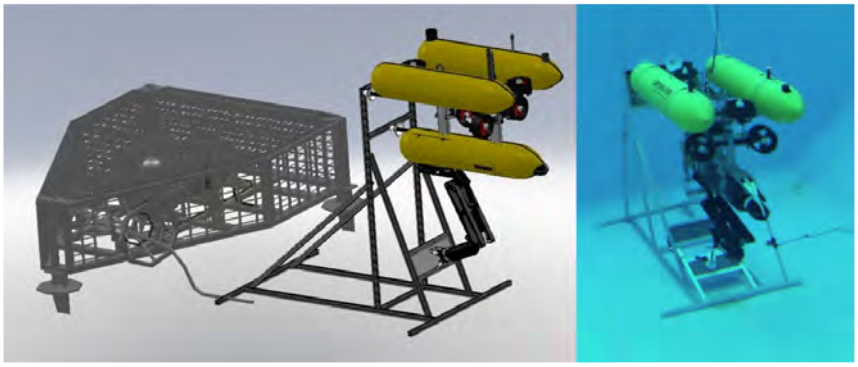


Figure 4.8: TRITON concept

4.1.8 PANDORA 2012-14

The main goal of the PANDORA project was to provide persistent autonomy to underwater robots, reducing the frequency of assistance requests [Lane et al.2012] (see Figure 4.9). The key to this objective is the ability to recognise failures and respond to them at all levels of abstraction and with time constraints. The project developed three themes: (1) describing the world for detecting failures in the task execution; (2) directing and adapting intentions by means of planning for responding to failures; and (3) acting robustly mixing learning and robust control to make actions resilient to perturbations and uncertainty. The project centred its validation tasks on AUVs acting in an oil field scenario in which the robots perform inspection, chain cleaning and valve turning tasks. This thesis has focused on the valve turning scenario proposed by this project. More details of this project have been presented in Sections 1.1 and 4.2. The results obtained in the underwater manipulation are presented in Section 5.

4.1.9 Hybrid AUV inspection, monitoring, and intervention of seafloor and sub-seafloor pipelines

This recent project proposes the cooperation between a surface and an underwater vehicle [Kaiser et al.2013] (see Figure 4.10). In this case, a wave glider is used on the surface for vehicle tracking and as a communications gateway. A Hybrid Remotely Operated

Figure 4.9: PANDORA concept

Vehicle (HROV), working in AUV mode, is used to perform an autonomous survey close to the sea-floor looking, for instance, for leakages in buried pipes. If an anomaly is detected, the underwater vehicle performs a detailed survey of the area while waiting for the assistance of a surface support vessel. By lowering an optical communications package, the surveyed data is downloaded and the vehicle is switched to ROV mode to perform a light intervention while a work-class ROV is brought to the site for heavier work. Although, this is a very recent project, wireless intervention for the tele-operation of NEREUS HROV through a high bandwidth optical/acoustic communication system has already been demonstrated [Farr et al.2010].

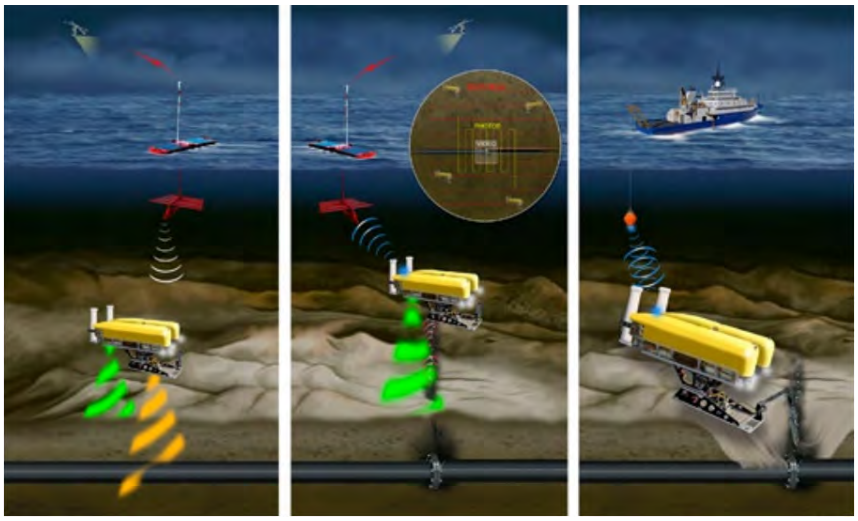


Figure 4.10: Hybrid AUV inspection, monitoring, and intervention of sea-floor and sub-sea pipelines concept.

4.1.10 MARIS 2013-16

The MARIS project goal is to develop underwater robotic systems for manipulation and transportation in the off-shore industry, in search-and-rescue tasks, and in various flavours

of scientific exploration [Casalino et al.2014]. As a key distinctive feature, MARIS aims at the development of AUVs capable of interacting with and in the underwater environment by object manipulation. The most interesting problem tackled within MARIS is the capability of dexterous manipulation for floating manipulation systems using a single vehicle or a cooperation of vehicle. The mission of the single or multiple vehicles is to grasp, manipulate and transport and object guaranteeing the safety of the object and the vehicle (see Figure 4.11).

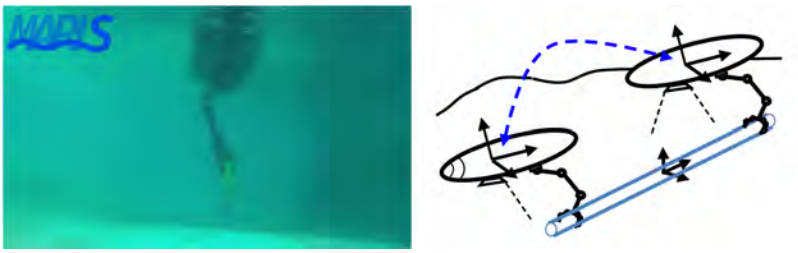


Figure 4.11: MARIS concept

4.1.11 MERBOTS 2015-2017

MERBOTS project designs a methodology for an underwater cooperative intervention, composed by up to three heterogeneous vehicles [MERBOTS2016]. The aim of the project has been focused in underwater archaeology, although the same technologies could be applied in other scenarios. The goal of the mission developed is first survey an area, then perform a second more detailed survey to inspect the interesting elements detected, and finally recover some of them. Figure 4.12 shows the two distinct phases of this mission. On the left, it is represented a cooperative survey phase composed by an AUV performing the survey, and an Autonomous Surface Vehicle (ASV) providing absolute localisation and communication to the AUV. On the right, it is represented a cooperative semi-autonomous intervention phase, developed by means of an HROV assisted by an AUV providing an external view of the intervention, and an ASV providing communication and localisation.

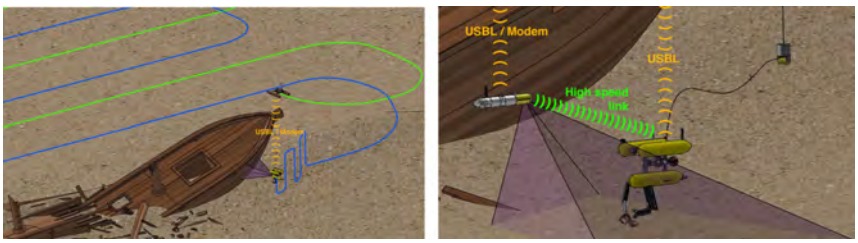


Figure 4.12: MERBOTS concept

Furthermore, to develop the mission with the proposed methodology requires a good

communication link between vehicles. Hence, the project is developing new wireless underwater communication systems, allowing the vehicles to exchange commands and images.

4.1.12 DexROV 2015-2018

DexROV proposes implementing and evaluating novel operation paradigms focused on safety, cost reduction and time efficiency for ROV operations [Gancet et al.2015] (see Figure 4.13). This project proposes using an HROV deployed from a vessel with a reduced crew, and linked to a control center by satellite communications. In this sense the cost of deployment and control will be reduced. The HROV is equipped with two bi-dexterous manipulators. This new capability will allow the ROV to accomplish tasks that till now have only been performed by divers, and extend this kind of intervention to deep-water scenarios. Even if the project does not focus on autonomy intervention, it is relevant for innovative approaches related to underwater manipulation.

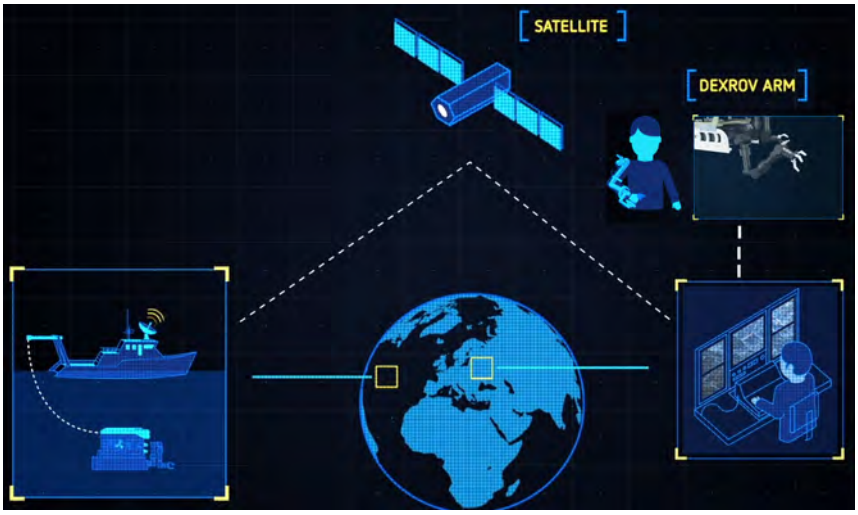


Figure 4.13: DexRov concept.

4.1.13 ROBUST 2015-20

The ROBUST project [ROBUST2016] aims to tackle the need to develop an autonomous, reliable, cost effective technology to map vast terrains, regarding mineral and raw material contents. This will aid in reducing the cost of mineral exploration, currently performed by ROVs and dedicated Seismic Support Vessels (SSVs). Furthermore, there is a need to identify, in an efficient and non-intrusive manner (minimum impact to the environment), the richest sites.

The project goal is to develop seabed in-situ material identification through the fusion of two technologies, a laser-based in-situ element-analyzing capability, called Laser Induced

Breakdown Spectroscopy (LIBS), merged with AUV technologies for seabed 3D mapping. This AUV (see Figure 4.14) equipped with both technologies will generate a general 3D seabed mapping where the possible mineral deposits will be identified. Then the vehicle endowed with a manipulator bearing the LIBS probe will perform a close scan of the identified targets (mineral deposits).

Figure 4.14: ROBUST concept

4.1.14 OceanOne: underwater humanoid

OceanOne is a project devoted to the creation of a new kind of ROV for underwater intervention [Lab2016], born from the need to study coral reefs deep in the Red Sea (Egypt). No existing vehicles at that point were able to dive with the skill and care of a human diver to study the coral reef.

Hence, OceanOne is a bi-manual underwater humanoid robot with haptic feedback allowing human pilots an unprecedented ability to explore the depths of the oceans in high fidelity (see Figure 4.15). The hands of the OceanOne have full articulated wrist equipped with force sensors that relay haptic feedback to the pilot's controls, so the human can feel whether the robot is grasping something firm and heavy, or light and delicate and actuate in consequence.

Until these days the robot has been used successfully to study coral reef and the exploration of shiprecks(La Lune, France 2016) recovering delicate archaeological elements.

4.1.15 Crabster CR200

Crabster CR200 (see Figure 4.16) [Yoo et al.2015] is an alternative to propeller-driven ROVs and AUVs which are ill-equipped to deal with strong tidal currents in shallow seas. To overcome this hurdle Crabster is designed to scuttle the seabed using six legs, like a crustacean. This design is more stable under strong currents and will not stir up as

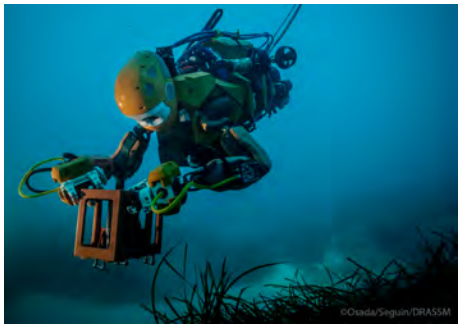


Figure 4.15: OceanOne

much debris as propellers. Moreover, like a crab or a lobster, the robot's two front legs are equipped with manipulators that can grasp objects, which can be stored in a frontal compartment. The researchers also designed the robot's shell to deflect strong currents by adjusting its overall posture.

Figure 4.16: Crabster CR200

Crabster was designed from the need to help the archaeologist to examine 12th century shipwrecks in the Yellow Sea. However, it can be used in the off-shore industry to work in pipelines or even in rescue operations. For instance in April of 2014 during the sinking of MV Sewol Ferry in South Korea, the Crabster and its scientific team collaborate with the coast guard for more than 15 hours elaborating an exhaustive survey of the sunken ferry.

4.2 Definition of the mock-up valve turning environment

The work in this thesis has been devoted to resolving the valve turning task proposed in the PANDORA project using an I-AUV in a controlled underwater environment. All the experiments have been performed in the water tank at the CIRS. Since the installation doesn't possess any real sub-sea panel (see Figure 4.17), the authors have built a mock-up sub-sea panel composed of T-Shape valves (see Figure 4.18). Each valve is labelled as to its function, similar to a real sub-sea panel. This panel can be easily installed in the water tank's wall and can be moved from one position to another during the experiments.

The panel can contain from 1 to 4 valves. The necessary torque to turn each valve can

(a) Sub-sea panel with docking bars from the camera of an Intervention ROV of OCEANEERING

(b) Subsea panel from the camera of an Intervention ROV of DEEPOCEAN.

Figure 4.17: Two sub-sea panel images recorded during different underwater interventions in the off-shore industry.

Figure 4.18: The mock-up sub-sea panel built for the PANDORA project. The panel is composed of 4 T-shape valves.

be adjusted at different levels, even blocked. Furthermore, the turn of each valve is limited to 90 degrees.

Moreover, the setup includes different kinds of perturbations to increase the realism of the water tank environment. A perturbation system (see Figure 4.19), composed of one or two thrusters has been built with this goal. The system can be configured to generate perturbations at different intensities and orientations. Other perturbations that can be generated are: movement of the sub-sea panel, movement of the I-AUV, movement of the end-effector, and visual occlusion of the valve panel (see Figure 4.20).

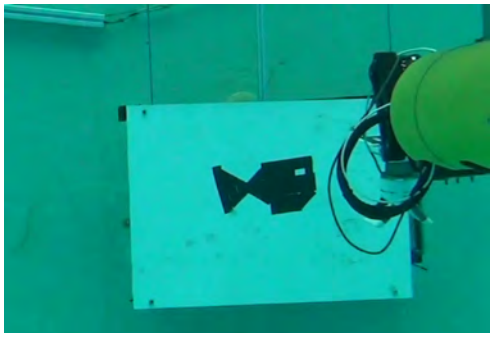
4.3 Vehicle Girona 500 AUV

The GIRONA 500 AUV [Ribas et al.2012] is a compact-size AUV designed and developed in the University of Girona for a maximum operating depth of 500m (see Figure 4.21).The

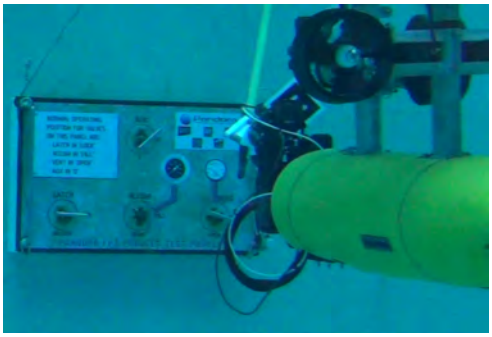
(a) Perturbation system composed by one thruster.

(b) Pertubation system composed by two thrusters.

Figure 4.19: The two different perturbation systems built to simulate underwater currents.



(a) White panel, between the valve panel and the Girona 500 I-AUV to avoid the detection of the panel



(b) Stick hitting the manipulator to simulate an unexpected object in the trajectory of the manipulator.

Figure 4.20: Two different kinds of disturbances, performed with the AUV during the manipulation.

vehicle is built around an aluminium frame which supports three torpedo-shaped hulls as well as other elements like the thrusters. The overall dimensions of the vehicle are 1 m in height, 1 m in width, 1.5 m in length and a weight, on its basic configuration, of about 140 kg. The two upper hulls, which contain the flotation foam and the electronics housing, are positively buoyant, while the lower one contains the heavier elements such as the batteries and the payload. This particular arrangement of components provides the vehicle with passive stability in *pitch* (ϕ) and *roll* (θ), making it suitable for tasks requiring a stable platform such as video surveying or intervention.

4.3.1 Mechatronics description

The most remarkable characteristic of the GIRONA 500 is its capacity to be reconfigured for different tasks. On its basic configuration, the vehicle is equipped with typical navigation sensors like DVL, Attitude and Heading Reference System (AHRS), pressure gauge and Ultra Short BaseLine (USBL) as well as basic survey equipment like profiler sonar, side scan sonar, video camera and sound velocity sensor. In addition to these sensors, almost half the volume of the lower hull is reserved for a mission-specific payload, which makes it

Figure 4.21: Underwater image of the GIRONA 500 AUV during a test.

possible to modify its sensing and actuation capabilities as required. A similar philosophy has been applied to the propulsion system which can be set to operate with a different number of thrusters, ranging from 3 to 8, to actuate the necessary degrees of freedom and provide, if required, some degree of redundancy.

Figure 4.22: 3D representation of the insides.

In the context of autonomous intervention, several payloads have been developed according to the needs of each task. The payload used for this thesis and the PANDORA project integrates a small size 4 DOFs manipulator into the GIRONA 500 AUV to demonstrate the autonomous free-floating operation of valves on an intervention panel (see Figure 4.23). For that purpose, a fixed V-shaped tool was installed as an end-effector to actuate the valves. The custom end-effector includes a camera and a F/T sensor (see Section 4.5 for more details). The manipulator was mounted on the front part of the vehicle to provide a convenient workspace. Finally, a stereo camera and an illumination system was installed to view the panel and the manipulator.

4.3.2 Software architecture

The software architecture for intervention proposed in this thesis is composed of several modules which are organized in four layers (see Figure 3.2 in Section 3.3). Starting from

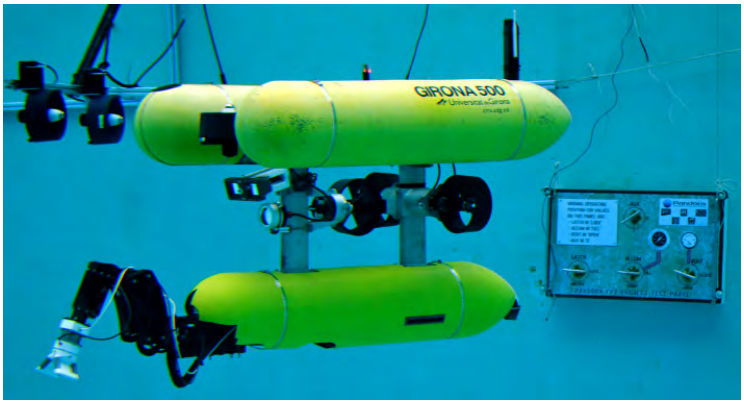


Figure 4.23: GIRONA 500 AUV with the PANDORA payload in the water tank.

the bottom, the first layer contains all the sensor and actuator drivers. The next layer processes sensor data. It contains the localization filter and perception modules that process the cameras and the F/T sensor. On top of that, the control layer includes the AUV and manipulator controllers in charge of following the commands requested by the LbD system. Finally, in the top level layer, the LbD system is in charge of acquiring data from the demonstrations (i.e., phase 1 from LbD algorithm), learning the model (phase 2) and reproducing the task by generating velocity requests (phase 3).

The next sections focus on the I-AUV control architecture, explaining in more detail the different layers describing the functionalities and interactions of each element. Figure 4.24 shows a detailed diagram of the software architecture for the GIRONA 500 I-AUV.

4.3.3 Sensing and actuation

Starting from the bottom of Figure 4.24, this layer contains all the sensors and actuator drivers. It uses the AHRS to measure the angular velocity (ν_2) and the robot's attitude (a_2), the DVL to measure the linear velocity (ν_1), the depth sensor included within the Sound Velocity Sensor (SVS) is used to measure the depth (z), and a stereo camera is used to gather stereo imagery ($I = \{I_1, I_2\}$). This layer also includes the drivers for the thruster actuators, which are controlled in speed (revolutions per minute (rpm)).

4.3.4 Perception module: Visual detector

The visual detector is the main element in the *perception module*. In our proposal, a vision-based algorithm analyses the images gathered and compares them with an a priori known template of the sub-sea panel. With this information, the main system is able to detect the presence of the sub-sea panel, as well as accurately estimating its position and orientation when a sufficient number of features are matched between the current image and the template.

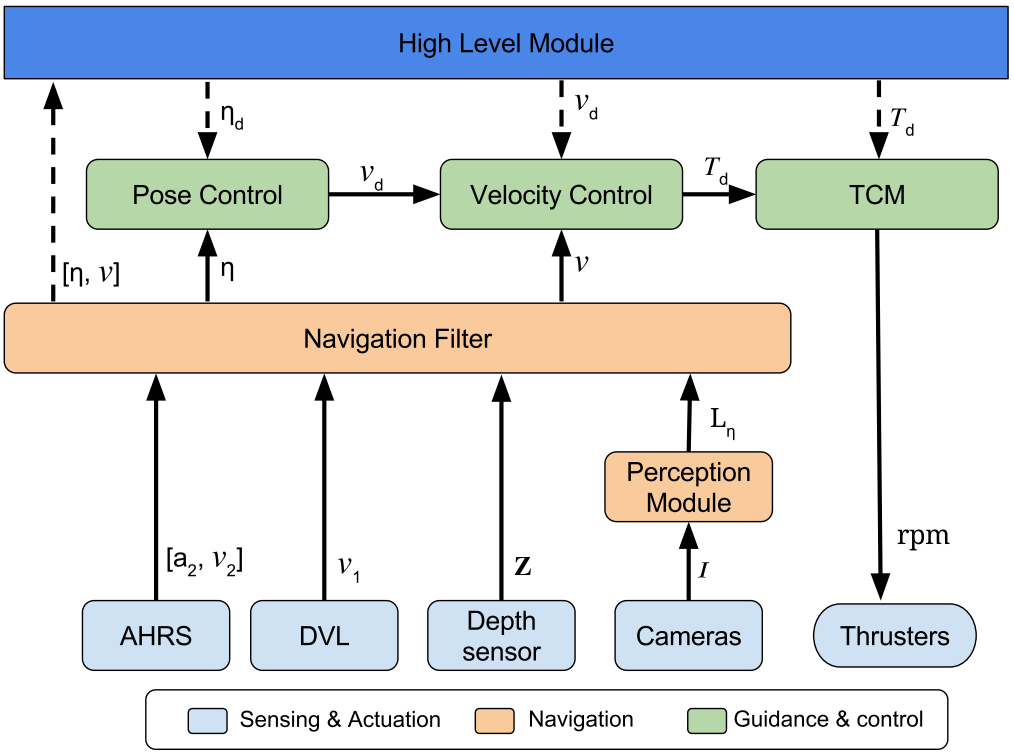


Figure 4.24: Detailed diagram of the I-AUV control architecture layer.

The proposed method uses oriented FAST and rotated BRIEF (ORB) [Rublee et al.2011] feature extractors for their suitability in real-time applications. The ORB feature extractor relies on features from an accelerated segment test (FAST) corner detection [Rosten and Drummond2006] to detect features or key-points in the image. These are obvious features to detect on man-made structures and can be detected. Moreover, there is a descriptor (binary) vector of the key-point based on binary robust independent elementary features (BRIEF) [Calonder et al.2010]. This allows us to obtain the difference between descriptors rapidly and allows real-time matching of key-points at higher image frame-rates when compared to other commonly used feature extractors such as Scale-Invariant Feature Transform (SIFT) [Lowe2004] and Speeded-Up Robust Features (SURF) [Bay et al.2008].

Figure 4.25 illustrates the matching between the panel template and the image received from the camera. A minimum number of key-points must be matched between the template and the camera image to satisfy the panel detection requirement. A low number of matched key-points indicates that the panel is not in the camera's field of view (less than 30 in our case). The correspondences between the template and camera image can be used to compute the transformation, or homography, of the template image to the detected panel in the camera image. This allows us to compute the image-coordinates of the corners of the panel in the camera image. Using the known geometry of the panel and the camera's

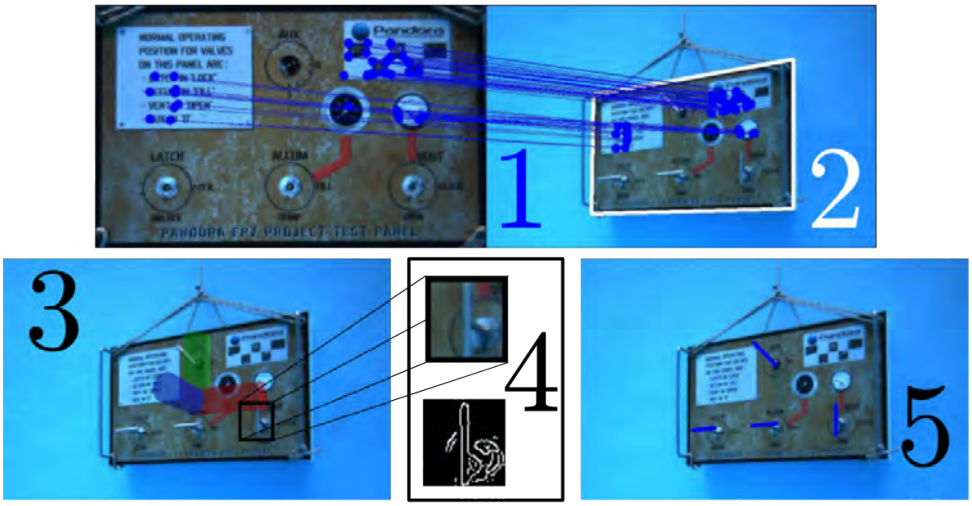


Figure 4.25: Steps in the image analysis performed on an image to obtain the position and orientation of the valves in the mock-up sub-sea panel.

matrix, we are able to determine the pose of the panel in the camera's coordinate system. This process corresponds with steps 1, 2, and 3 of Figure 4.25.

Due to the known relationship between the panel's centre and the valve, it is possible to estimate the position of the valves in the camera's coordinate system. Taking advantage of this, the proposed method extracts a small bounded region in the image to search for the orientation of each valve. The Hough line transform is used to detect the main line in this bounded region that corresponds to the valves' orientation. Outliers are limited by constraining the length of the lines and the orientations that can be detected. Steps 3, 4, and 5 in Figure 4.25 illustrate this process.

This method can be used with images from the AUV's camera or the end-effector's camera. The AUV's camera has a wider field of view, which allows the detection of the panel throughout the entire intervention, while the camera in the end-effector is only useful for short periods of time. Moreover, the panel and valve positions detected by the end-effector's camera are less precise and reliable than the AUV's camera due to the uncertainty associated with the position of the end-effector. The author has experimented with different approaches in which both systems have been used simultaneously or have alternated between the two cameras, depending on the stage in the intervention task. The best results have been obtained using only the AUV's main camera throughout the whole intervention, disabling the detection of the valves' orientation during the final part. This is necessary because the end-effector occludes the valve in the final part of the intervention and the algorithm confuses the end-effector for the valve and this changes the estimated orientation. However, the camera in the end-effector has been useful during the tele-operation phase, improving the understanding of the intervention.

The panel and valves' pose are provided to the *navigation filter* and to the *high level modules* to be used to improve the localization of the vehicle and also to define the frame in which the LbD algorithm learns and executes the trajectory.

4.3.5 Navigation filter

An Extended Kalman Filter (EKF) [Kalman1960, Kalman and Bucy1961] is in charge of estimating the vehicle's position and velocity ($\vec{x_k} = [\eta_1^T \, \nu_1^T]^T$). Vehicle orientation and angular velocity are not estimated but directly measured by an AHRS. This filter is also able to map the pose in a world with various landmarks, working as a SLAM. Even though the filter is designed to deal with several landmarks, in the valve turning task a single landmark (i.e., the sub-sea panel) is used. A vision-based algorithm [Palomeras et al.2013] included in the perception module (Section 4.3.4) identifies the intervention panel and computes its relative position.

The information to be estimated by the SLAM algorithm is stored in the following state vector:

$$x_k = [x \; y \; z \; u \; v \; w \; l_1 \; ... \; l_n]^T, \tag{4.1}$$

where ($[x \; y \; z \; u \; v \; w]$) are vehicle position and linear velocity and ($l_i = [lx_i \; ly_i \; lz_i \; l\phi_i \; l\theta_i \; l\psi_i]$) is the pose of a landmark in world coordinates.

A constant velocity kinematic model is used to determine how the vehicle's state will evolve from time $k-1$ to k. Landmarks are expected to be static. The predicted state at time k, x_k^- follows the equations:

$$x_k^- = f(x_{k-1}, \; n_{k-1}, \; u_k, \; t). \tag{4.2}$$

$$x_k^- = \begin{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \end{bmatrix} + R(\phi_k\ \theta_k\ \psi_k) \left(\begin{bmatrix} u_{k-1} \\ v_{k-1} \\ w_{k-1} \end{bmatrix} t + \begin{bmatrix} n_{u_{k-1}} \\ n_{v_{k-1}} \\ n_{w_{k-1}} \end{bmatrix} \frac{t^2}{2} \right) \\ u_{k-1} + n^t_{u_{k-1}} \\ v_{k-1} + n^t_{u_{k-1}} \\ w_{k-1} + n^t_{u_{k-1}} \\ l_{1_{k-1}} \\ \dots \\ l_{n_{k-1}} \end{bmatrix}, \tag{4.3}$$

where t is the time period, $u = [\phi\ \theta\ \psi]$ is the control input determining the current vehicle orientation and $n = [n_u\ n_v\ n_w]$ is a vector of zero-mean white Guassian acceleration noise whose covariance, represented by the system noise matrix Q, has been set empirically:

$$Q = \begin{pmatrix} \sigma^2_{n_u} & 0 & 0 \\ 0 & \sigma^2_{n_v} & 0 \\ 0 & 0 & \sigma^2_{n_w} \end{pmatrix} \tag{4.4}$$

Associated with the state vector x_k there is the covariance matrix P_k. Following the standard EKF operations, the covariance of the prediction at time k is obtained as:

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T, \tag{4.5}$$

where A_k is the Jacobian matrix of the partial derivatives of f with respect to the state 4.1 and W_k is the Jacobian matrix of the partial derivatives of f with respect to the process noise n.

Three linear measurement updates are applied in the filter: pose, velocity and landmark updates. Each sensor measurement is modelled as:

$$z_k = H x_k + s_k, \tag{4.6}$$

where z_k is the measurement itself, H is the observation matrix that relates the state vector with the sensor measurement, and s_k is the sensor noise. Updates are applied by

means of the equations:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}, \quad (4.7)$$

$$x_k = x_k^- + K_k(z_k - H x_k^-), \quad (4.8)$$

$$P_k = (I - K_k H) P_k^-, \quad (4.9)$$

where K_k is the Kalman gain, R the measurement noise covariance matrix and I an identity matrix. Below, how to define z_k and H for each sensor to perform the updates applying equations (4.7) is shown.

Several sensors provide the position information, which can be used to initialize the vehicle's position and bound dead-reckoning errors. A GPS receiver measures the vehicle's position in the plane (x, y) while the vehicle is at the surface, a pressure sensor transforms the pressure values into depth (z) and a USBL device measures the vehicle's position (x, y, z) while submerged. To integrate any of these position sensors the following is applied:

$$z_k = [x\ y\ z], \quad (4.10)$$

$$H = [I_{3x3}\ O_{3x3}\ O_{3x6n}], \quad (4.11)$$

where I_{3x3} denotes the 3 x 3 identity matrix and O_{3x6n} denotes the 3 x $6n$ zero matrix with n being the number of landmarks. If only (x, y) or (z) is available, z_k and H have to be properly arranged.

Velocity updates are provided by a DVL sensor that measures linear velocities with respect to the sea bottom or the water below the vehicle. To integrate the velocity the following is applied:

$$z_k = [u\ v\ w], \quad (4.12)$$

$$H = [O_{3x3}\ I_{3x3}\ O_{3x6n}]. \quad (4.13)$$

When only velocity updates are available, the filter behaves as a deadreckoning algorithm that drifts over time. However, if position updates or landmarks are detected, the localization filter is able to keep its error bounded.

Landmarks are introduced to the filter using the relative position and orientation between the vehicle and the landmark. This information is introduced in the localization filter to improve both the vehicle and panel's landmark position. To integrate the relative

measurement is applied:

$$z_k = [L_x\ L_y\ L_z\ L_\phi L_\theta\ L_\psi], \tag{4.14}$$

$$H = \begin{bmatrix} -\mathbf{Rot}^T & O_{3x3} & \mathbf{Rot}^T & O_{3x3} & \cdots \\ O_{3x3} & O_{3x3} & O_{3x3} & I_{3x3} & \cdots \end{bmatrix}, \tag{4.15}$$

where $[L_x\ L_y\ L_z]$ is the relative position of the landmark with respect to the vehicle, $[L_\phi\ L_\theta\ L_\psi]$ is the landmark's orientation with respect the world and **Rot** is the vehicle's orientation rotation matrix.

Figure 4.26 presents a block diagram showing how each navigation sensor is related to the localization filter. The state vector is initialized according to the GPS and pressure sensor or the USBL if it is present and the vehicle is submerged. Linear velocities are set to zero. No landmarks are present when the filter is initialized. The first time a landmark is identified, its pose is introduced in the state vector as a landmark by compounding its relative position with respect to the vehicle's position; this being the position contained in the state vector.

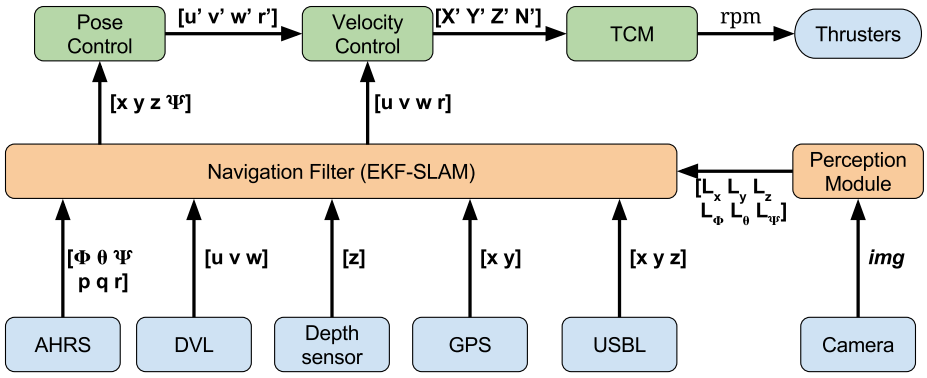


Figure 4.26: Detailed schema of the EKF-SLAM of the navigation filter.

4.3.6 Guidance and control

The GIRONA 500 I-AUV can be actuated in 4 DOF (surge, sway, heave and yaw) while it is stable in roll and pitch. It can be controlled by means of body force requests (T_d or $[X'\ Y'\ Z'\ N']$), body velocity requests (ν_d or $[u'\ v'\ w'\ r']$) and waypoint requests (η_d or $[x'\ y'\ z'\ \psi']$). A cascade control scheme is used to link these controllers, as shown in Figures 4.24 and 4.26.

The first controller is a 4 DOF Proportional-Integral-Derivative (PID) called pose control. It receives as inputs the vehicles pose ($[x\ y\ z\ \psi]$) and the desired waypoint($[x'\ y'\ z'\ \psi']$). The pose controller output is the desired velocity($[u'\ v'\ w'\ r']$). The standard PID form is

used:

$$\nu'(t) = K_p\Big(e(t) + K_i \int_0^t e(t)dt + K_d \frac{d}{dt}e(t)\Big), \tag{4.16}$$

where $e(t)$ is the error for each DOF $[e_x, e_y, e_z, e_\psi]$ computed as:

$$\begin{bmatrix} e_x \\ e_y \\ 1 \end{bmatrix} = \begin{pmatrix} \mathbf{R}(\psi)^T & -\mathbf{R}(\psi)^T & \begin{bmatrix} x \\ t \end{bmatrix} \\ 0_{1x2} & 1 & \end{pmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}, \tag{4.17}$$

$$e_z = z' - z, \tag{4.18}$$

$$e_\psi = normalized(\psi' - \psi), \tag{4.19}$$

where $\mathbf{R}$ is a 2D rotation matrix.

The pose controller output (ν_d) is sent to the velocity controller together with the vehicle's current velocity (ν) following the cascade scheme. The velocity controller computes the desired force and torque by combining a 4 DOF PID (4.16) with an open loop model-based controller. Here, $e(t)$ is computed directly by subtracting the desired velocity for each DOF to the current one (e.g., $e_u = u' - u$). The open loop model-based controller generates the force and torque needed to keep a constant velocity without considering the vehicle's current velocity.

$$T_d = \text{PID}(\nu, \nu_d) + \text{Model}(\nu_d). \tag{4.20}$$

The output of this controller is the desired force and torque (T_d) defined in the vehicle's body frame. The desired force is later allocated to the thrusters using the thruster allocation matrix within the *Thurster Control Matrix* (TCM) module. Once the force to be exerted by each thruster is known, a static thruster model is used to convert from force to the thruster setpoint.

From the several control options presented in this thesis, the velocity control has been most frequently used. The tele-operation and LbD modules send velocity commands (ν_d) according to the estimated pose (η) from the *Navigation filter*.

4.4 ECA ARM 5E manipulator

The ECA ARM 5E MICRO is a commercial manipulator manufactured by the ECA group. This manipulator is composed of 4 DOFs and a gripper, all of which are powered by brushless electrical motors. The Manipulator is designed to be used in light ROVs with a weight of 2.75 kg in the water and 10 kg outside, offering a lifting capacity of 10 kg. The system is able to operate in up to 300 m depths without oil, the oiled version is able to

resist up to 3000 m.

Figure 4.27: ECA ARM 5E Micro.

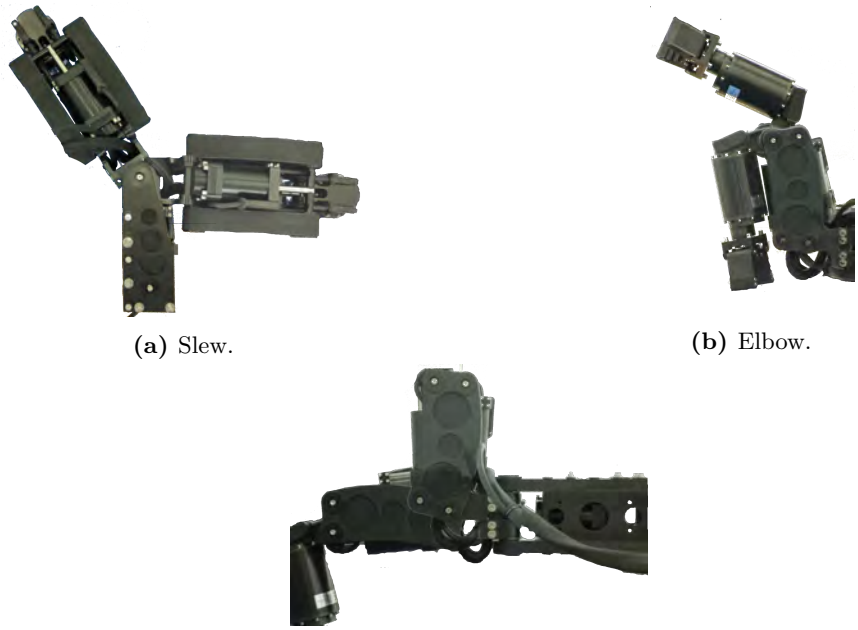
4.4.1 Mechatronics description

The manipulator is composed of 4 cylinders mounted on an aluminium frame. Each cylinder contains a brushless electric motor, except for the last cylinder which contains two. The motors provide a linear force which is converted into angular movement by the frame's design. For this reason, the joints have a limited work area (see Figure 4.28). The *slew* is the only joint controlling the movement in the horizontal plane, and is able to move 120 degrees (Figure 4.28a). The construction design allows control of only one side of it from the base of the manipulator. The *elevation* controls the vertical movement with a workspace of 90 degrees (Figure 4.28c). The *elbow* also actuates in the vertical plane with a workspace of 130 degrees (Figure 4.28c). The *wrist* is a continuous joint controlling the orientation of the end-effector/gripper, so it has no physical limits. Finally, the gripper can be actuated using the last motor, the maximum aperture on the tip of the gripper is 125 mm, and can perform with a force of 50 kg.

The manipulator is equipped with an extra cylinder with the electronics to control the joints and to establish communication with the vehicle. The communication between the electronics and the vehicle is made using an RS-485 link, and a proprietary protocol.

4.4.2 Software architecture

The ECA ARM 5E Micro manipulator is a product developed for use in light ROVs, hence the software offered by the manufacturer is limited to a direct control of each joint. For this reason, a proper control software has been developed to operate the manipulator. Figure 4.29 shows a diagram of this controller that uses a similar schema to the GIRONA 500 AUV.



(a) Slew.

(b) Elbow.

(c) Elevation.

Figure 4.28: Images of the manipulator reaching the limits of the slew, elbow and elevation joints.

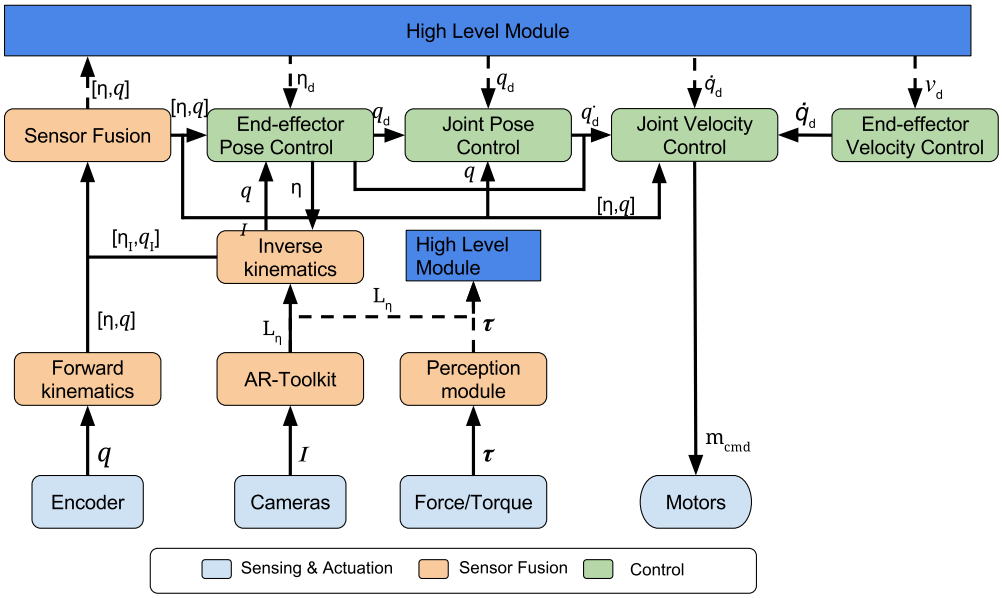


Figure 4.29: Detailed diagram of the Manipulator layer.

4.4.3 Encoders

The ECA ARM 5E Micro does not dispose of absolute encoders, instead it uses a hall sensor. This implies that every time the power is removed, the encoders' position is reset. For this reason, the authors have developed a calibration method to find the absolute

position of each joint. This method is explained in detail in Section 4.4.8.

A conversion from the motor's linear movement to the joint's angular movement is necessary because the steps of the hall sensor don't have a constant relationship with the increment on the joint angle. For this reason, the mechatronics design and the step range of the encoder have been used to estimate the joint movement. A linearisation formula has been defined to convert the encoder values to angular positions. However, this method has a drift over time that forces a repetition of the calibration process after a certain period.

4.4.4 Forward Kinematics (FK)

The FK is a mathematical method based on kinematic equations to compute the end-effector position of the robotic arm or manipulator using the joints position. The robotics community has established as a convention the principles based on the Denavit-Hartenberg (D-H) [Denavit and Hartenberg1965] matrices to define the FK. In this convention, coordinate frames are attached to the joints between two links so that one transformation is associated with joint $[Z]$, and the second is associated with link $[X]$.

$$[T] = [Z_1][X_1][Z_2][X_2]\dots[Z_n][X_n]. \tag{4.21}$$

The coordinate transformations along the manipulator consist of the concatenation of these transformations until the end-effector $[T]$. To define the Z and X transformations, four parameters are used:

- d offset along the previous z to the common normal.
- θ angle about the previous z, form old x to new x.
- a length of the common normal. Assuming a revolute joint, this is the radius about the previous z.
- α angle about the common normal, from the old z axis to the new z axis.

In Figure 4.30, the robot arm is in *home* position (defined by the authors), where all the joints are equal to zero. Each joint is indicated using a frame and distances between joints are pointed out by their names (a_n, d_n). Table 4.1 shows the 4 parameters defining the D-H convention to construct the FK. The values for the constant distances are $a1 = 0.108$m , $a2 = 0.264$m , $a3 = 0.100$, and $d4 = 0.308$.

4.4.5 Inverse Kinematics (IK)

The IK is the method used to compute the joints' configuration of a manipulator given a desired position and orientation for the end-effector. There are two different approaches to solve this problem; the geometric approach and the algebraic approach. The most popular

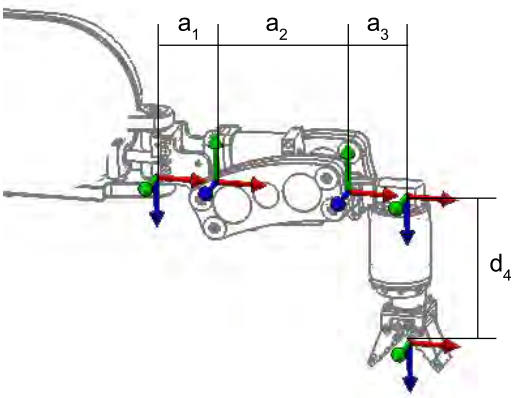


Figure 4.30: Schema of the ECA ARM 5E Micro, in the defined home position with the different joints marked with a frame. The red, green, and blue define the x, y and z axis of the frame.

DOF	θ	d	a	α	Home
1	q_1	0	a_1	-90	0
2	q_2	0	a_2	0	0
3	q_3	0	a_3	90	0
4	q_4	d_4	0	0	0

Table 4.1: D-H table describing the kinematic chain of the ECA ARM 5E Micro.

is the algebraic approach, which uses an iterative method (i.e., [Grudic and Lawrence1993]) to find the solution.

In our case, since the manipulator is under actuated (less than 6 DOFs) the algebraic solutions have difficulties finding the correct solution. Nevertheless, it is quite simple to find the IK using the geometric approach.

Figures 4.31, 4.32, and 4.33 show the configuration position used to do the geometric analysis to extract the equations of the IK. In the following lines these equations are detailed.

To find the value for the *slew* angle (q_1), only the x_{ee} and the y_{ee} of the desired position are used:

$$q_1 = \text{atan2}(y_{ee}, x_{ee}), \tag{4.22}$$

as shown in the diagram in Figure 4.31. From this configuration we can also extract the distance between the base and the desired position of the end-effector (d).

$$d = \sqrt{x_{ee}^2 + y_{ee}^2}. \tag{4.23}$$

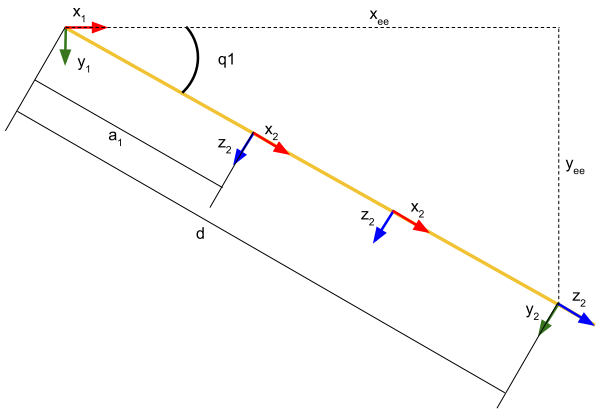


Figure 4.31: Diagram of the geometric configuration to extract the equations used to compute the value for the *slew* (q_1). The yellow lines represent the structure of the manipulator and the joints are represented by the axis of the frame in the represented 2D plane.

The equation for the *elbow* angle (q_3) is defined by:

$$q_3 = -\pi + \beta + \alpha. \tag{4.24}$$

This equation is obtained from Figure 4.32. α and β are angles, which need to be found. Using the cosine theorem, angle β is computed using l and a_2:

$$\beta = \text{acos}\left(\frac{t^2 - a_2^2 - l^2}{-2a_2 l}\right), \tag{4.25}$$

where $l = \sqrt{a_3^2 + d_4^2}$, and $t = \sqrt{x_e e^2 + (d - a_1)^2}$. α value is computed as follows:

$$\alpha = \text{atan2}(d_4, a_3). \tag{4.26}$$

The desired value of the *elevation* angle (q_2) is computed using the following equation

$$q_2 = \gamma + \phi, \tag{4.27}$$

extracted from Figure 4.33, where γ and ϕ are computed as follows:

$$\gamma = \text{atan2}(l\text{sin}(\alpha - q_2), a_2 + (l\text{cos}(\alpha - q_2))) \tag{4.28}$$

$$\phi = \text{atan2}(-z_{ee}, d - a_1). \tag{4.29}$$

The proposed IK controller allows control of x, y, z and *roll*. It is worth noting that while *roll* is a direct conversion between the *wrist* (q_4) and the desired *roll*, *pitch* and *yaw* are not controlled in order to increase the range of movements because most of the

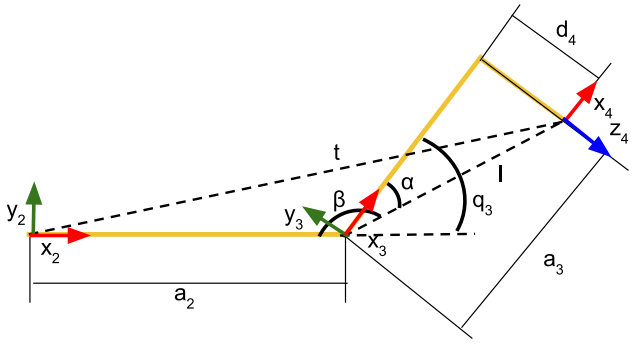


Figure 4.32: Diagram of the geometric configuration to extract the equations used to compute the value for the *elbow* angle (q_3). The yellow lines represent the structure of the manipulator and the joints are represented by the axis of the frame in the represented 2D plane.

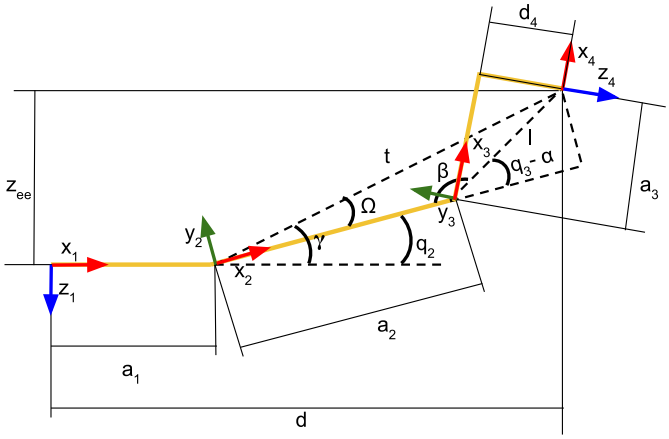


Figure 4.33: Diagram of the geometric configuration to extract the equations used to compute the value for the *elevation* angle (q_2). The yellow lines represent the structure of the manipulator and the joints are represented by the axis of the frame in the represented 2D plane.

positions can only be reached in one particular *pitch* and *yaw* configuration.

This geometric approach doesn't consider the joints limits. Hence, before returning the position for the requested end-effector, the method checks if the obtained values are within the joints' limits.

4.4.6 Perception module: Contact detection and torque control

As a complementary part of the *perception module* in the GIRONA 500 AUV explained in Section 4.3.4, the data obtained from the F/T sensor (Section 4.5.3) is filtered and processed in this module to provide information to the *high level modules*.

In the case of valve manipulation, the F/T information is used in two stages of the intervention. First, to confirm the successful grasping of the valve. The second corresponds

to the application of a constant force and torque while turning the valve without exceeding a predefined maximum. The constant pressure allows a better intervention under external perturbations (e.g., currents) and the limitation of the torque indicates if the valve is locked and prevents damaging the manipulator and the sub-sea panel.

4.4.7 Control

In order to be able to carry out the greatest number of applications, the control designed for the manipulator accepts multiple types of requests. The control algorithm below follows a cascade controller to group all the requests in the same controller. Figure 4.29 shows the proposed control schema.

The control is performed by four controllers that accept four different kinds of requests as explained below:

- **End-effector pose control**: A desired pose for the end-effector (η_d) is given, the first step for the control module is to check the feasibility of the pose using the IK and subsequently send the new desired joint positions (q_d) to the Joint Pose controller. If the desired η_d is requested in *isochronous* mode, where all the joints finish the movement at the same time, then the controller produces joint velocities ($\dot{q}_d$) which are sent directly to the joint velocity controller, skipping the joint pose control.

- **Joint pose control**: Similar to the end-effector pose controller first the feasibility of the desired joint position (q_d) is verified using the known limits of the joint, then the q_d is converted to velocity joint commands ($\dot{q}_d$) using a standard Proportional-Derivative (PD) controller:

$$\dot{q}(t) = K_p\left(e(t) + K_d\frac{d}{dt}e(t)\right), \tag{4.30}$$

  where the $e(t)$ is the error for each DOF joint position ($e = q_d - q$). Resulting velocity commands are sent to the Joint velocity controller.

- **End-effector velocity control**: The desired velocity of the end-effector (v_d) is converted to joint velocity commands ($\dot{q}_d$). The pseudo-inverse Jacobian ($J^\dagger$) of the manipulator is used to obtain velocity commands with the following equation:

$$\dot{q} = J^\dagger v_d. \tag{4.31}$$

- **Joint velocity control**: The joint velocity commands (q_d) are converted into manipulator commands (m_{cmd}) using a standard PD controller (4.30) where $e(t) = \dot{q}_d - \dot{q}$. The electronics in the manipulator has its own control to apply the m_{cmd} correctly. However, before sending the m_{cmd}, it is necessary to verify that q_d will not

drive the joint outside the workspace. If this is detected, the whole movement of the manipulator is stopped and an error message is sent to the system.

4.4.8 Calibration

Due to the manipulator using hall sensors to simulate the encoders, every time the electronics power down, the position obtained from the simulated joint encoder is restarted. To estimate the absolute values using these sensors, two procedures have been designed.

4.4.8.1 Mechanical Calibration

This method only uses the information provided by the manipulator. To do the calibration, the algorithm commands each joint to be moved in a predefined direction to find a known limit. The intensity consumed by the motor increases when a joint reaches its limit, at which point, the movement is stopped and the encoder's position is replaced by the corresponding value at this limit. The roll calibration has to be done manually since the motor has no physical limit. In this case, the human operator has to manually move the joint to the zero position.

This method is effective but slow, since all the joints have to reach their corresponding limit and this can take from 2 to 5 minutes, depending on the initial joint configuration. Furthermore, the human operator has to be present if the roll value has to be calibrated.

4.4.8.2 Visual Calibration

During an intense usage of the manipulator, it needs to be calibrated regularly to obtain the best precision. For this reason, a visual calibration procedure has been developed.

This method uses a visual detector and the previous mechanical procedure to calibrate the manipulator. Therefore, a visual marker has been installed at the link between the *elbow* and the *wrist* in order to estimate the pose of the end-effector. A stereo camera installed on the vehicle is used to identify the visual marker (see Figure 4.34) and estimate its pose.

The procedure to calibrate the manipulator using the visual marker works as follows. First, the manipulator is moved to a predefined position where, if the manipulator is approximately calibrated, the marker will be visible by the stereo camera. If the marker is not detected, the wrist will start slowly rotating the end-effector, first to 180° and then to -360°. During this process, if the marker is detected in a feasible position and orientation, the IK is computed and the joint values are updated with the values obtained through the visual feedback. Otherwise, the manipulator is moved to the limit of all the joints and calibrated using the mechanical calibration method. The visual landmark provides a more accurate calibration, and can be regularly applied whenever the user requires a precise

(a) Stereo camera inside an underwater housing.

(b) Image from the stereo camera while the GIRONA 500 I-AUV is performing an intervention task.

Figure 4.34: The image on the left shows the stereo camera used in the GIRONA 500 AUV and on the right an image recorded with this camera. The image is shown in grayscale to highlight the ARMarker in red.

calibration. Moreover, the calibration time is shorter compared to the mechanical solution, and the roll is automatically calibrated without the assistance of the operator.

The algorithm used to compute the pose of the marker with respect to the I-AUV camera relies on the *ARToolkit* software library [Kato and Billinghurst USA] to identify, detect and track markers using a monocular camera (see Figure 4.34b).

4.5 End-effector

In order to correctly perform a valve turning task using the ECA manipulator, a custom end-effector has been designed to minimize possible errors. The first goal of the custom end-effector is to compensate for small misalignments in *pitch* and *yaw* that cannot be compensated from the manipulator's side due to the reduced number of DOFs. These misalignments depend on the position and orientation of the AUV, which sustains the manipulator, and cannot be corrected at a centimetre scale (i.e., the AUV's motion precision is lower than the manipulator's). Also, there are always some detection and calibration errors which generate some inherent error in the position of the end-effector. The second goal is to integrate additional sensors to improve perception of the valve and the sub-sea panel. For the sake of these modifications, the manipulator's actuated griper has been replaced by a fixed one.

Figure 4.35 shows the custom end-effector, composed of three parts, explained in the following subsections.

4.5.1 Passive end-effector

This is the most external part of the end-effector, and has been designed in order to obtain the best possible grip with the manipulated element. In our case, to grasp the T-shaped valve correctly, two different shapes have been tested. Figure 4.36 shows a 3D model of

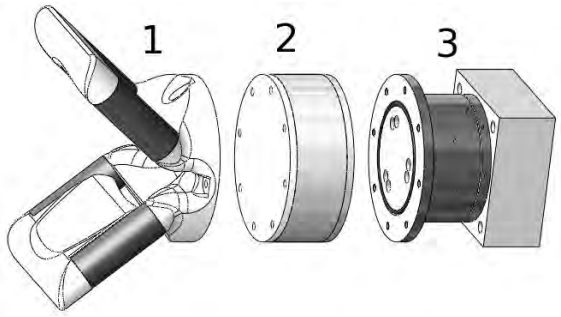


Figure 4.35: 3D model of the custom end-effector with the three different parts separated; the passive end-effector, camera in hand, and the F/T sensor.

these two shapes.

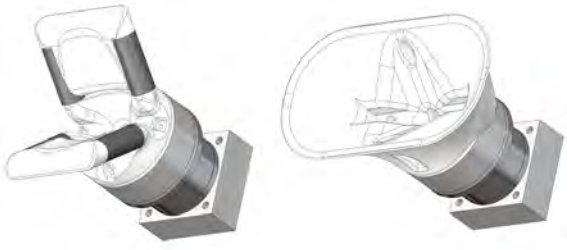


Figure 4.36: The shape on the left is more flexible and can have different contacts with the valve. The shape on the right is more closed but ensures an almost perfect grip.

Both shapes have been used successfully with the LbD algorithm but the most robust shape and with a bigger success rate has been the end-effector with the V-shape (image on the left of Figure 4.36). The open shape on the lateral allows grasping the T-shape valve from different positions. With this shape, although it is possible to incorrectly grasp the valve in the initial contact, is it possible to correct the grasping position with small adjustments. Moreover, to perform the valve turning correctly, it is not necessary to have the valve perfectly centred in the manipulator, it can be done from the sides ([Jamali et al.2015] explored different types of contacts). The V-shape also includes flexible parts to absorb small impacts and avoid damaging other difficult to be replaced or repaired parts.

4.5.2 Camera in-hand

At the initial stage, when the authors were exploring the tele-operation phase, we discovered that the configuration of the manipulator occludes the pilot's vision during the final part of the manipulation task. For the algorithm this is not an obstacle, since the positions are estimated according to the entire panel. However, for a human operator, this abstraction is more complex. For this reason, a small camera has been added in the centre of the end-effector. This camera allows visual feedback for the pilot during the manipulation, and

can also be used during the manipulation for the perception module to identify the panel and the state of the valves. Section 4.3.4 discuses the use of the in-hand camera in more details.

4.5.3 Force/Torque (F/T) sensor

To correctly perform the valve turning, it is necessary to perceive the contact with the valve and the necessary torque that must be applied in order to turn it (e.g., if the valve is locked, an excessive torque may break the handle of the valve or damage the manipulator). Since the manipulator can not give this information an F/T sensor has been installed in the end-effector.

The authors have encapsulated an ATI Mini45 F/T sensor that uses strain gauges to determine applied forces and torques in the 3 DOFs. To adapt the sensor to the underwater environment, the output signal generated is adjusted according to the depth of the end-effector, in order to compensate for the water pressure on the sensor. The measurements of the strain gauges drift over the time, even if stimulus are not applied. For this reason, the drift is compensated before performing a manipulation.

4.6 Combined control of GIRONA500 I-AUV

Several approaches to combine the control of a manipulator and a vehicle to perform intervention tasks have been designed and tested throughout this thesis. This section presents two methods to control the end-effector and the vehicle in a coupled form which means that the user or algorithm only provides requests to move the end-effector and the manipulator and the vehicle are moved to accomplish the requested command.

4.6.1 End-effector extended movements by means of AUV

This control schema was proposed at the beginning of this thesis. The manipulator was not installed in the vehicle, and the author's whole experience was from industrial manipulators and a simulated model. The idea of this control schema is to focus all the movement in the manipulator while the vehicle keeps a fixed position using the two systems as uncoupled elements. Due to the reduced workspace of the manipulator, a small range of positions is available. For this reason, the vehicle is moved when the requested pose is not in the workspace of the manipulator. Otherwise, the vehicle is kept in a fixed pose. The aim is to reduce movement of the less precise part of the system, the AUV.

This control scheme has two parts, as can be seen in Figure 4.37. First, the IK module is used to verify if the requested pose for the end-effector is reachable. If it is, the manipulator commands are generated in order to reach the joint configuration computed by the IK module while the AUV keeps its position. On the other hand, when the requested position

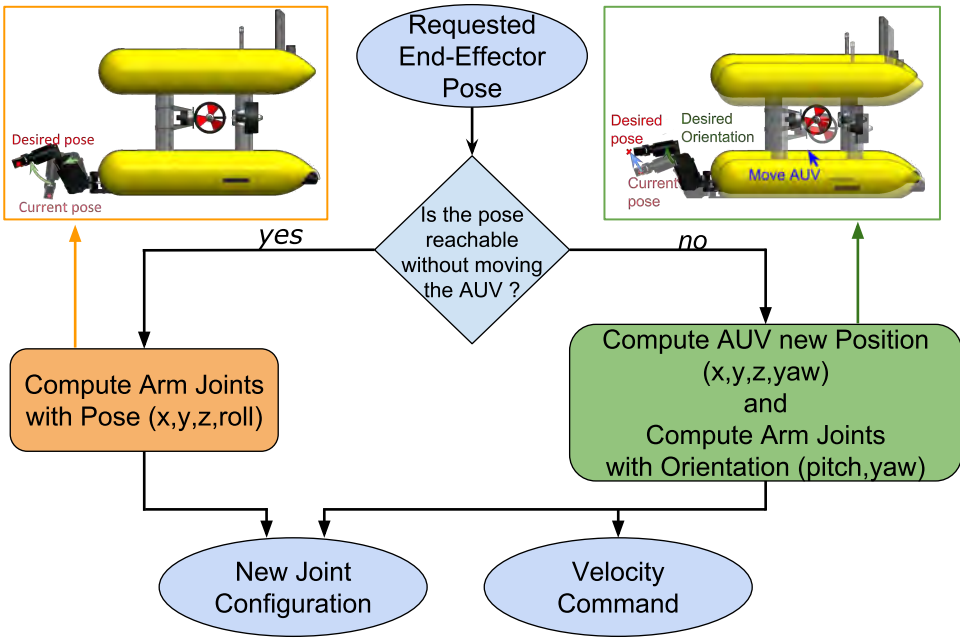


Figure 4.37: Action diagram of the end-effector's extended movements.

is not reachable, the movement is divided into two parts. First, the configuration of the manipulator is changed considering only the orientation and ignoring the position. Simultaneously, the controller computes the distance between the end-effector's new position and the desired position and generates the velocity commands to move the AUV, and consequently the end-effector, to this position.

A discussion of the results obtained using this control schema are shown in Section 5.3.

4.6.2 Underwater Vehicle-Manipulator System (UVMS)

The UVMS is a control scheme designed to control the position and orientation of the end-effector coordinating the movement of all the DOFs in the system at the same time, in this case the vehicle (4 DOFs) and the manipulator (4 DOFs). The scheme uses a priority policy to resolve the redundancies in the DOF. On the other hand, UVMS control scheme can not solve that the Vehicle-Manipulator system is still underactuated. However, the coordinated control mitigates the reduced workspace of the manipulator.

In other words, the UVMS defines the differential kinematics relationship between the end-effector velocities expressed in a fixed frame and the vehicle-fixed system velocity, composed of the manipulator and the AUV. The Jacobian matrix J represents this relationship.

Considering the UVMS Jacobian matrix $J \in \Re^{6\times 8}$ obtained as in [Antonelli2014] and the velocity in the configuration space $\dot{q} \in \Re^8$ (*surge*, *sway*, *heave*, *yaw*, *slew*, *elbow*,

elevation and *roll*), the end-effector velocity vector $x_E \in \Re^6$ is as follows:

$$\dot{x}_E = J\dot{q}. \tag{4.32}$$

The redundancy is handled in the configuration space using the following task priority control scheme:

$$\dot{q} = K^{-1}J_K^{\dagger}\dot{x}_E + K^{-1}(I - J_K^{\dagger}J_K)K_{\xi}\xi, \tag{4.33}$$

where $J_K^{\dagger}$, with $K \in \Re^{10\times 10}$ being a diagonal positive definite matrix, is the weighted pseudo-inverse of J (i.e., $J_K^{\dagger} = (JK^{-1})^{+}$) and $\xi = \nabla V(q)$ with $V(q)$ as an appropriately selected criterion function. In our case, we take into account the joint limitations of the manipulator constructing $V(q)$ as follows:

$$V(q) = \prod_{i \in I_{constraints}} (q^i - q^i_{min})(q^i_{max} - q^i). \tag{4.34}$$

A discussion of the results obtained using this control schema are showed in Section 5.5.

5

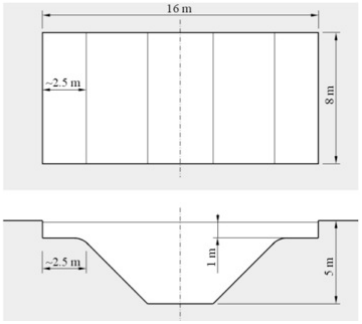
Experimental Results

Throughout this the thesis several experiments have been done in the context of a valve turning scenario introduced in Section 5.1. Experiments have followed an incremental evolution, starting from conceptual tests to final validation trials with multiple perturbations and running for several hours. Section 5.2 starts with a concept validation using an industrial manipulator to allow the author to familiarize himself with the LbD algorithm. Section 5.3 introduces the first results in a simulated environment. Then, the authors studied two possible controllers combined with the LbD algorithm, Section 5.4 introduces the results with an uncoupled controller while Section 5.5 shows the results of a coupled controller using a UVMS and compares them with the uncoupled one. Section 5.6 introduces the results obtained under different levels of perturbations and Section 5.7 presents a persistent experiment for a valve turning intervention performed in the context of the PANDORA project. Later, to improve the results under strong underwater currents, was proposed a parametric version of the algorithm, whose results are presented in Section 5.8. To conclude, a system to simplify the use of the proposed method, by using an algorithm to autonomously tune the DMP parameters, has been studied in Section 5.9.

5.1 Description of common elements

All the experiment in this thesis, except the test with the industrial manipulator (Section 5.2) and the simulated tests (Section 5.3), have been performed in the underwater tank of CIRS which has a dimension of 16 x 8 x 5 meters, see Figure5.1.

(a) Picture of the water tank and the laboratory of the CIRS



(b) Schematic representation of the water tank dimensions.

Figure 5.1: Images of the water tank installations.

The mock-up valve panel built for the PANDORA project and used in all the experiments in this thesis has a size of 0.8 x 0.5 m and is composed of 4 valves (Figure 5.2a). The 4 valves are T-shaped, and measure of 0.11 x 0.1 m in size with a thickness of 0.01 m (Figure 5.2b).

(a) Picture of the mock-up of the sub-sea panel, with 4 valves

(b) Picture of the mock-up valve used in the experiments with the areial manipulator.

Figure 5.2: Images of the mock-up elements developed to represent the sub-sea valve panel.

The visual detection system presented in Section 4.3.4 assigns the panel axis, as can be seen in Figure 5.3. The perpendicular distance with respect to the panel is represented in the z axis, while the horizontal and vertical axis are x and y respectively. The valves use similar axes, z is the perpendicular distance to the valve, y follows the handle of the valve, and x is perpendicular to the handle. With this configuration, when the valve is grasped, the *roll* of the end-effector is zero. It is worth mentioning that the robotics community establishes the orientation of an end-effector differently from the orientation of the vehicle. Hence, the *roll* is assigned to the rotation over z instead of x, the *pitch* is assigned to y

and the *yaw* is assigned to x instead of y.

Figure 5.3: Image of the sub-sea panel with the frame of the panel center overlayed. The x, y, and z of the axis are depicted in red, green and blue.

5.1.1 Common elements for the DMP

In all the experiments, the learning has been done at the trajectory level and the demonstration method has been tele-operation. The only exception has been the demonstration to the industrial manipulator, where the kinesthetic guiding method, with the arm in a gravity compensation mode, has been used. Demonstrations have been presented to the algorithm as a batch of demonstrations.

In all trajectories learned, we have used the target valve as the center of the learning frame. In this way, all the trajectories tend to reach position (0,0,0). However, the orientation used has been fixed to the orientation of the valve panel, the valve frame has only been used for the *roll* of the end-effector, to align the valve and the end-effector correctly. Moreover, all the learned models only include the trajectories, in each DOF, until the valve is grasped. These two characteristics allow us to reuse the same model learned for all the valves in the panel and for different valve orientations and turns.

The DMP algorithm presented in Section 3.2.1 has been used in the experiments. Four different parameters have to be adjusted in order to adapt the DMP method to different tasks, represented by a trajectory. The first parameter is the α, which, in this case, is always set to the fixed value of 1.0, meaning that the reproduction has to be performed in the same time as the demonstrations. The next parameter is the the number of Gaussians used to define the trajectory. A high number of Gaussians will allow the representation of movements with many restrictions and low flexibility while a small number is better to represent trajectories with few restrictions and more variability. The duration of the demonstrations also influences the number of Gaussians needed, since they are distributed uniformly along the trajectory. So these values will be different for each trajectory case. The other two parameters to be set are K^P_{min} and K^p_{max}. These values define the initial value to search for the stiffness (K^P) and the damping of the system (K^V) associated with each Gaussian.

These two parameters are highly related with the element to be controlled and the kind

of control to be used. These parameters define the limits of the commands generated to reach the desired position. In our case, these parameters have been modified and adjusted depending on the environmental conditions. The goal is to find the minimum values that allow for a smooth and stable trajectory. However, when we introduce perturbations in the environment, small K^P and K^V can excessively limit the velocities and thus prevent the vehicle from keeping the desired trajectory. Under this scenario, we can set K^P_{min} and K^P_{max}, to expand to a larger range and therefore obtain bigger K^P and K^V values that will minimize the restriction on the velocities and therefore handle the perturbations better. Hence, it is clear that the value of these parameters is always a trade-off between the vehicle's stability along the trajectory and its capability of overcoming perturbations.

5.2 Industrial Manipulation

The initial LbD experiments of the thesis, the author used an industrial manipulator and a visual tracking system (see Figure 5.4) to get used to the DMP algorithm in a simpler scenario. The purpose of this experiment was to validate the use of the DMP for the underwater valve turning scenario and test the combination of the DMP with a reactive system.

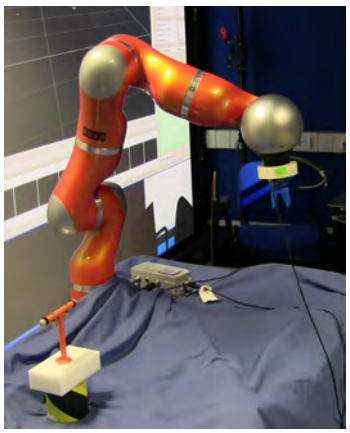


Figure 5.4: Image of the IIT Facilities with the Optitrack system, the KUKA 7 DOFs manipulator and a mock-up of an underwater valve.

5.2.1 Experimental Set-up

To simulate the underwater valve turning task in a simpler laboratory environment, two elements were used:

- KUKA/DLR manipulator, a light weight robotic arm with 7 DOFs designed to work in the same space as humans operators. The arm disposes of compliant control modes, where a human can interact with the movement of the arm safely.

- Optitrack visual system, a high precision position system, which uses a set of cameras and markers and is able to track the position of rigid bodies in an area.

The end-effector position and the valve position were tracked using the Optitrack system. However, this data was processed before being used by the LbD algorithm, to simulate a more realistic environment. The information was transformed to simulate two sensors, the valve position detected by a visual algorithm and the vehicle's attitude generated by an AHRS sensor in the AUV. These two data streams were filtered using an EKF simulating a more realistic set up in a real AUV.

On the other hand, the reactive system was evaluating the safety of the intervention using simulated data, the time without a camera update, and the stability in the movements of the AUV provided by the AHRS sensors.

A version of the DMP controlling only 3 DOFs was used to learn the trajectory defined by the end-effector to grasping the valve. The trajectory was recorded using the valve position as a reference frame. The DMP used the modification of the canonical system explained in Section 3.4.1, to allow the modification of the behaviour by the reactive system. Figure 5.5 shows the information flow through the system's elements.

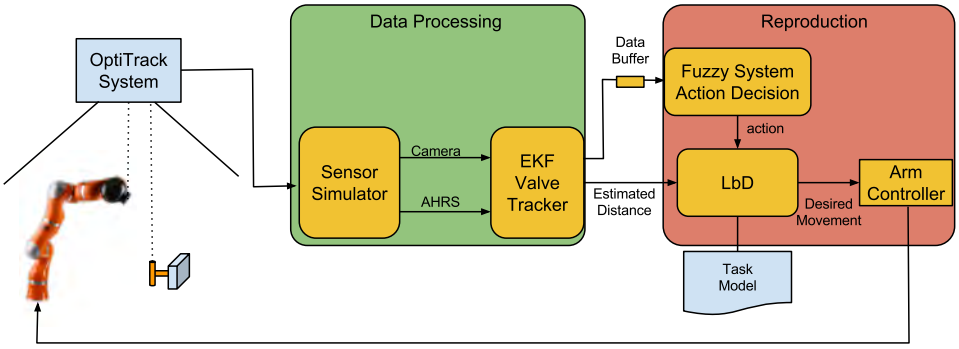


Figure 5.5: This diagram represents the data flow from the Optitrack to the different phases of the experiment to finally convert commands to the robot.

5.2.2 Results

To demonstrate the task, 8 demonstrations have been performed with the robotic arm in gravity compensation mode. In this mode, the human operator can easily move the manipulator to the desired positions using the hands (see Figure 5.6). In this experiment, the operator moved the end-effector from several initial positions until the valve was grasped.

Due to the reduced number of DOFs learned for this experiment and the facility of understanding the trajectory, this data is also used to analyse how to define the number of Gaussians necessary to configure the DMP. Figure 5.7 shows the eight trajectories in

Figure 5.6: Human operator performing a demonstration of the robotic arm in gravity compensation mode.

the XY plane, and the model learned using 2, 3, 4, and 5 Gaussians. In the figures, you can see that the model with 4 and 5 Gaussians only provides redundant information, so, for this case, the more complete solution is the one with 3 Gaussians and an acceptable one, but not as accurate, could be the one with only 2 Gaussians. The values for k^P_{min} and K^P_{max} are defined at 0 and 20.

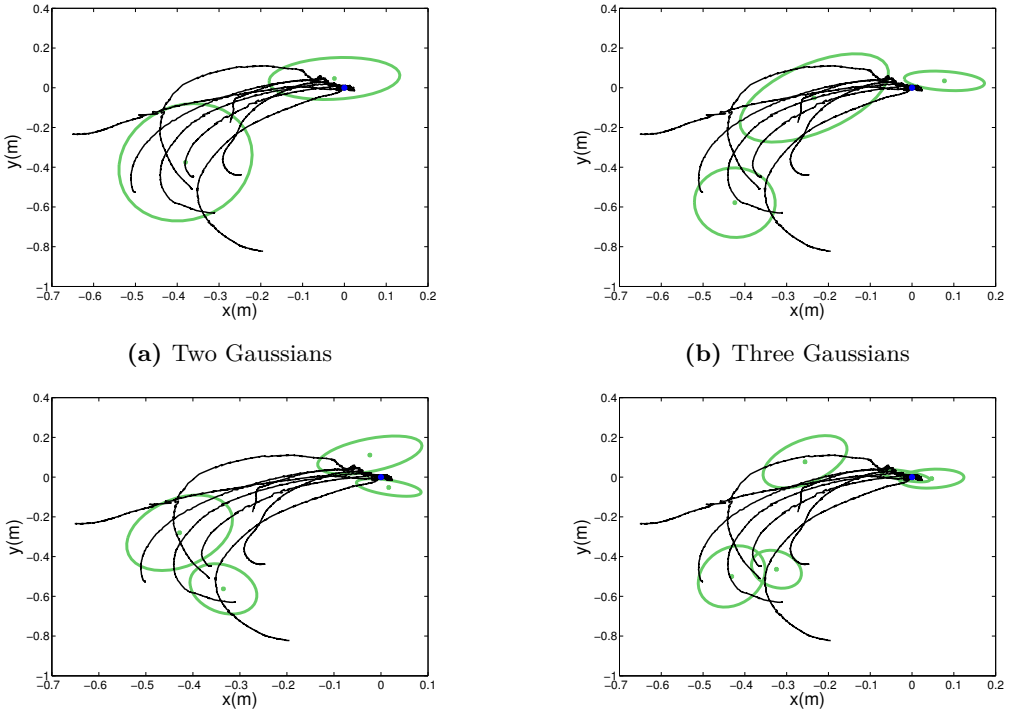
To validate the proposed set-up, we performed two sets of experiments. One without perturbations to see if the DMP was able to grasp the valve correctly and another using the reactive system to evaluate the perturbations introduced by an external element. Two different perturbations were included: the movement of the valve, representing an instability on the simulated AUV, and pushing the manipulator to move the end-effector into unwanted positions.

The system successfully completed the task in both situations as can be seen in Figure 5.8. When there are no perturbations, the end effector advances following a smooth path (see Figure 5.8a). When there are perturbations, the trajectory is unstable and the end effector stops and goes backward and forward according to the state situation (see Figure 5.8b). Figure 5.9 shows how the time line goes backward on three occasions during the experiment due to the perturbations in the system.

5.2.3 Discussion

The satisfactory results obtained from these experiments encouraged the authors to continue using the proposed DMP algorithm throughout the thesis. Moreover, these experiments helped to detect relevant key-points for the next experiments.

First, the definition of the target valve as a frame center for the learned trajectory. This approach allows us to reuse the same model in different positions. Second, the position



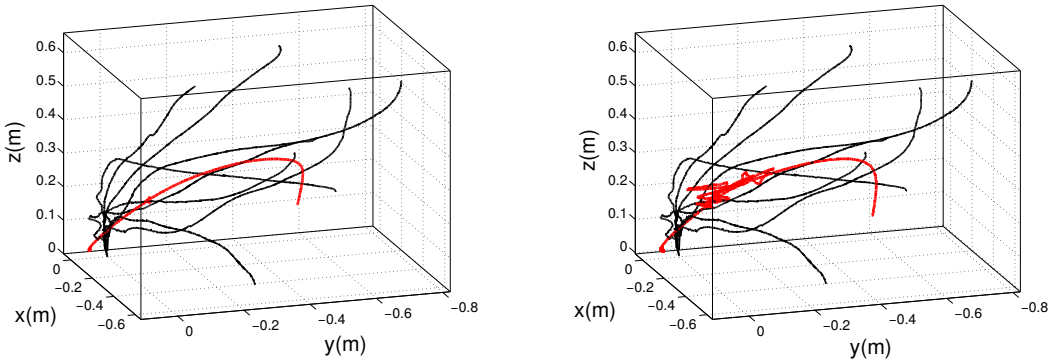
(a) Two Gaussians

(b) Three Gaussians

(c) Four Gaussians

(d) Five Gaussians

Figure 5.7: The four figures show the set of 8 demonstrated trajectories the model learned represented by the Gaussians.



(a) Autonomous trajectory without external perturbations

(b) Autonomous trajectory with external perturbations

Figure 5.8: In these two figures the 8 demonstrations (black) and one autonomous reproduction (red) are depicted. All the trajectories are represented in the frame of the target valve. The plot is made using a 3D representation of the trajectory performed by the end-effector.

of the valve needs to be filtered to avoid instability in the trajectory of the end-effector. For this reason, the valve panel position estimation provided by the visual detector were included in the EKF-SLAM filter in the GIRONA 500 AUV localization module. Third, the need to use external modules to evaluate the safety of the intervention and modify the

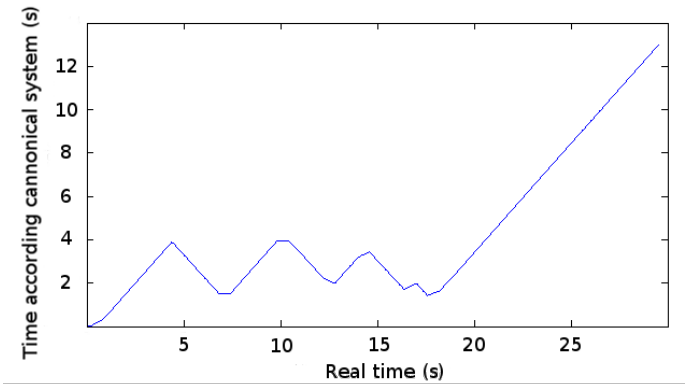


Figure 5.9: This graph shows how the time generated by the canonical system advances in the real time of the experiment.

performance of the trajectory was required to obtain a robust system.

5.3 Coupled control by extended movements of the end-effector

While the integration of the ECA Micro 5E arm in the GIRONA 500 AUV was still in progress, a simulated version of the manipulator was developed in the UWSim [Prats et al.2012b] simulator (see Figure 5.10). The purpose of this experiment was to test the inclusion of a combined control, explained in Section 4.6.1, to resolve the valve turning task using a DMP algorithm which learns 6 DOFs. The simulation was carried out under two situations, in calm conditions without perturbations and under small perturbations applied to the position of the AUV.

Figure 5.10: Image of the UWSim where the simulated GIRONA 500 I-AUV is performing a valve turning.

A new scenario was developed in the UWSim including all the elements necessary to test new developments for the underwater valve turning. An accurated 3D representation of the panel was created to test the visual detection algorithm with a virtual camera and manipulation. Also a 3D representation of the manipulator in order to test all the software

developed to control it and test the behaviours developed.

To test the control scheme for the end-effector with extended movements, the GIRONA 500 I-AUV is situated in a position where the manipulator is able to grasp the valve without moving. This position can be defined experimentally or mathematically, using the workspace of the manipulator. From this position, the operator controls the end-effector moving the manipulator from a folded configuration, used during the navigation of the vehicle, to the position where the valve is grasped. This movement goes from the right side of the AUV to a perpendicular position respect the valve panel and aligned with the valve. To illustrate this movement, a sequence of of images taken during a demonstration are shown in Figure 5.11.

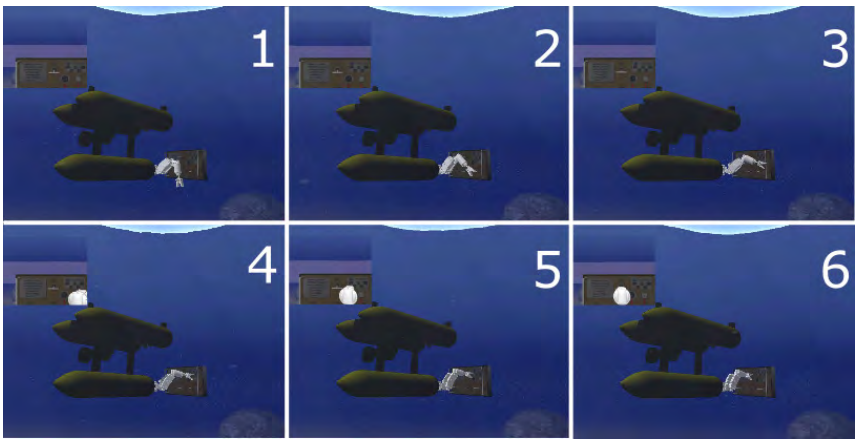


Figure 5.11: Sequence of images recorded during the manipulation of a simulated valve without perturbations.

5.3.1 Results

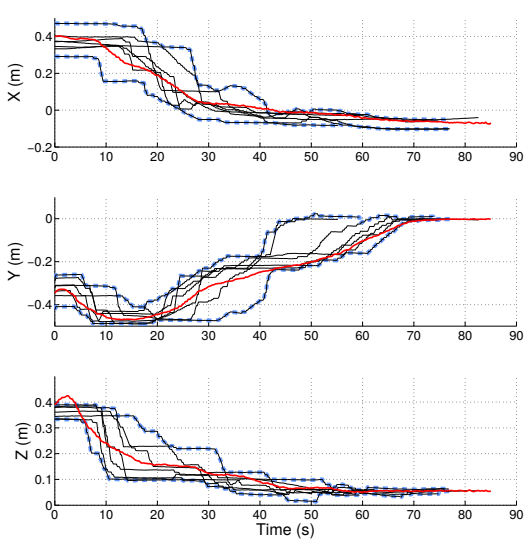
To demonstrate the task, 7 demonstrations have been recorded, and to generate the learning model, 30 Gaussians and a K^P_{min} equal to 2 and K^P_{max} 10 have been used. The elevated number of Gaussians was required in order to be able to follow the trajectory with a high accuracy, due to the limitation of the workspace, avoiding any unreachable positions, which would imply an undesired movement of the AUV. The trajectory learned is defined by 6 DOFs representing the position and orientation of the end-effector.

The reproduction of the task has been performed under two different conditions, without perturbations and with small perturbations applied randomly. The perturbations cause a displacement of the AUV's position, making impossible for the end-effector to reach the valve without changing the vehicle's position. This situation tested the combined control of the end-effector and the vehicle.

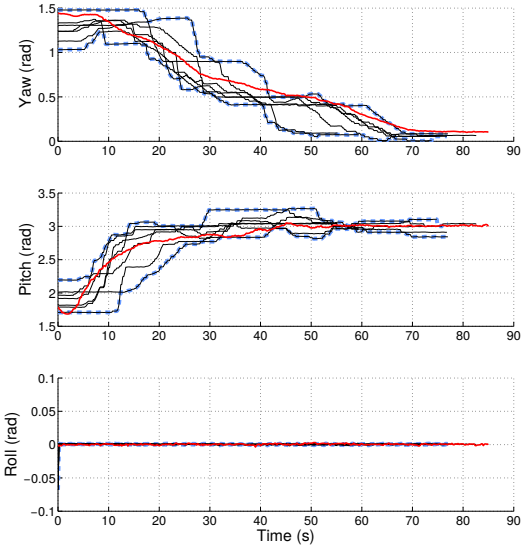
Figure 5.12 and 5.13 show the results without perturbations and under perturbations. The reference frame in the trajectory is the target valve to be manipulated, for this reason,

all the trajectories finish close to zero. In both conditions, the trajectory is smoother than in the demonstrations.

In the result without perturbations, the trajectory is kept inside the limits of the demonstrations, except with the values for *pitch* and *yaw*, which, in some parts, are outside or close to the limits of the demonstrations. These DOFs are not controllable at the same time as the x, y, z, and *roll*. However, the limited workspace in the manipulator only makes it possible to reach the x, y, and z positions with a small range of *pitch* and *yaw* values. Hence, even if the *pitch* and *yaw* of the end-effector are not controlled, they are followed with acceptable values.



(a) Positions of the end-effector



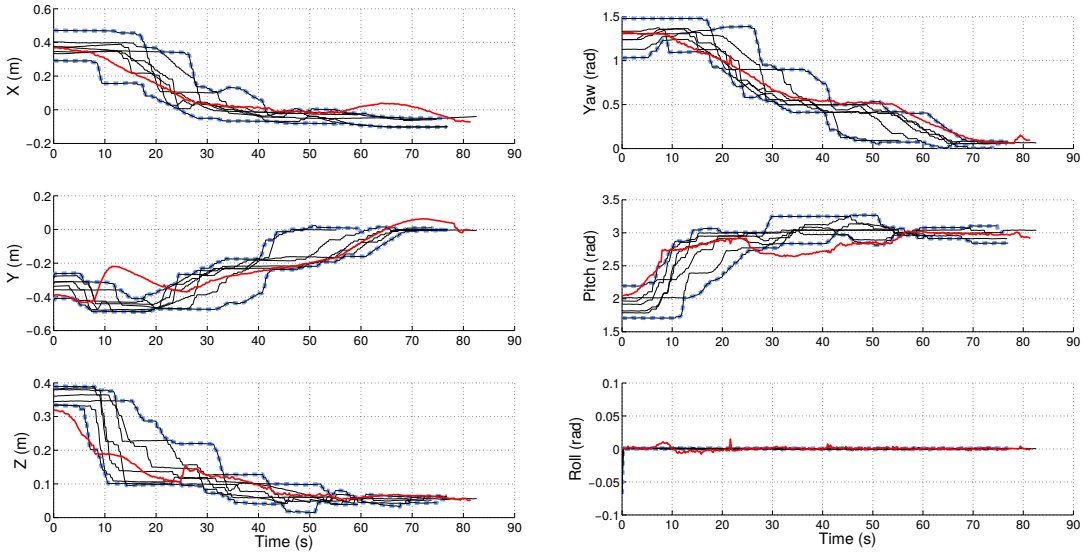
(b) Orientations of the end-effector

Figure 5.12: Seven demonstrated trajectories (black) for the valve grasping task. The upper and lower limit of all the demonstrations is depicted in dashed-blue and an autonomous reproduction is depicted in red. All the trajectories are represented in the frame of the target valve. Each plot shows a single DOF of the end-effector.

During the reproduction under perturbations, the system received two perturbations; one strong disturbance between 8 and 12 seconds affecting the y and z axis, and a smaller one between 55 and 65 seconds affecting the y and x. In both cases, the reaction of the controller in the *pitch* angle can be seen. For example, after the first perturbation, the position in y returns progressively to the limits and the *yaw* and *pitch* orientations are kept inside the limits. Then, when the manipulator is moved again in position and *roll*, the *yaw* is stopped so as to be tracked correctly (from the second 25 to 45).

5.3.2 Discussion

The coupled control of the end-effector by extended movement of the AUV obtained good results in a simulated environment even under punctual perturbations. However, the



(a) Positions of the end-effector

(b) Orientations of the end-effector

Figure 5.13: Seven demonstrated trajectories (black) for the valve grasping task. The upper and lower limit of all the demonstrations is depicted in dashed-blue and an autonomous reproduction is depicted in red. All the trajectories are represented in the frame of the target valve. Each plot shows a single DOF of the end-effector.

transition of this method to the GIRONA 500 I-AUV showed some errors in the proposed concept. The first issue was found in the differences between the control of the vehicle. In a real environment, reaching and keeping the position in which the manipulator is able to grasp the valve is very difficult. The second issue was tele-operation of the end-effector. Understanding which system will be moved when a command is sent and being aware of the limits of the manipulator, is a difficult task and not intuitive. This can cause errors in relevant movements of the manipulation, e.g., when the valve is about to be grasped, the manipulator is close to its limits. This can cause a change between controlling the end-effector and the vehicle, thus causing a small impact or possibly losing the grasp. The third issue was to control the 6 DOFs of the end-effector. Understanding when the orientation or the position have to be controlled to correct the pitch and yaw is tricky and requires a trained skill and a proper human-robot interface. The fourth issue was the appearance of chattering phenomena, especially when the manipulator was close to the limits and the vehicle was in an unstable situation due to the environment conditions. A temporary solution was developed, which was reducing the workspace of the manipulator when the control scheme was in the AUV manipulator control. It allowed the manipulator to reach a non-limit configuration before returning to the only manipulator mode.

To improve control of the vehicle and end-effector, the authors proposed two different control schemes. An uncoupled control, where it is only possible to control the available DOFs; the vehicle's 4 DOFs and the end-effector's 4 DOFs. The results can be seen in Section 5.4. Alternatively, the authors also developed a more sophisticated coupled control

using a UVMS scheme, the results of which are shown in Section 5.5.

On the other hand, to improve the human-robot interface, instead of using a regular gamepad, a Haptic device of 7 DOF was integrated to tele-operate the vehicle and manipulator using the coupled and uncoupled control. Moreover, the haptic device has force feedback which was used to provide information about the limits and contact of the end-effector with the target valve.

5.4 Uncoupled controller

The ECA Micro 5E manipulator was integrated successfully in the GIRONA 500 AUV, however, the customized end-effector presented in Section 4.5 was still under development. For this reason, a provisional end-effector was used. The provisional end-effector was made by foam and carried only a camera in the hand, as can be seen in Figure 5.14.

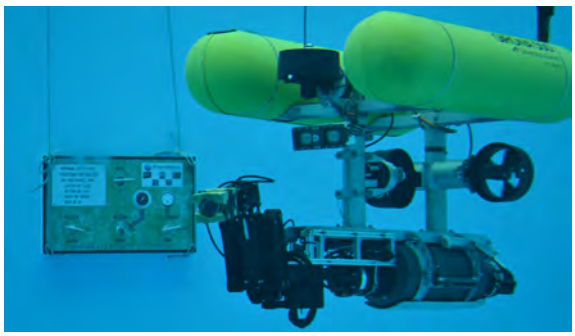


Figure 5.14: Image of the GIRONA 500 I-AUV with the manipulator and the provisional end-effector installed.

The Uncoupled control DMP algorithm learns a total of 8 DOFs composed of x, y, z, and yaw for the vehicle and x ,y, z, and $roll$ for the end-effector (Figure 5.15). The learned trajectory uses the target valve as the reference for all the whole trajectory. To resolve redundancy in the x, y, and z between the manipulator and the vehicle, the velocity commands sent to the end-effector subtracted the velocity commands sent to the vehicle.

The valve turning has been performed using two different strategies:

- **Close Manipulation**: The GIRONA 500 I-AUV is in-front of the sub-sea panel at a suitable inspection distance (between 1.75 and 2.0 meters). This pose is kept stable while the manipulator is moved from a compact configuration to a position appropriate for the manipulation of the grasp. Next, the vehicle and the manipulator are moved to grasp the valve. The AUV performs the bigger changes and the manipulator corrects the small errors in the movement of the AUV.

- **Far Manipulation**: The GIRONA 500 I-AUV is far from the sub-sea panel, at a distance where the sub-sea panel can start to be identified (between 4 and 5 meters).

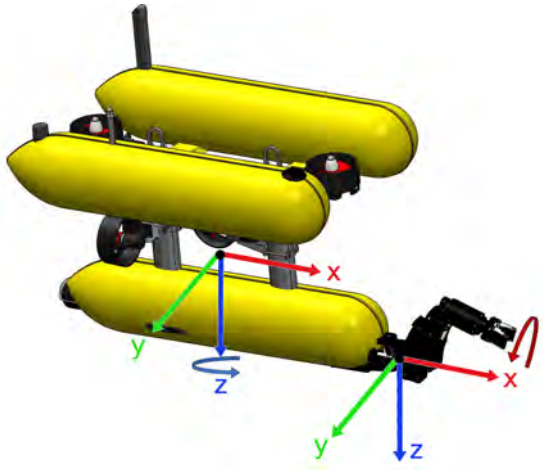


Figure 5.15: 3D model of the GIRONA 500 I-AUV with the actuable DOFs overlaid.

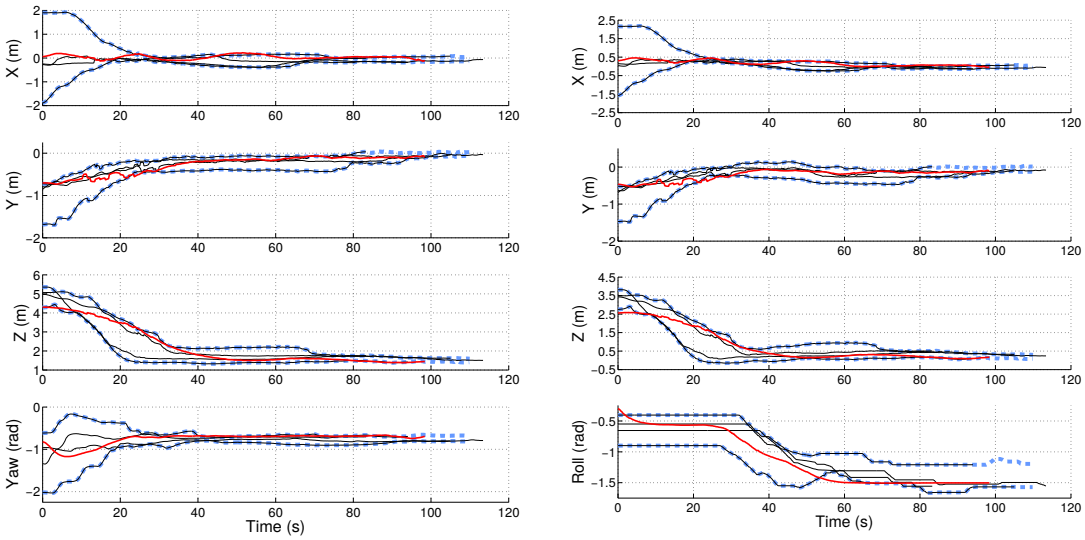
The vehicle is moved to an inspection distance (between 1.75 and 2.0 meters). When the position is reached and the valve status is identified, the *close manipulation* procedure starts as explained before.

5.4.1 Results

This section shows the results obtained in a *far manipulation*. The results obtained in the *close manipulation* are shown together with the UVMS in Section 5.5 with the UVMS. Moreover, the other experiments performed in this thesis have focused on *close manipulation*, due to the greater relevance of the LbD algorithm.

To teach the trajectory, 4 demonstration and 16 Gaussians have been used , K^P_{min} equal to 1 and K^P_{max} equal to 7. Figure 5.16 shows the trajectories reproduced using the valve as a frame of reference. In this case, only the learned DOFs have been plotted. It can be appreciated that the trajectory of the end-effector seems almost equal to the trajectory performed by the end-effector, due to the difference in scale between the AUV's movements and the end-effector's. Figure 5.17 shows the trajectory performed using the base of the arm as a frame of reference.

As aforementioned in the *far manipulation*, the vehicle is moved to the inspection position (from 0 s to 25 s), with the arm folded on the right side of the vehicle. Next, the manipulator is adjusted to a proper configuration to grasp the valve (from 25 s to 70 s), also from this position the *roll* of the end-effector is aligned with the valve's orientation. In the final part (from 75 s to 95 s), the vehicle is slowly moved forward to grasp the valve, in the plotted demonstration the position was stable, which means the end-effector didn't have to correct any errors introduced by the vehicle in this position.



(a) AUV positions and orientations

(b) End-effector positions and orientations

Figure 5.16: Four demonstrated trajectories (black) for the valve grasping task using *far manipulation* strategy. The upper and lower limit of all the demonstrations is depicted in dashed-blue and an autonomous reproduction is depicted in red. All the trajectories are represented in the frame of the target valve. Each plot shows a single DOF of the AUV (left) or end-effector (right).

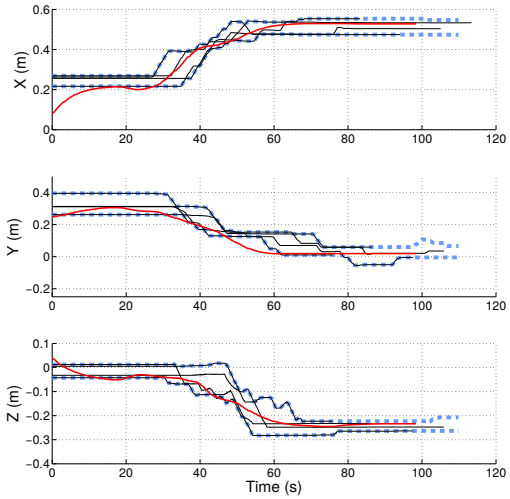


Figure 5.17: Four demonstrated trajectories (black) for the valve grasping task using *far manipulation* strategy. The upper and lower limit of all the demonstrations is depicted in dashed-blue and an autonomous reproduction is depicted in red. All the trajectories are represented in the frame of base of the arm. Each plot shows a single DOF of the end-effector.

5.4.2 Discussion

This approach has allowed the operation of the vehicle and manipulator independently, supporting the demonstration performed by two operators. Also, this uncoupled learning facilitates the learning of tasks where the vehicle is moved during a part of the task with

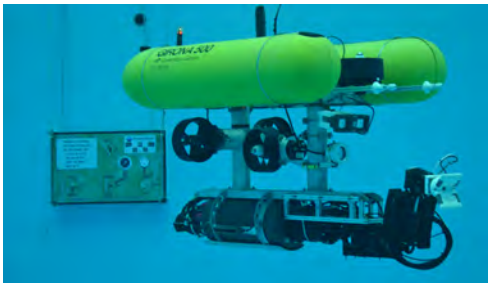
the manipulator in a fixed configuration. Hence, the learned models can be more complete including displacements within the manipulation.

The grouping of the controlled DOFs passing from 6 DOFs to 4 with the 4 DOFs facilitating the control. The tele-operation can be performed by two operators where one controls the vehicle and the other the manipulator, or one operator switching from one system to the other.

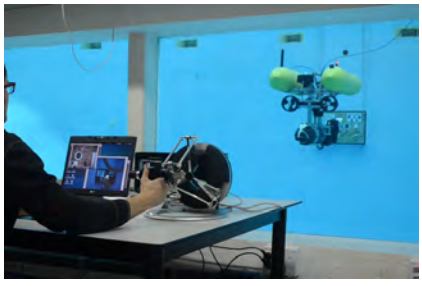
However, these kinds of trajectories with movements in diverse scales make the adjustments of the DMP parameters difficult. For this reason, the results presented selected parameters to obtain a smooth trajectory but not too accurate in some instances where the end-effector goes outside the demonstration area. This less constrained parameters appear to be better, but due to the reduced workspace of the end-effector, this movement can bring the end-effector to a limit of the manipulator, preventing a successful manipulation. For this reason, the number of Gaussians of the DMP need to be adjusted according to the complexity of the task to try to avoid unexplored areas in the workspace.

5.5 Coupled control using UVMS controller

The GIRONA 500 I-AUV was equipped with the custom end-effector presented in Section 4.5, and can be seen in Figure 5.18a. Furthermore, to improve the tele-operation and facilitate control of the end-effector's 6 DOFs at the same time a Force Dimension Omega 7 haptic device was used to control the end-effector (see Figure 5.18b). This device provides the possibility of controling 7 DOFs using only one hand, and has force feedback connected to the F/T sensor.



(a) Image of the GIRONA 500 I-AUV with the custom end-effector.



(b) Tele-operation of the GIRONA 500 I-AUV using the haptic device.

Figure 5.18: The two images show the experimental set-up for the experiments with the UVMS coupled controller.

A *close manipulation* strategy was followed to teach and perform the valve turning task with the I-AUV. The vehicle was at an inspection distance(2-3 meters) from the sub-sea panel and the manipulator was folded in-front of the vehicle, not at the side. In this case, the operator sent commands of position and orientation to the end-effector.

To evaluate the result of the UVMS, the same demonstration done with the uncoupled

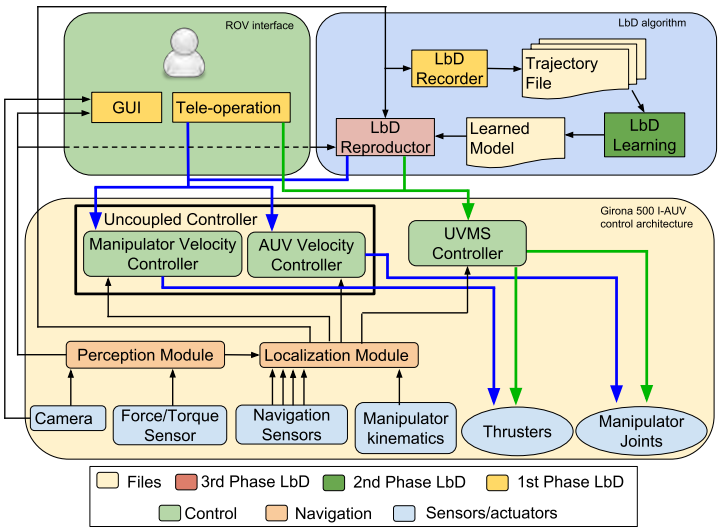


Figure 5.19: Diagram of the framework with the two different control schemes. The blue arrows represent the flow of the uncoupled control scheme and the green arrows the flow of information of the UVMS scheme.

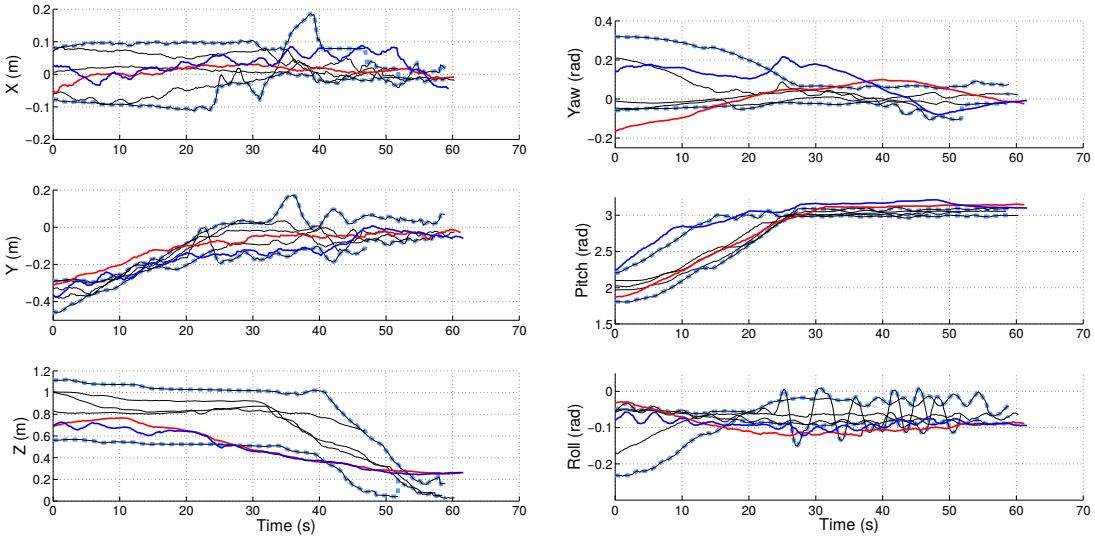
controller presented in the previous section has been used. Figure 5.19 shows the intervention framework presented in Section 3.3 with the flow of information with the coupled and uncoupled controller.

5.5.1 Results

Figure 5.20 shows the results obtained using the coupled (in red) and uncoupled (in blue) controllers. A batch of 4 demonstration has been made using the UVMS controller. The number of Gaussians used was 12 and the K^V and K^P where 0.5 and 5.5, for both controllers. The plots of this experiment only show the trajectory of the end-effector to the target valve since the vehicle is not controlled by the UVMS directly.

By comparing both reproductions, you can appreciate that the UVMS performed a smoother and more stable trajectory than the uncoupled controller. This is specially visible in the x and y DOFs. The *pitch* and *yaw* orientations were followed more precisely in the UVMS than in the uncoupled controller, due to the uncoupled controller did not learn those DOFs. Finally, the z DOF had an error between the demonstration phase and the reproduction phase. The valve was in a closer position than in the demonstration and for this reason the trajectory stopped in both cases at 0.3 m instead of almost zero.

The UVMS controller prioritizes the movement of the manipulator over the movements of the the vehicle when there is there is a redundancy in the command and the manipulator is not in the limits. On the other hand, the uncoupled controller is constantly controlling the AUV and the manipulator, sending commands to both systems. Therefore, as a consequence of more vehicle movement, the system produces a less stable trajectory.



(a) End-effector trajectory

(b) End-effector orientation

Figure 5.20: Five demonstrated trajectories (black) for the valve grasping task. The upper and lower limit of all the demonstrations is depicted in dashed-blue, an autonomous reproduction performed by the UVMS depicted in red and an autonomous reproduction performed by the uncoupled controller in blue. All the trajectories are represented in the frame of the base of the arm. Each plot shows a single DOF of the end-effector.

5.5.2 Discussion

The results of the UVMS in this experiment were smoother than the uncoupled controller, however the author decided to continue using the uncoupled controller due to the following arguments.

The tele-operation of the system becomes more difficult. For the operator, it is difficult to see if the end-effector can be moved, using only a change in position or if a position and orientation change is required to move only the end-effector in the desired direction. This error causes involuntary movements of the vehicle and the operator reaches the requested position with an inappropriate orientation.

On the other hand, the UVMS coupled control limits the use of the same system for multiple purposes. For example, the performance of the *far manipulation* using the UVMS will require an extension of the method to be able to change the priorities of movement according to the part of the demonstration. In the first part of the demonstration, the vehicle needs to have priority and in the second part, the manipulator needs to have priority.

It is worth mentioning that the parameters used in both learnings are equal, and where adjusted using the UVMS scheme as a reference, it could be possible to obtain better results in the uncoupled control if the parameters were adjusted differently.

Finally, the authors are aware that the UVMS can be extended and improved increasing the number of secondary priority tasks, adding more criterions (in our case, has been

defined a criterion related to the joint limits). This allows the tailoring of the control for a particular task and reduces the flexibility of the system. However, the goal of this thesis is to obtain the most flexible system, where the end-user can program new tasks without requiring any programming or mathematical knowledge. The use a more perfectioned UVMS will reduce this possibility.

5.6 Uncoupled control under the effect of underwater currents

The GIRONA 500 I-AUV was controlled using the uncoupled controller and performed the valve turning task following the two strategies *far manipulation* and *close manipulation* under different degrees of perturbations. However, the perturbation system only effected the vehicle when it was close to the sub-sea panel. For this reason, the experiments presented are focused on the valve turning task in *close manipulation*. Figure 5.21 shows the vehicle, the sub-sea panel and the perturbations system installed in the water tank. This system is composed of two thrusters placed next to the sub-sea panel and can produce a force up to 20 kg.

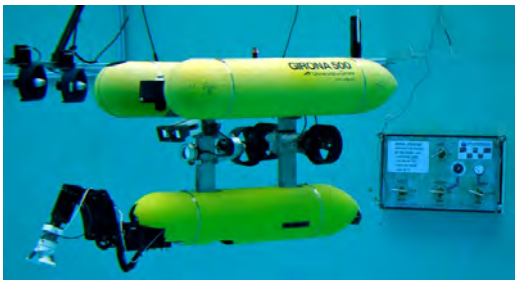


Figure 5.21: GIRONA 500 I-AUV in a water tank with a mock-up of a sub-sea panel in the background and two Sea-eye thrusters (on the left) used to introduce perturbations during the manipulation

The Intervention framework (see Figure 5.22) was extended in order to integrate the end-effector's F/T sensor. Contact with the valve was detected using this sensor and the LbD algorithm was stopped. Moreover, a simple procedure was added to perform a constant push against the valve and to control the torque in case it was blocked and if so, cancelling the task.

On the other hand, to improve the DMP algorithm, the different DOFs have been grouped according to the type of movements and characteristics of the controller. For example, the x and y DOFs have been grouped together since their range of movement was similar. Also, the vehicle's DOFs and the end-effector have been separated in order to define the K^V and K^P parameters more precisely. These are in charge of defining the relationship between the model and the real system.

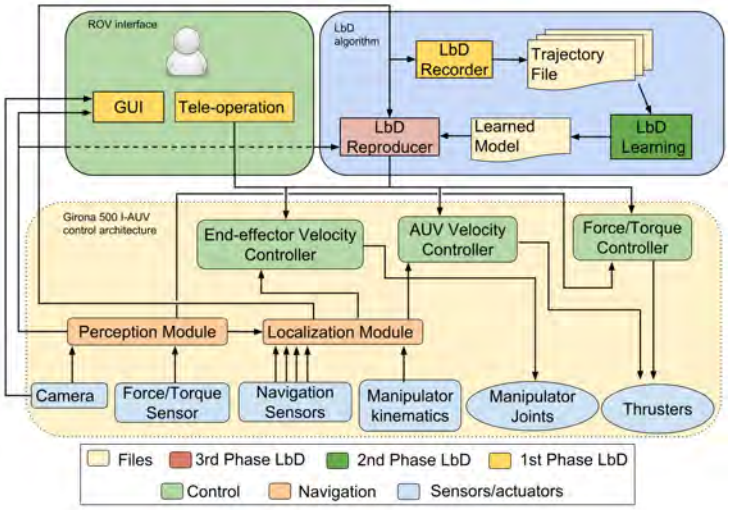


Figure 5.22: Extended intervention framework to include the F/T controller.

5.6.1 Results

During the experiment, the valve turning task has been repeated more than 60 times. In two thirds of these tests, we decided to introduce different levels of water currents to test the system's resilience to perturbations. The water currents have been introduced by means of two thrusters, which can produce up to 20 Kg of force, placed next to the submerged panel. The LbD system must deal with these perturbations and generate higher velocity requests to compensate for the external force while keeping the necessary precision to perform the valve turning. Water currents were increased systematically during the whole experiment, reaching 70% of the power (14 Kg). At this point, the vehicle was not able to keep a perpendicular position in front of the panel, making it impossible to execute the valve turning task as the manipulator could not adapt its configuration to reach the valve handle. It is worth noting that in our set-up, the water current affects the vehicle laterally and the only thruster the AUV has to compensate in this DOF is rendered insufficient to cope with such an amount of perturbation. Under slightly lower perturbations, the vehicle was not able to keep the perpendicularity with respect to the panel either, however, the DMP algorithm was able to find a trajectory for the end-effector from which to reach the valve and accomplish the task.

The parameters to configure the DMP algorithm are specified in Table 5.1. The DOFs have been defined as x, y, and yaw of the AUV, the z of the AUV, the x, y, and $roll$ of the end-effector, and the z of the end-effector.

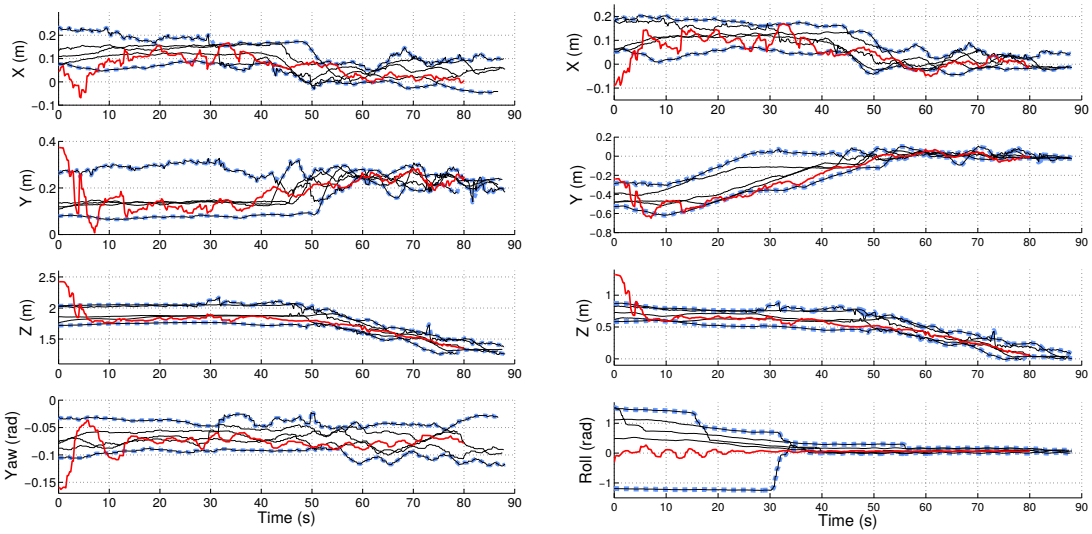
The results of these experiments have been divided between two different figures. Figure 5.23 shows the demonstrations and one reproduction with no perturbations, and Figure 5.25 shows one autonomous reproduction under a different group of perturbations.

		Parameters for the DMP		
		Gaussians	K^P_{min}	K^P_{min}
AUV	x, y, yaw	15	0.5	2.5
AUV	z	11	0.1	1.0
End-effector	x, y, $roll$	15	1.0	3.0
End-effector	z	11	2.0	3.0

Table 5.1: Groups and parameters defined for the learning under the effect of underwater currents.

The perturbations have been grouped in five intervals: no-perturbations [0-20)%, low-perturbations [20-35)%, medium-perturbations [35-50)%, high-perturbations [50-70)% and maximum-perturbations [70-100]%.

Figure 5.23 validates the chosen approach since it is possible to follow the trajectories correctly even starting in a different position than the one in the demonstrations. In the figure, you can appreciate that the x, y and yaw are less stable than in other experiments. This is caused by the larger distance between parameters K^P_{min} and K^P_{max}. This difference implies a greater reaction capacity to perturbations but reduces the smoothness and stability.



(a) AUV positions and orientations

(b) End-effector positions and orientations

Figure 5.23: Demonstrated trajectories (black) for the valve grasping task. The upper and lower limit of all the demonstrations is depicted in dashed-blue and an autonomous reproduction is depicted in red. All the trajectories are represented in the frame of the target valve. Each plot shows a single DOF for the manipulator and the end-effector.

Figure 5.24 shows the force and torque during the valve turning. The graph is divided into four phases defined according to the action performed. In the first phase (yellow),

contact with the valve was detected and the learning was stopped. Next (orange), the vehicle pushed and got constant contact with 15 N to prepare the turning. Then it kept contact with the valve and turned (red area) it with a limit of 0.5 Nm. During the initial part of the turning, there were small changes on contact due to a slight correction of the grasping caused by the turning. Finally (green), the vehicle moved backward, finishing the valve turning.

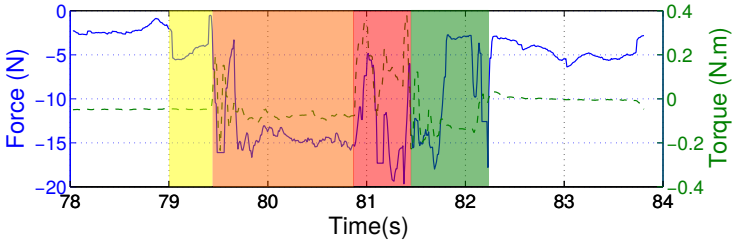
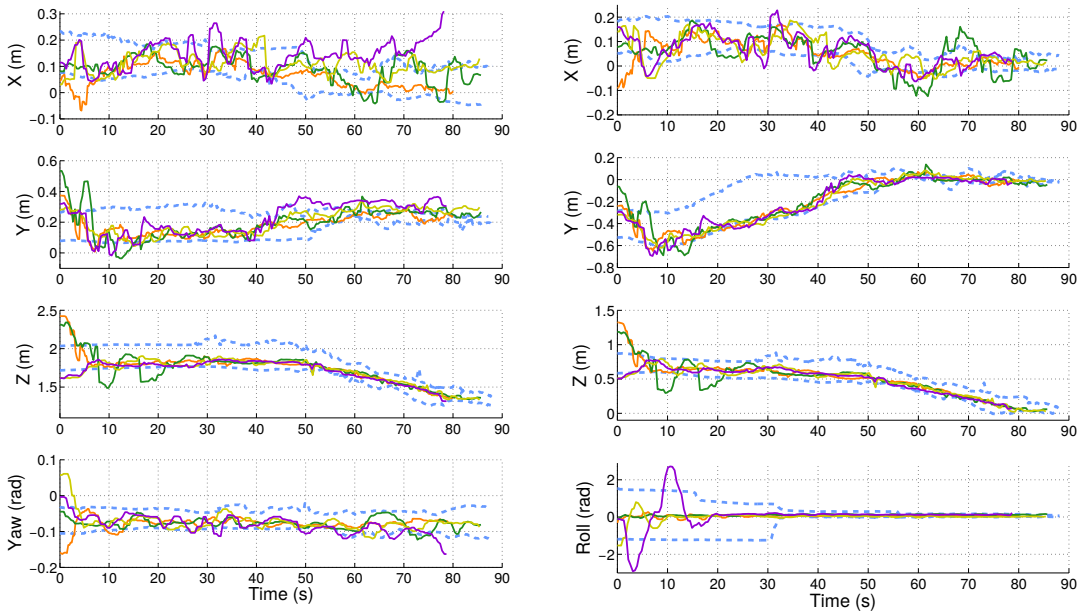


Figure 5.24: Force and Torque recorded in the z axis in the final part of the valve turning task. The force is depicted using a blue solid line and the torque is depicted in green slashed line. The graph highlights four different areas: contact (yellow), constant contact (orange), turning (red) and green (release).

Figure 5.25 compares 4 autonomous trajectories obtained under different levels of perturbation. We can appreciate that the DOFs that are most affected by the perturbations are the AUV's x and yaw and, as a consequence, the end-effector's x. However, the LbD is able to compensate for the AUV's error by moving the manipulator to a different configuration in order to reach a similar end-effector position. To clarify this behaviour, Figure 5.26a shows two trajectories for the end-effector's x DOF with respect to the manipulator's base, one with no-perturbations (orange) and another with high-perturbations (purple). We can appreciate how the end-effector's position with respect to the manipulator's base was significantly different in order to reach the same position with respect to the target valve, thus compensating for the AUV's position error. On the other hand, Figure 5.26b shows the difference between the velocity commands in the x axis between a trajectory with no-perturbations and low-perturbations where the velocity of the vehicle was able to compensate for the perturbations increasing the velocity commands.

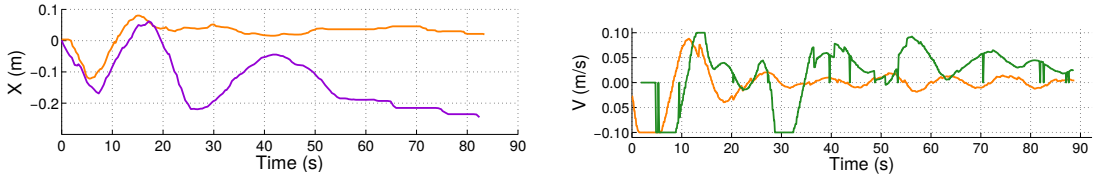
Table 5.2 shows the average distance and standard deviation of the AUV's and end-effector's final positions between the demonstrated and the autonomously reproduced trajectories. As expected, the distance for the AUV increases with the perturbations. However, the distances measured in the end-effector are lower and almost equal for no-perturbations, low-perturbations, and moderate-perturbations experiments. This means that the LbD algorithm is able to find a new configuration for the manipulator that compensates for the difference in the AUV's position. For high-perturbations, the average distance is high for both the AUV and the end-effector because the manipulator's workspace is insufficient to compensate for the difference in the AUV's position, failing to grasp the valve.



(a) AUV positions and orientations

(b) End-effector positions and orientations

Figure 5.25: Four autonomous trajectories depicted together with the upper and lower limits of the demonstrations (dashed-blue). The trajectory under no-perturbations is depicted in orange, low-perturbations in green, medium-perturbations in yellow and high-perturbations in purple. All the trajectories are represented in the frame of the target valve. Each plot shows a single DOF for the manipulator and the end-effector. The time axis corresponds to the real time of the experiments, which, in this case, is the equivalent to the one generated by the canonical system



(a) Two trajectories of the x DOF of the end-effector, using as a frame of reference the base of the manipulator. Orange trajectory under no-perturbations and purple under high-perturbations.

(b) Two AUV velocity commands in the x DOF recorded during under no-perturbations (orange) and low-perturbations (green).

Figure 5.26: Two complementary plots of Figure 5.25 to show the effects of underwater perturbations on the behaviour of the AUV and end-effector.

The success rate of correctly performed valve turns throughout the entire experiment has been 64.8% (46 out of 71 attempts). Results grouped by perturbation intervals can be seen in Table 5.3. This table shows that the LbD system offers a reasonably good performance when the conditions are similar to those of the demonstrations (no-perturbations). The 12.5% error observed in this case can be attributed either to a bad detection of the target's pose (i.e., valve pose) that relocates the whole trajectory causing the vehicle to miss the valve, or to a problem in the manipulator's calibration. To diminish problems caused by the later, a re-calibration procedure was scheduled every two valve turning attempts.

	Perturbations (%)			
	[0-20)	**[20-35)**	**[35-50)**	**[50-70)**
AUV Mean	0.0689	0.0778	0.0894	0.1530
AUV σ	0.0225	0.0255	0.0244	0.0884
end-effector Mean	0.0489	0.0415	0.0530	0.1150
end-effector σ	0.0187	0.0171	0.0203	0.1158

Table 5.2: Mean and standard deviations of the euclidean distance (m) between the end position of the average demonstrated trajectory and the reproduction trajectory for each perturbation interval.

Under low-perturbations, moderate-perturbations and high-perturbations the success rate decreases. However, the system is still able to perform more than half of the attempted valve turnings. On the other hand, under maximum-perturbations, the AUV is not able to keep its position in front of the panel, making the task infeasible.

	Perturbations (%)				
	[0-20)	**[20-35)**	**[35-50)**	**[50-70)**	**[70-100)**
Rate	87.5	64.28	59.09	50.0	0.0
Attempts	24	14	22	6	5

Table 5.3: Percentage of success rates associated with each perturbation group.

5.6.2 Discussion

The proposed method has been tested extensively with this configuration and has provided good results in several conditions. For this reason, they have been used in the long term experiment of the PANDORA project, presented in the next section. Hence, the method has been integrated with other elements developed in the project to perform a persistent experiment of more than 3 hours in a water tank.

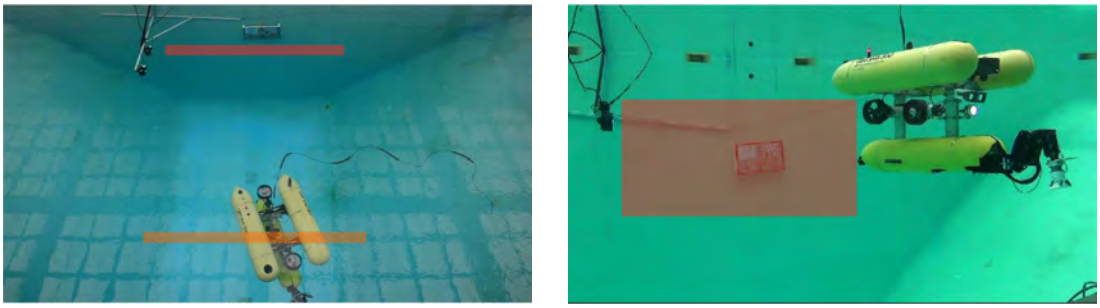
However, there are some parts that can be improved in order to obtain a more robust system. We have focused on improving two issues detected during this experiment. First, the difficulty of performing the valve turning under high-perturbations and maximum-perturbations. Hence, to improve the success rate under these conditions, using multiple strategies learned by a parametric LbD has been proposed. This method is presented in Section 3.4.3 with its results shown in Section 5.8. Second, the increment of parameters obtained with the organization in groups of learned DOFs makes it more difficult to adjust the parameters properly. For this reason, it has been decided to aid the user by adding a system to auto-tune those values. This method is presented in Section 3.4.4 with its

results shown in Section 5.9.

5.7 Long term

As part of the final demonstration of the PANDORA project, the proposed intervention framework composed of visual based detection, a localization system, a robust control scheme, and a reactive decision evaluation together with a temporal planing architecture [Cashmore et al.2015] was set-up to operate a repetitive valve panel mock-up under several disturbances including water currents, blocked valves, unknown panel position and panel occlusion.

This long-term mission was carried out in the water tank using the GIRONA 500 I-AUV. The vehicle had to locate the intervention panel from various locations (see Figure 5.27) and modify the valve handles to achieve different panel configurations. The planning algorithm generated the inspection points to locate the panel, which was located by a vision-based detection system. The vision system also determined the state of the valve handles and was used by the planning system to generate the actions to turn the valves and to achieve the desired panel configuration. The action of turning the valve was programmed using the LbD intervention framework, using the same model learned in the experiments in Section 5.6. After attempting a valve turning, the planning checked the state of the valves again, and decided new actions. For more than 3 hours, the I-AUV changed the panel to 9 different configurations, requiring the turning of 29 valves.



(a) Top View **(b)** Underwater View

Figure 5.27: In these images you can see the GIRONA 500 I-AUV performing an inspection of the defined points during a long term mission. The inspection points are defined between 2 and 3 meters from the walls of the water tank (red and orange area in the left image). From these positions, the visual system covers all possible positions for the sub-sea panel (e.g., red area in the right image).

5.7.1 Results

In order to accomplish the 9 different configurations, the planner generated 37 valve turning actions: 23 were successful, 10 failed because the valve was blocked, and 4 failed because

the platform could not execute the action due to the high current perturbations. Tables 5.4 and 5.5 show the summarized results of the experiment. The *unknown panel position* field represents the area in which the panel's position was known but was then unknown as the panel was moved to a new position in the water tank. The *panel occluded* field represents when some external perturbation was introduced occluding the panel and making the reactive system respond.

Figure 5.28 shows an autonomous trajectory obtained with the combination of the DMP algorithm and the Reactive Fuzzy Decision Maker (RFDM) module [Ahmadzadeh et al.2014]. For the long term experiment only the perturbation of occluding the sub-sea panel has been included, other perturbations detected by this system (i.e., sub-sea panel and AUV instability) have been tested individually after the long term experiment with similar results. In this case, the panel has been occluded by placing another object in front of it so the vehicle's cameras were unable to detect it during a valve turning. In the attempt in Figure 5.28, the panel was occluded for two time intervals: from second 40 to 70 (green area) and from 100 to 120 (yellow area). During this attempt, the system was under low-perturbations currents (the trajectory corresponds to the configuration 6 turning valve 0). Due to the similarity between both the AUV's and end-effector's trajectories, only the later have been plotted. The first thing to note is that the total time to complete the intervention task has increased from 90 to 160 seconds given that the trajectory reproduced in the learning system has been paused and moved backward when the panel has been occluded and forward again when the panel has come back into sight. Between seconds 40 and 70, the trajectory reproduction has been paused as the RFDM was too unsure to keep with the reproduction as a consequence of the panel occlusion. During the second occlusion, the same thing happened but, in this case, due to the proximity of the vehicle to the panel, the reactive control triggered the reproduction of the learned trajectory backwards for safety. This can be specially appreciated in the y and z DOFs between seconds 100 to 120, where the trajectory is reproduced backwards. Finally, when the panel comes back into sight, the end-effector continues its way to the valve after second 125. The spike around second 122 (x and z DOFs) is produced by the new detection of the sub-sea panel causing a correction in the estimated position in the EKF-SLAM algorithm, which produces a small but abrupt jump in the end-effector's position.

5.7.2 Discussion

The Long term experiment has helped the author to validate the concept developed in this thesis and demonstrate that the LbD intervention framework could be integrated in an AUV to perform underwater tasks. The LbD offers a flexible method to be configurable according to the needs of the task. Also, this method has been able to work under different types of perturbations. Moreover, the method has been integrated with a planning system, so in the future, it could be possible to teach different tasks and include them in the planning in

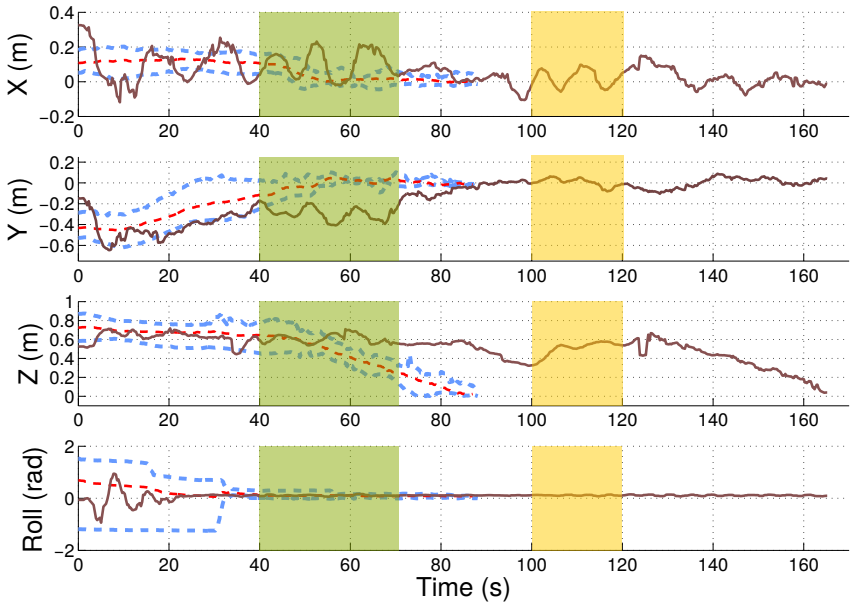


Figure 5.28: Autonomous trajectory obtained during an experiment with two visual occlusions of the sub-sea panel. The autonomous trajectory of the end-effector is depicted in brown together with the upper and lower limits of the demonstrations (dashed-blue) and the average trajectory (dashed-red). The panel has been occluded during two periods of time (highlighted with a green and yellow areas). The time axis corresponds to the real time of the experiments.

order to use the same algorithm with different models according to the requested action.

Configuration	1					2			3					4			5					
Valve to turn	v0	v1	v2	v3	v4	v0	v1	v3	v0	v1	v2	v2	v3	v0	v1	v3	v0	v1	v1	v2	v2	v3
Panel pose unknown																						
Panel occluded																						
% Thrust, perturbation	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	30	0
Achieved	S	S	B	B	S	S	S	S	S	S	B	B	S	S	S	S	S	F	S	B	B	S

Table 5.4: Summary table of the long-term experiment including the first 5 panel configurations. Where, the achieved raw can be successful(S), detected as blocked(B), and failure (F).

Configuration	6			7					8				9		
Valve to turn	v0	v1	v3	v0	v0	v1	v2	v2	v0	v1	v3	v3	v0	v1	v1
Panel pose unknown															
Panel occluded															
% Thrust, perturbation	30	35	35	50	50	35	25	25	25	15	35	35	35	60	60
Achieved	S	S	S	F	S	S	B	B	S	S	B	B	S	F	F

Table 5.5: Summary table of the long-term experiment including the last 4 panel configurations. Where, the achieved raw can be successful(S), detected as blocked(B), and failure (F).

5.8 Parametric DMP against strong perturbations

The purpose of this experiment is to test the feasibility of the GIRONA 500 I-AUV to adapt its strategy to preform a task under different underwater conditions in order to increase its success rate in different current conditions. With this goal in mind a p-DMP algorithm to merge different learned models according to a set of parameters defined by the environment has been developed. In this experiment, the parameters are defined according to the water current perceived by the vehicle. To generate the underwater currents, a system with two thrusters providing up to 20 kg each have been placed in the vicinity of the sub-sea panel (Figure 5.21 shows the test environment, which is the same as for Sections 5.6 and 5.7).

To resolve the valve turning task the operator performed two batches of demonstrations, one under no-perturbations ([0-20)%) and another under the high-perturbations ([50-80)%). Each one of these scenarios requires a different strategy to solve correctly the turning valve task. Without perturbations the strategy is to perform the *close manipulation* strategy presented before. The AUV is perpendicular in front of the panel while the manipulator is moved to an appropriate configuration to manipulate the valve, and afterwards, the valve is grasped and turned combining the movement of the AUV and the manipulator. In the second case, under high perturbations, the AUV can not keep a perpendicular position in front of the sub-sea panel. Then, the panel is approached with an angle near 45°. At this angle, the vehicle is able to hold its position and detect the panel. Meanwhile, the manipulator is moved laterally to an appropriate configuration to manipulate the valve. Finally, after adjusting the position of the AUV and the manipulator, the valve is grasped and turned. Note that due to inherent manipulator limitations, it is only possible to operate the valve from the right side of the AUV. For this reason the current perturbations have always been applied from the same side. Figure 5.29 represents two possible sets of demonstrations and learned models for the two strategies.

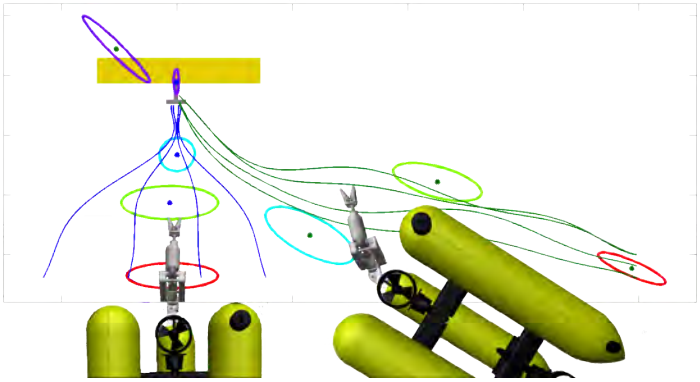


Figure 5.29: Two batches of simulated demonstrations, and the DMP model learned in a 2D representation of the end-effector trajectory.

Since the GIRONA 500 I-AUV didn't have any specific sensor to measure the underwater currents, a module to estimate the currents affecting the vehicle has been developed. This module requires an initialization phase where the vehicle needs to discover the necessary forces to keep a static position during a period of time in regular conditions. To obtain a better estimation, the static position is relative to a static landmark (i.e., in front of a sub-sea valve panel). Once this phase has been performed, the values are stored and can be reused as long as the environmental conditions for the initial/calm state do not change. Before starting an intervention attempt, this module is activated and tries to keep the same position for the same period of time as in the initialization phase. Then, the forces are compared and the current is estimated. Figure 5.30 shows the inclusion of this new module and the p-DMP.

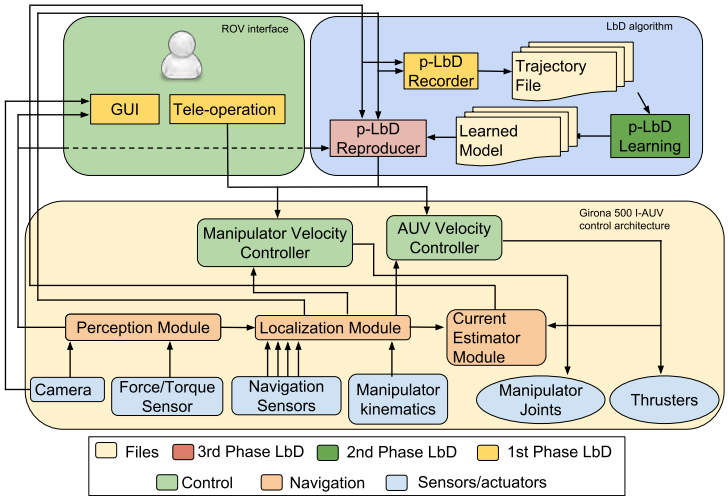


Figure 5.30: Modified Framework architecture including the p-DMP and the current estimator module.

5.8.1 Results

The proposed approach has been validated under different degrees of perturbation, some similar to the situation in the demonstrations and others in new conditions. All the valves on the sub-sea panel have been turned several times achieving different success rates.

Two different models have been learned, the zero perturbation strategy has an average associated force of x=0.1, y=0.5 and z=0.4 N and the high perturbations strategy has an average associated force of x=1.2, y=8.5 and z=-2.5 N. We can appreciate that the most affected parameter is the y axis, but the z and x axis also become affected as a consequence (Fig. 5.31 shows the set of demonstrations). Tables 5.6 and 5.7 show the different parameters found experimentally to learn the valve turning task with the two strategies.

The first parameter to be adjusted is the number of Gaussians. A high number of Gaussians allows the representation of movements with a lot of restrictions and low flexibility, while a small number is better to represent trajectories with more variability. The duration of the demonstrations also influences the number of Gaussians needed, since they are uniformly distributed along the trajectory. In our case, this value has been adjusted experimentally to 8 and 11 Gaussians for the zero and the high current strategies. The trajectory with no perturbations needs less time to resolve the task (75 seconds) compared to the high perturbation strategy which needs 110 seconds. For this reason, they require a different number of Gaussians to obtain similar results.

The other two parameters are K^P_{min} and K^P_{max}. In this case, it has been possible to adjust the two parameters with a smaller difference to obtain smoother trajectories than in previous experiments for the no-perturbations model, since it will not cope directly with the perturbations. However, the difference between the two parameters chosen for the high-perturbation example are bigger, due to the perturbations effecting the trajectory.

		Parameters for the DMP		
		Gaussians	K^P_{min}	K^P_{min}
AUV	x, y, yaw	8	0.5	1.5
AUV	z	8	0.1	1.0
End-effector	x, y, $roll$	8	1.5	3.0
End-effector	z	8	2.0	3.0

Table 5.6: Parameters defined for the strategy without underwater currents.

		Parameters for the DMP		
		Gaussians	K^P_{min}	K^P_{min}
AUV	x, y, yaw	11	0.1	2.5
AUV	z	11	0.1	2.0
End-effector	x, y, $roll$	11	1.0	3.0
End-effector	z	11	2.0	3.0

Table 5.7: Parameters defined for the strategy with high underwater currents.

We have performed an experiment consisting of three turns for each of four valves in three different environments: no-perturbations, moderate-perturbations ([20-50)%), and high-perturbations ([50-80)%). The success rate of the correctly performed valve turnings throughout the whole experiment has been 80.05% (29 out of 36 attempts). The 19.95% error can be attributed to the sum of different small errors in the estimation of the target's

pose (i.e. valve pose) and an error in the estimation of the manipulator's calibration. To diminish the problem in the arm-calibration, a re-calibration procedure is performed every two valve turning attempts.

	Perturbations (%)		
	[0-20)	**[20-50)**	**[50-80)**
Rate	91.6	75.0	75.0
Attempts	12	12	12

Table 5.8: Percentage of success rates associated with each perturbation group.

Table 5.8 shows the success rate in the different environment situations. It illustrates how the perturbations introduce more instability in the performance of the AUV. However, although the new strategy is more complex to perform, it is more robust when confronted with perturbations. Comparing these results with previous experiments in Section 5.6 where there was only one strategy (perpendicular approach), the success rate has increased from 58.33% to 75% in moderate perturbations and from 27.27% to 75% in high perturbations.

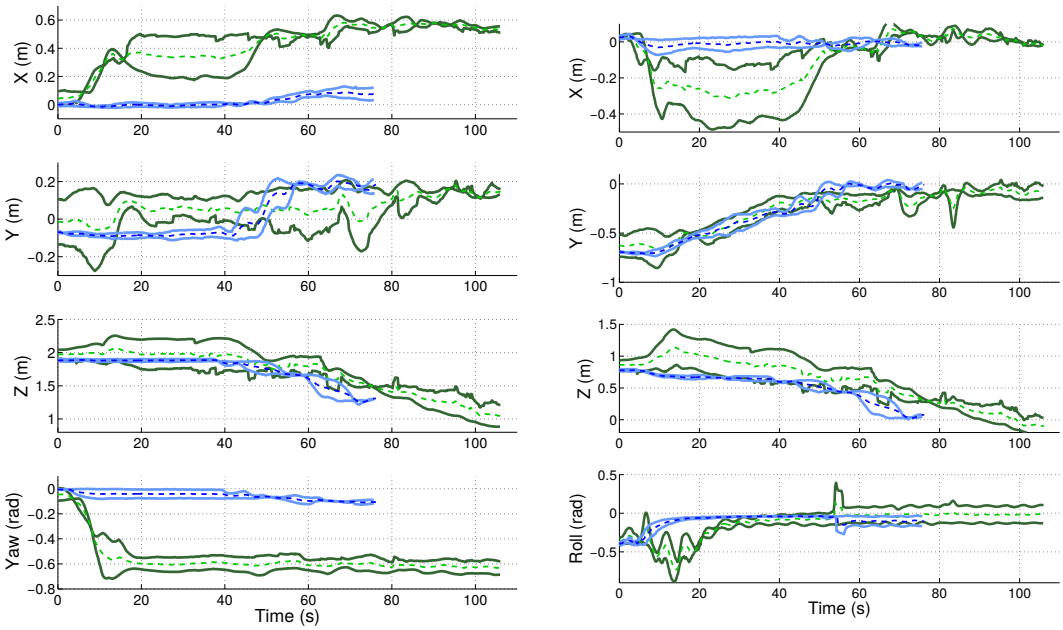
Figure 5.32 shows the two groups of demonstrated trajectories and one successful trajectory for each group. The autonomous trajectories depicted in the figure for no-perturbation (x=0.0, y=0.41 and z=0.6 N) and high-perturbations (x=1.4, y=8.0 and z=-2.4 N) show how the AUV and the end-effector follow the average trajectory of the demonstration in a smooth movement. The moderate perturbation (x=0.7, y=4.1 and z=-1.1 N) trajectory follows a new strategy which requires an average of 90 second to be performed and follows a new trajectory that is a mixture of the two learned models.

5.8.2 Discussion

The results obtained with these experiments have proved the suitability of the proposed p-DMP to increase the success rate under underwater currents. The method has been able to combine the strategies taught depending on the environmental conditions at each execution.

However, manipulation of the sub-sea panel by the AUV from a lateral position introduces a new difficulty, which was to detect the sub-sea panel and the valve positions correctly. The solution adopted in this case was a temporary solution where the orientation of the camera was adapted manually before starting the experiment. For more complete work, the use of a rotatory payload able to move the camera to the correct position will be needed, to ensure good quality detections.

Moreover, with the p-DMP, the number of parameters has increased considerably, from the first version of the DMP, where only 3 needed to be adjusted, to this case with 24 (3 parameters of the DMP x 4 groups of the DOFs x 2 types of strategies). For this reason,



(a) End-effector trajectory

(b) End-effector orientation

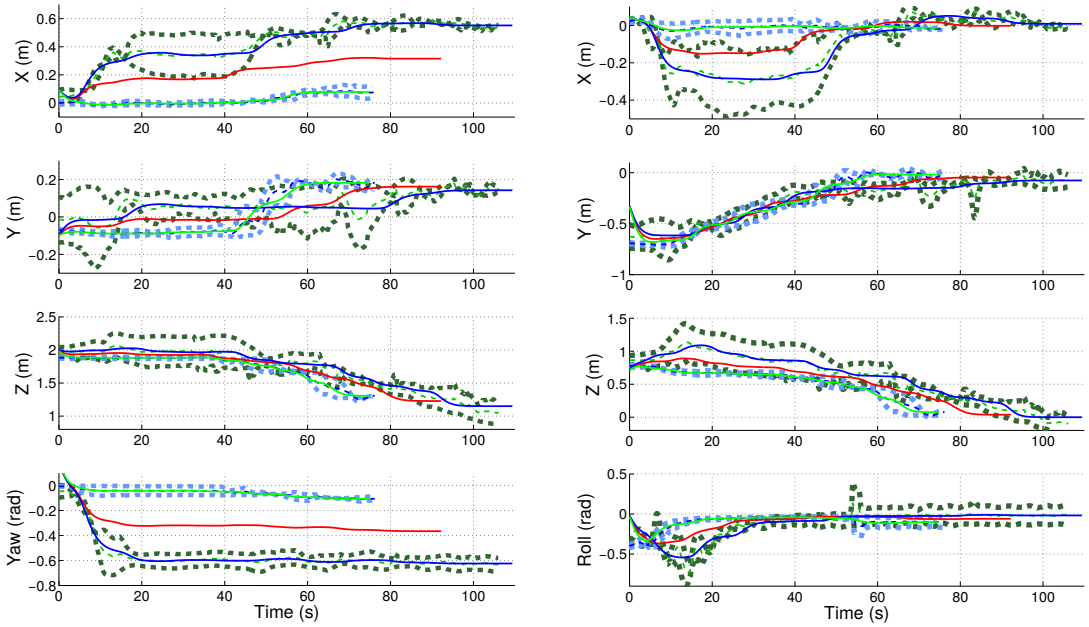
Figure 5.31: Average trajectory (dashed-line) and upper and lower limits of the 4 demonstrated trajectories. Each plot shows a single DOF for the manipulator and the end-effector. Depicted in blue, the group of no perturbations and in green the demonstrations under perturbations. All the trajectories are represented in the frame of the target valve.

the next section presents an algorithm to adjust these values automatically.

5.9 Automatic Learning

The number of parameters to be adjusted has increased considerably with the last modifications of the DMP. To simplify the use of the algorithm for diverse tasks with I-AUV, an algorithm (Section 3.4.4) to find a correct set of parameters has been proposed. This algorithm performs a global search in order to obtain the most accurate model according to a defined comparison method in the range of the defined parameters.

Defining a strategy to compare different models is paramount to use the method correctly. In our case, we decided to compare the two models by comparing the average trajectory obtained from the batch of demonstrations with a simulated trajectory generated by each model. To simplify the comparison, both trajectories have been re-sampled with the same number of time intervals. A weighted average of the difference between the two trajectories (error) is computed since the most relevant part of the task is the final part of the trajectory. Hence, the weight increases monotonically until the last ten seconds of the trajectory when the weight doubles that of the initial part of the trajectory. This weighted average error is used to define the function $compare\_traj\_dof()$. Moreover, the comparison is evaluated according to the defined groups of DOFs. Hence, a set of



(a) AUV trajectory

(b) End-effector trajectory

Figure 5.32: Three autonomous trajectories are depicted together with the average and upper and lower limits of the two demonstrated groups (dashed green and blue lines). The trajectory under no perturbations is depicted in green, moderate-perturbations in red, high-perturbations in blue. All the trajectories are represented in the frame of the target valve. Each plot shows a single DOF for the manipulator and the end-effector. The time axis corresponds to the real time of the experiments, which, in this case, is the equivalent to the one generated by the canonical system.

parameters is better than another if the error for all the DOFs is smaller.

To simulate the trajectories generated, the same dynamics model used in the UWSim simulator presented in the Section 5.3 is used. The dynamics model has been uncoupled from the simulator and can be used without any time restriction to speed up the reproduction.

5.9.1 Results

To test the algorithm the same trajectories used in the experiments with perturbations in the Section 5.6 have been used and the exploration parameters are presented in Table 5.9.

Gaussian min	Gaussian max	Gaussian Step	K^P_{min}	K^P_{max}	K^P Step
3	20	1	0.0	6.5	0.1

Table 5.9: Exploration parameters used to find the best parameter combination.

The time required to find a proper solution is highly dependent on the exploration parameters selected by the user. Due to the algorithm's definition, it explores all the possible values so the time needed can be defined as follows:

$$time = \left(\left(\frac{n_{params}!}{(n_{params}-2)!2!} \right) * (t_m + t_s) \right) * \left(\frac{G_{max} - G_{min}}{G\text{step}} + 1 \right), \quad \text{where } n_{params} \text{ is: } \frac{(K^P_{max} - K^P_{min})}{K^P\text{step}} + 1. \tag{5.1}$$

The t_m and t_s are the time required to generate the model and simulate the trajectory respectively. Using an Intel Core i7-4710HQ CPU @ 2.50GHz and 12 GB of RAM pc this operation takes approximately 2 seconds. This time could be reduced by removing the writing in a file of all the obtained trajectories, but they have been used to verify the results and reach other conclusions studying the results. However, the method has been optimized, parallelising the exploration of the Gaussian parameter.

Table 5.10 presents the results using the same groups from the previous experiments. We can see that the number of Gaussian is the maximum of the range defined. This is expected since the algorithm searches for the best accuracy in the trajectory, and the increase of Gaussians increases the accuracy. On the other hand, we can appreciate that the average error of the end-effector is bigger than the AUV's. The error in both cases is small, but since this contradicts with the statement that the AUV is less accurate than the end-effector, the authors have decided to separate the *yaw* and *roll* and repeat the exploration.

Table 5.11 presents the results found with the new grouping of the DOFs. We can see that the error in the case of the end-effector has decreased, so the influence of the different control in roll was influencing the other DOFs and will be better to separate the angles.

Finally, Table 5.12 shows the results obtained with the parameters defined manually for the experiment with perturbations presented in Section 5.6. We can see here that the average error is also small, but not as small as the values found by the algorithm proposed in this section.

		Parameters for the DMP			
		Gaussians	K^P_{min}	K^P_{min}	**Avg Error**
AUV	x, y, yaw	20	0.0	3.0	[8.588e−4, 0.006, 0.0024]
AUV	z	20	0.0	1.4	0.0018
End-effector	x, y, $roll$	20	0.0	3.3	[0.0037, 0.0177, 0.153]
End-effector	z	20	1.0	1.8	6.88e−7

Table 5.10: Best parameters obtained using the search parameters from Table 5.9 and the {x, y, $angle$} and {z} DOFs groups for the AUV and manipulator.

		Parameters for the DMP			
		Gaussians	K^P_{min}	K^P_{min}	**Avg Error**
AUV	x, y	20	0.0	3.0	[8.588e−4, 0.006]
AUV	z	20	0.0	1.4	0.0018
AUV	yaw	20	0.0	0.6	2.6715e−4
End-effector	x, y	20	0.3	0.6	[1.377e−4, 3.592e−4]
End-effector	z	20	1.0	1.8	6.88e−7
End-effector	$roll$	20	5.4	5.5	0.0089

Table 5.11: Best parameters obtained using the search parameters from Table 5.9 and the new organization of the DOFs, which is as follows $\{x, y\}$, $\{angle\}$ and $\{z\}$ for the AUV and manipulator.

		Parameters for the DMP			
		Gaussians	K^P_{min}	K^P_{min}	**Avg Error**
AUV	x, y, yaw	15	0.5	2.5	[0.0829, 0.0188, 0.0752]
AUV	z	11	0.1	1.0	0.0124
End-effector	x, y, $roll$	15	1.0	3.0	[2.084e−3, 1.681e−3, 0.0109]
End-effector	z	11	2.0	3.0	0.0016

Table 5.12: Average errors obtained using the parameters from Section 5.6 and the $\{x, y, angle\}$ and $\{z\}$ DOFs groups for the AUV and manipulator.

5.9.2 Discussion

The authors are aware that the algorithm proposed is not efficient and requires a huge amount of time to compute the results. However, the aim was focused on testing the correct definition of an evaluation function for the models generated. This function can be used together with more sophisticated and efficient methods in order to find a proper K^P_{min}, K^P_{max}, and number of Gaussians faster. On the other hand, when exploring the results obtained by the algorithm in order to verify the values found, the authors have realized that several parameters obtain the same results or close values. This means that there are several options to find a correct configuration, which allows a rougher exploration and the use of local search methods instead of global methods.

The algorithm has found different parameters from those defined by the user in the real experiments. Even though the model found by those parameters was good enough to obtain an acceptable average error in the simulation and obtained good results in the real environment. However, it is important to point out that simulators usually have differences with real elements, so a perfect result in a simulation could not be the optimal value in a

real environment.

To summarize, the results obtained using this algorithm could help new users of the DMP algorithm and the intervention framework proposed to find appropriate values for the parameters in order to learn a model. It can also be used when the method is applied with a new AUV, manipulator or controller to find a first version of the new parameters. On the other hand, the fact that multiple parameter values obtain similar values proves again the robustness of the method, and allows greater flexibility in the definition of the parameters.

6

Conclusion

This chapter concludes this thesis by presenting a summary of the completed work in Section 6.1. We review the main contributions under Section 6.2. Finally, compelling areas for future work are outlined in Section 6.4.

6.1 Summary of completed work

This thesis was devoted to resolving an underwater intervention task (i.e., valve turning) by means of LbD employing an I-AUV. Introducing autonomous systems to address underwater intervention tasks supposes a reduction in personal, dedicated equipment, time and consequently economic cost. Hence, the vessel is only required for the deployment and recovery phases, and the specialised personnel is reduced. Furthermore, the automation of a repetitive task usually improves accuracy, repeatability, and robustness eliminating errors due to fatigue in a human operator. On the other hand, the LbD offers a flexible framework where the robot is taught in a natural way by a human operator. Hence, the LbD reduces the cost of programming a new task, eliminating the need for a specialised team of developers. The combination of these two elements opens the door to current and new users to perform more and different kinds of interventions, due to cost reduction and the flexibility and robustness of the method.

Chapter 2 introduced an overview of machine learning techniques and then focused on LbD methods. We presented the common elements of LbD methods and different ways to classify them according: the recording of the demonstrations (demonstration or imitation), the number of demonstrations (in a batch or incrementally), the level of learning (symbolic or trajectory), and the kind of encoding and reproduction algorithm (classification or regression). To illustrate the different types, the following four LbD methods were explained: Task Graph through LCS (Section 2.3.1) and Global task graph for MP from kinesthetic Demonstration (Section 2.3.2) to illustrate symbolic learning, and GMM with GMR (Section 2.3.3) and DMP (Section 2.3.4) to exemplify trajectory learning. Finally, to conclude the Chapter, a list of significant methods for robotics is presented detailing the demonstration process, the level of the skill learned, the encoding and reproduction, a brief description, and the application used to validate the method.

Chapter 3 focused on the LbD framework developed in this thesis to perform an underwater intervention using an I-AUV. First, the DMP method, used as the core of the intervention framework was presented and compared with the classical DMP implementation presented in Chapter 2. Next, a generic version of the underwater framework was explained, listing the required elements to build the framework and their interactions. Furthermore, this section has presented the three phases (demonstration, learning, and reproduction) necessary to perform the intervention using the developed framework. To finalise, the various modifications carried out on the DMP in order to tailor the method to the complexity of the underwater environment were detailed.

Chapter 4 concentrated on the underwater intervention environment. It started with a state-of-the-art of the work developed during the last two decades in this topic. It included several projects devoted to advance the state-of-the-art designing innovative ways to perform the underwater intervention task, usually requiring the development of

new methods and dedicated vehicles. Afterwards, a brief description of the valve turning task designed in this thesis was presented. Then, the chapter continued introducing the different elements used to resolve the valve turning task composed of: the GIRONA 500 AUV mechatronics description and software architecture; the ECA ARM 5E manipulator mechatronic description, dynamics description, software control, and calibration methods; and the custom end-effector with the different designs developed and the sensors included (camera in-hand and F/T sensor). Moreover, the chapter presented two methods to control in a coupled way the vehicle and manipulator only providing commands to the end-effector element.

Chapter 5 provided an extensive summary of all the experimental results obtained in the thesis. The authors followed an incremental approach presenting in each iteration a more challenging valve turning task. The experiments started with the laboratory tests using an industrial manipulator and an optical tracking system to validate the proposed approach. After, still in the way to prove the concept, other tests were performed in an underwater simulator including the GIRONA 500 AUV and the ECA ARM E5 Manipulator. Next results were presented using an uncoupled control for the GIRONA 500 AUV and its manipulator in the water tank. Then, similar experiments were performed using the UVMS controller with the goal to analyse which control approach offers better benefits for our task. Afterwards, the Chapter 5 presented the first results of the intervention under moderate and low perturbations. Then, the long term experiments were presented in the Chapter, where the previous method was integrated with the work of the other partners in the PANDORA project. As a consequence of the results observed in the long term experiments was developed a Parametric DMP to overcome the difficulties to perform the task under strong perturbations. Finally, were presented the results of an algorithm to adjust the parameters of the proposed DMP automatically. To sum up, this chapter proved the viability of the proposed approach to resolve underwater intervention tasks using LbD.

6.2 Review of Contributions

This thesis has advanced the current state-of-the-art of underwater interventions proposing a new framework based on a LbD algorithm, granting a new flexible and natural way to program complex tasks for an I-AUV. Moreover, this is the first time that well-known LbD techniques for industrial manipulators have been introduced into an underwater environment and consequently to an I-AUV. This overall contribution can be broken down into more specific steps:

Human robot interaction: an essential part to be able to use LbD is the capability to perform the demonstrations correctly, and in a comfortable and natural way. During the development of this Thesis, the authors have improved the human-robot interaction including different systems and aids for the operators. The final approach

was composed of a system with multiple cameras, one in the end-effector and another in a central part of the vehicle's frame. The images were augmented with information from the localization and perception systems. The other key part has been the control of the vehicle and end-effector, for which different kinds of controllers were used. First, a 7 DOF Haptic device was integrated, which allowed the control of the vehicle and the manipulator with one hand, as well as perceiving the force and torque measured by the F/T sensor mounted in the end-effector. Lately, a gamepad controller was developed, combining the control of the AUV and the end-effector or splitting the control in two different gamepads with two operators.

Extension of a LbD algorithm in an underwater environment: the DMP as the chosen LbD method was extended and tailored to the particularities of the underwater intervention and the I-AUV control.The main contributions is the increment from 3 or 6 DOFs to 10 DOFs, requiring a different organization in order to consider the overlapping of the DOFs and the influence between them correctly. On the other hand, a parametrized version has also been developed to be able to learn multiple styles to perform the task according to an external parameter and mix these models to generate new versions of the skill learned, more suitable to the new conditions. Furthermore, minor changes have been applied to the method to control the time allowing the reproduction in different speeds and to advance or to move back the trajectory according to an external parameter.

Development of an intervention framework: an intervention framework has been constructed identifying and developing the required elements or modules necessary to perform an underwater intervention task. The core of this framework is the DMP, but the other components can be easily replaced if they share the same interface. For instance, the vision-based perception system could be replaced for any other perception approach that provides a compatible position estimation. Also, the algorithm in the perception system can be easily replaced to detect other elements different from a valve or a valve panel. On the other hand, the framework proportionates a thigh cooperation between the AUV and the manipulator. All these factors allow to the framework to be reused in different applications where the I-AUV has to interact with one or multiple objects.

Robustness against perturbations: the proposed method has been tested against various degrees of perturbation, the method has proved its robustness generating stronger commands to compensate the external forces or adapting the trajectory according to the external forces. On the other hand, it also has been combined with systems which evaluate the safety detecting instability, failure of sensors, and interference of elements. In case they are detected, the movement of the end-effector and AUV is firstly reduced, then stopped and, finally, moved backwards, waiting for safer

conditions to perform the task or abort it.

6.3 Review of Publications

Throughout this Thesis, the divulgation has been done periodically presenting the results obtained to the scientific community through conferences, and the most relevant and complete results have been presented in journals articles.

The [**CIT-12**] presented the results achieved during the stage at the IIT at the beginning of the thesis, where using an industrial manipulator together with a mock-up scenario, the viability of the DMP algorithm for the underwater valve turning was validated. These results can be read in Section 5.2.

During the next year, in parallel with the integration work between the manipulator and the GIRONA 500 AUV, a preliminary work developed in simulation was selected as a Student Poster in OCEANS Bergen 2013 [**OCEANS-13**]. This work showed an innovative control for I-AUV, and was presented in Section 5.3.

In 2014, after the successful integration of the ECA CSIP Manipulator with the GIRONA 500 AUV, several publications were made. A paper containing first results of GIRONA 500 I-AUV developing an autonomous intervention task without perturbations was chosen as a Student Poster in OCEANS Taipei 2014 [**OCEANS-14**], which has been detailed in Section 5.4. Then, the discussion between the use of an uncoupled controller or a coupled one (using an UVMS) was presented at the ICRA Workshop on Persistent Autonomy for Marine Robotics in Hong Kong [**WS-ICRA-14**], which has been detailed in Section 5.5.. Finally, a paper at CCIA conference [**CCIA-14**] presented an improved version of the intervention framework with satisfactory results under moderated perturbations.

In 2015, two different journal papers were published: in the Patter Recognition Letters [**PR-15**] we presented an exhaustive form of the intervention framework and the different algorithm integrated to perform a long term autonomous intervention. These results can be found in Section 5.6. The other journal paper was published in Autonomous Robots journal [**AR-15**], and discusses the final results obtained in the valve turning scenario of PANDORA project, where the intervention framework developed in this thesis was an essential part. At the same time, the work of extending the intervention framework to learn multiple strategies according to the underwater conditions (presented in Section 5.8) was selected as a Student Poster at OCEANS Genova 2105 [**OCEANS-15**]. Finally, an overview of the obtained results was presented at NGCUV IFAC workshop [**NGCUV-15**].

6.4 Future Work

The work done in this thesis has proposed a first approach in order to facilitate programming of new underwater intervention tasks by a user without any specific knowledge of

programming or control algorithms. However, to obtain a robust system capable of learning any task, there is still a lot of room for improvement. The next points identify some areas worth exploring:

Continuous learning: In order to continuously improve the learned task model and adapt it to a constantly changing underwater environment, the framework could improve the model through experience during the reproductions. There are several methods to perform this kind of exploration. The combination of the RL and the LbD could be one option, where the initial model extracted from the demonstrations using the LbD is improved by reproductions using the RL. The RL could be executed continuously or after a certain number of executions to force a constant adaptation. Another option is to use an ILC algorithm in conjunction with the LbD. The ILC can adjust the accuracy of a repetitive task, so the ILC will improve the LbD model. Finally, another option based more in the interaction between a human an robots is to focus the effort on developing a Cognitive Architecture where the knowledge can be transfered to the robot using a LbD and a human or the cognitive architecture itself can evaluate the performed trajectories and decide to improve the current model with new examples extracted from the reproductions.

Other tasks: This thesis has only focused on one task that can be developed with a passive end-effector. For other tasks, an active end-effector with a gripper or a hand with several fingers might be needed in order to pickup objects. An extension of the current DMP will be needed to include the DOFs to learn the hand's behaviour. On the other hand, other kinds of tasks (e.g., inspecting for corrosion using a potential profile by direct contact) could require a constant use of the F/T sensor. For this reason, it would be interesting to include the necessary F/T information in an extended version of the DMP.

Hierarchical Learning Complex and long tasks are difficult to learn accurately as a whole, since they include several parts which can be different. For this reason, it is interesting to divide a task hierarchically into sub-tasks that can be more easily and precisely learned than a complete task.

Bibliography

[ABB2013] ABB (2013). YuMI Robot. `http://new.abb.com/products/robotics/es/yumi`. 10

[Abbas and MacDonald2011a] Abbas, T. and MacDonald, B. (2011a). Generalizing topological task graphs from multiple symbolic demonstrations in programming by demonstration (PbD) processes. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3816–3821. 25, 26, 40

[Abbas and MacDonald2011b] Abbas, T. and MacDonald, B. A. (2011b). Generalizing topological task graphs from multiple symbolic demonstrations in programming by demonstration (PbD) processes. In *ICRA*, pages 3816–3821. IEEE. 24

[Ahmadzadeh et al.2014] Ahmadzadeh, S. R., Kormushev, P., Jamisola, R. S., and Caldwell, D. G. (2014). Learning reactive robot behavior for autonomous valve turning. In *Humanoid Robots (Humanoids), 2014 IEEE International Conference on*, Madrid,Spain. IEEE. 117

[Amat et al.1999] Amat, J., Monferrer, A., Batlle, J., and Cufi, X. (1999). GARBI: a low-cost underwater vehicle. *Microprocessors and Microsystems*, 23(2):61–67. 13

[Antonelli2014] Antonelli, G. (2014). *Underwater robots*, chapter Kinematic Control of UVMSs, pages 127–168. Cham: Springer International Publishing. 91

[Argall et al.2009] Argall, B., Chernova, S., Veloso, M. M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483. 20

[Armstrong et al.2006] Armstrong, R. A., Singh, H., Torres, J., Nemeth, R. S., Can, A., Roman, C., Eustice, R., Riggs, L., and Garcia-Moliner, G. (2006). Characterizing the deep insular shelf coral reef habitat of the Hind Bank marine conservation district (US Virgin Islands) using the Seabed autonomous underwater vehicle. *Continental Shelf Research*, 26(2):194–205. 8

[Batlle et al.2005] Batlle, J., Ridao, P., Garcia, R., Carreras, M., Cufi, X., El-Fakdi, A., Ribas, D., Nicosevici, T., Batlle, E., Oliver, G., et al. (2005). URIS: Underwater Robotic Intelligent System. *Automation for the Maritime Industries*, pages 177–203. 13

[Bay et al.2008] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359. 73

[Billard et al.2008] Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). Robot Programming by Demonstration. In Siciliano, B. and Khatib, O., editors, *Handbook of Robotics*, pages 1371–1394. Springer, Secaucus, NJ, USA. 11, 20

[Bingham et al.2010] Bingham, B., Foley, B., Singh, H., Camilli, R., Delaporta, K., Eustice, R., Mallios, A., Mindell, D., Roman, C., and Sakellariou, D. (2010). Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle. *Journal of Field Robotics*, 27(6):702–717. 8

[Bishop2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. 31

[Bosch et al.2015] Bosch, J., Gracias, N., Ridao, P., and Ribas, D. (2015). Omnidirectional Underwater Camera Design and Calibration. *Sensors*, 15(3):6033–6065. 14

[Bruno et al.2015] Bruno, D., Calinon, S., Malekzadeh, M., and Caldwell, D. G. (2015). Learning the Stiffness of a Continuous Soft Manipulator from Multiple Demonstrations. In Liu, H., Kubota, N., Zhu, X., Dillmann, R., and Zhou, D., editors, *Intelligent Robotics and Applications*, volume 9246 of *Lecture Notes in Computer Science*, pages 185–195. Springer. 14

[Calinon2009] Calinon, S. (2009). *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press. 23, 26, 31

[Calinon et al.2013] Calinon, S., Alizadeh, T., and Caldwell, D. G. (2013). Improving extrapolation capability of task-parameterized movement models. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Tokyo, Japan. 21

[Calinon and Billard2007] Calinon, S. and Billard, A. (2007). Incremental learning of gestures by imitation in a humanoid robot. In *2007 2nd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 255–262. 23

[Calinon et al.2014] Calinon, S., Bruno, D., and Caldwell, D. G. (2014). A task-parameterized probabilistic model with minimal intervention control. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 3339–3344, Hong Kong, China. 41, 53

[Calinon et al.2010a] Calinon, S., D'halluin, F., Sauser, E. L., Caldwell, D. G., and Billard, A. (2010a). Learning and Reproduction of Gestures by Imitation. *IEEE Robot. Automat. Mag.*, 17(2):44–54. 25, 39, 45

[Calinon et al.2007] Calinon, S., Guenter, F., and Billard, A. (2007). On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2):286–298. 14

[Calinon et al.2010b] Calinon, S., Sardellitti, I., and Caldwell, D. (2010b). Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 249–254. 45, 48

[Calonder et al.2010] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). BRIEF: binary robust independent elementary features. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pages 778–792, Berlin, Heidelberg. Springer-Verlag. 73

[Camilli et al.2010] Camilli, R., Reddy, C. M., Yoerger, D. R., Van Mooy, B. A., Jakuba, M. V., Kinsey, J. C., McIntyre, C. P., Sylva, S. P., and Maloney, J. V. (2010). Tracking hydrocarbon plume transport and biodegradation at Deepwater Horizon. *Science*, 330(6001):201–204. 8

[Carreras et al.2001] Carreras, M., Batlle, J., and Ridao, P. (2001). Hybrid coordination of reinforcement learning-based behaviors for AUV control. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems.*, volume 3, pages 1410–1415. IEEE. 13

[Carreras et al.2013] Carreras, M., Candela, C., Ribas, D., Mallios, A., Magí, L., Vidal, E., Palomeras, N., and Ridao, P. (2013). Sparus II, design of a lightweight hovering AUV. In *5th International Workshop on Marine Technology (Martech)*. SARTI. 13

[Carreras et al.2003] Carreras, M., Tiano, A., El-Fakdi, A., Zirilli, A., and Ridao, P. (2003). On the identification of non linear models of unmanned underwater vehicles. In *First IFAC workshop on guidance and control of underwater vehicles GCUVï£¡03*. 13

[Casalino et al.2001] Casalino, G., Angeletti, D., Bozzo, T., and Marani, G. (2001). Dexterous underwater object manipulation via multi-robot cooperating systems. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 4, pages 3220–3225 vol.4. 58

[Casalino et al.2014] Casalino, G., Caccia, M., Caiti, A., Antonelli, G., Indiveri, G., Melchiorri, C., and Caselli, S. (2014). MARIS: A national project on marine robotics for

interventions. In *22nd Mediterranean Conference on Control and Automation*, pages 864–869. 65

[Cashmore et al.2015] Cashmore, M., Fox, M., Long, D., Magazzeni, D., Carrera, A., Palomeras, N., Hurtos, N., and Carreras, M. (2015). ROSPlan: Planning in the Robot Operating System. In *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, June 7-11, 2015.* 116

[Chernova and Veloso2009] Chernova, S. and Veloso, M. M. (2009). A Confidence-Based Approach to Multi-Robot Learning from Demonstration. In *AAAI Spring Symposium: Agents that Learn from Human Teachers*, pages 20–27. AAAI. 23

[Choi et al.1994] Choi, S., Takashige, G., and Yuh, J. (1994). Experimental study on an underwater robotic vehicle: ODIN. *AUV'94. Proceedings of the 1994 Symposium Autonomous Underwater Vehicle Technology*, pages 79–84. 58

[Clarke et al.2009] Clarke, M. E., Tolimieri, N., and Singh, H. (2009). Using the seabed AUV to assess populations of groundfish in untrawlable areas. In *The future of fisheries science in North America*, pages 357–372. Springer. 8

[Cortes and Vapnik1995] Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. *Mach. Learn.*, 20(3):273–297. 31

[Cutler et al.2015] Cutler, M., Walsh, T. J., and How, J. P. (2015). Real-World Reinforcement Learning via Multifidelity Simulators. *IEEE Transactions on Robotics*, 31(3):655–671. 19

[Dautenhahn and Nehaniv2002] Dautenhahn, K. and Nehaniv, C. L., editors (2002). *Imitation in Animals and Artifacts*. MIT Press, Cambridge, MA, USA. 20

[Dempster et al.1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B(39):1–38. 32

[Denavit and Hartenberg1965] Denavit, J. and Hartenberg, R. S. (1965). A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. *Trans. ASME, J. Appl. Mech.*, 22(2):215–221. 82

[Ekvall and Kragic2006] Ekvall, S. and Kragic, D. (2006). Learning Task Models from Multiple Human Demonstrations. In *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, pages 358–363. 24, 40

[El-Fakdi and Carreras2013] El-Fakdi, A. and Carreras, M. (2013). Two-step gradient-based reinforcement learning for underwater robotics behavior learning. *Robotics and Autonomous Systems*, 61(3):271–282. 13

[Elibol et al.2011] Elibol, A., Garcia, R., and Gracias, N. (2011). A new global alignment approach for underwater optical mapping. *Ocean Engineering*, 38(10):1207–1219. 13

[Evans et al.2001] Evans, J., Keller, K., Smith, J., Marty, P., and Rigaud, O. (2001). Docking techniques and evaluation trials of the SWIMMER AUV: an autonomous deployment AUV for work-class ROVs. *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, 1:520–528. 58, 59

[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. *OCEANS 2003*, 4:2201–2210. 58, 60

[Farr et al.2010] Farr, N., Bowen, A., Ware, J., Pontbriand, C., and Tivey, M. (2010). An integrated, underwater optical /acoustic communications system. In *OCEANS 2010 IEEE - Sydney*, pages 1–6. 64

[Fernandez et al.2013] Fernandez, J., Prats, M., Sanz, P., Garcia, J., Marin, R., Robinson, M., Ribas, D., and Ridao, P. (2013). Grasping for the Seabed: Developing a New Underwater Robot Arm for Shallow-Water Intervention. *Robotics Automation Magazine, IEEE*, PP(99):1–1. 61

[Ferrer et al.2007] Ferrer, J., Elibol, A., Delaunoy, O., Gracias, N., and Garcia, R. (2007). Large-area photo-mosaics using global alignment and navigation data. In *MTS/IEEE OCEANS Conference, Vancouver, Canada*, pages 1–9. 13

[Foley et al.2009] Foley, B. P., Dellaporta, K., Sakellariou, D., Bingham, B. S., Camilli, R., Eustice, R. M., Evagelistis, D., Ferrini, V. L., Katsaros, K., Kourkoumelis, D., et al. (2009). The 2005 Chios ancient shipwreck survey: new methods for underwater archaeology. *hesperia*, pages 269–305. 8

[Forte et al.2012] Forte, D., Gams, A., Morimoto, J., and Ude, A. (2012). On-line motion synthesis and adaptation using a trajectory database. *Robotics and Autonomous Systems*, 60(10):1327–1339. 21, 42

[Galceran Yebenes2014] Galceran Yebenes, E. (2014). *Coverage path planning for autonomous underwater vehicles*. PhD thesis, Universitat de Girona. 13

[Gancet et al.2015] Gancet, J., Urbina, D., Letier, P., Ilzokvitz, M., Weiss, P., Gauch, F., Antonelli, G., Indiveri, G., Casalino, G., Birk, A., Pfingsthorn, M. F., Calinon, S., Tanwani, A., Turetta, A., Walen, C., and Guilpain, L. (2015). DexROV: Dexterous

Undersea Inspection and Maintenance in Presence of Communication Latencies. In *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV)*. 66

[German et al.2008] German, C. R., Yoerger, D. R., Jakuba, M., Shank, T. M., Langmuir, C. H., and Nakamura, K.-i. (2008). Hydrothermal exploration with the Autonomous Benthic Explorer. *Deep Sea Research Part I: Oceanographic Research Papers*, 55(2):203–219. 8

[Gilmour2012] Gilmour, B. (2012). Field resident AUV systems: Chevrons long-term goal for Autonomous Underwater Vehicle (AUV) Development. *Oceanic Engineering Society - IEEE AUV 2012*. 9

[Goodrich and Schultz2007] Goodrich, M. A. and Schultz, A. C. (2007). Human-Robot Interaction: A Survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275. 20

[Gracias et al.2013] Gracias, N., Ridao, P., Garcia, R., Escartin, J., L'Hour, M., Cibecchini, F., Campos, R., Carreras, M., Ribas, D., Palomeras, N., et al. (2013). Mapping the Moon: Using a lightweight AUV to survey the site of the 17th century ship 'La Lune'. In *MTS/IEEE OCEANS-Bergen*, pages 1–8. IEEE. 13

[Grudic and Lawrence1993] Grudic, G. Z. and Lawrence, P. D. (1993). Iterative inverse kinematics with manipulator configuration control. *IEEE Transactions on Robotics and Automation*, 9(4):476–483. 83

[Hernández et al.2011] Hernández, E., Carreras, M., Antich, J., Ridao, P., and Ortiz, A. (2011). A topologically guided path planner for an auv using homotopy classes. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2337–2343. IEEE. 13

[Hernandez et al.2015] Hernandez, J. D., Vidal, E., Vallicrosa, G., Galceran, E., and Carreras, M. (2015). Online path planning for autonomous underwater vehicles in unknown environments. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1152–1157. IEEE. 13

[Hersch et al.2006] Hersch, M., Guenter, F., Calinon, S., and Billard, A. G. (2006). Learning Dynamical System Modulation for Constrained Reaching Tasks. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 444–449. 21, 39, 45

[Hoffmann et al.2009] Hoffmann, H., Pastor, P., Park, D.-H., and Schaal, S. (2009). Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance. In *ICRA*, pages 2587–2592. IEEE. 22, 39

[Hovland et al.1996a] Hovland, G., Sikka, P., and McCarragher, B. (1996a). Skill acquisition from human demonstration using a hidden Markov model. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 3, pages 2706–2711 vol.3. 22, 39, 45

[Hovland et al.1996b] Hovland, G., Sikka, P., and McCarragher, B. (1996b). Skill acquisition from human demonstration using a hidden Markov model. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 3, pages 2706–2711 vol.3. 24

[Hurtós Vilarnau et al.2015] Hurtós Vilarnau, N., Ribas Romagós, D., Cufí i Solé, X., Petillot, Y. R., and Salvi, J. (2015). Fourier-based registration for robust forward-looking sonar mosaicing in low-visibility underwater environments. *© Journal of Field Robotics, 2015, vol. 32, núm. 1, p. 123-151.* 13

[Hwang et al.2003] Hwang, M. J., Chung, S. Y., and Lee, D. Y. (2003). Compliant motion planning for two manipulators via human demonstration. In *Advanced Intelligent Mechatronics, 2003. AIM 2003. Proceedings. 2003 IEEE/ASME International Conference on*, volume 1, pages 423–428 vol.1. 11

[Ijspeert et al.2013] Ijspeert, A. J., Nakanishi, J., Hoffmann, H., Pastor, P., and Schaal, S. (2013). Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors. *Neural Computation*, 25(2):328–373. 26, 35, 45, 48

[Jamali et al.2015] Jamali, N., Kormushev, P., Viñas, A. C., Carreras, M., and Caldwell, D. G. (2015). Underwater robot-object contact perception using machine learning on force/torque sensor feedback. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3915–3920. 89

[Kaiser et al.2013] Kaiser, C., A., B., Camilli, R., Yoerger, D., and Jakuba, M. F. N. (2013). Hybrid AUV inspection, monitoring, and intervention of seafloor and sub-seafloor pipelines. 63

[Kalman1960] Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME – Journal of Basic Engineering*, pages 35–45. 75

[Kalman and Bucy1961] Kalman, R. E. and Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Trans. ASME, Ser. D, J. Basic Eng*, page 109. 75

[Kato and Billinghurst USA] Kato, H. and Billinghurst, M. (October 1999, San Francisco, USA.). Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System. In *2nd International Workshop on Augmented Reality (IWAR 99).* 88

[Kelley et al.2005] Kelley, D. S., Karson, J. A., Früh-Green, G. L., Yoerger, D. R., Shank, T. M., Butterfield, D. A., Hayes, J. M., Schrenk, M. O., Olson, E. J., Proskurowski, G., Jakuba, M., Bradley, A., Larson, B., Ludwig, K., Glickson, D., Buckman, K., Bradley, A. S., Brazelton, W. J., Roe, K., Elend, M. J., Delacour, A., Bernasconi, S. M., Lilley, M. D., Baross, J. A., Summons, R. E., and Sylva, S. P. (2005). A Serpentinite-Hosted Ecosystem: The Lost City Hydrothermal Field. *Science*, 307(5714):1428–1434. 8

[Konidaris et al.2012] Konidaris, G., Kuindersma, S., Grupen, R. A., and Barto, A. G. (2012). Robot learning from demonstration by constructing skill trees. *I. J. Robotic Res.*, 31(3):360–375. 21, 24, 40

[Kormushev et al.2010] Kormushev, P., Calinon, S., and Caldwell, D. G. (2010). Robot motor skill coordination with EM-based reinforcement learning. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3232–3237. IEEE. 14

[Kormushev et al.2011] Kormushev, P., Calinon, S., and Caldwell, D. G. (2011). Imitation Learning of Positional and Force Skills Demonstrated via Kinesthetic Teaching and Haptic Input. *Advanced Robotics*, 25(5):581–603. 21, 25, 42, 45

[Kormushev et al.2013] Kormushev, P., Calinon, S., and Caldwell, D. G. (2013). Reinforcement Learning in Robotics: Applications and Real-World Challenges. *Robotics*, 2(3):122–148.

[Kronander and Billard2012] Kronander, K. and Billard, A. (2012). Online learning of varying stiffness through physical human-robot interaction. In *ICRA*, pages 1842–1849. IEEE. 21, 25, 42

[Kruger et al.2010] Kruger, V., Herzog, D., Baby, S., Ude, A., and Kragic, D. (2010). Learning Actions from Observations. *Robotics Automation Magazine, IEEE*, 17(2):30–43. 22, 41, 53

[Krüger et al.2012] Krüger, V., Tikhanoff, V., Natale, L., and Sandini, G. (2012). Imitation learning of non-linear point-to-point robot motions using dirichlet processes. In *ICRA*, pages 2029–2034. IEEE. 21, 42

[KUKA-Robotics2016] KUKA-Robotics (2016). LWR iiwa. `http://www.kuka-robotics.com/es/products/industrial_robots/sensitiv/`. 10

[Kunz et al.2009] Kunz, C., Murphy, C., Singh, H., Pontbriand, C., Sohn, R. A., Singh, S., Sato, T., Roman, C., Nakamura, K.-i., Jakuba, M., et al. (2009). Toward extraplanetary under-ice exploration: Robotic steps in the Arctic. *Journal of Field Robotics*, 26(4):411–429. 8

[Lab2016] Lab, S. R. (2016). OceanOne ROV. `http://cs.stanford.edu/group/manips/ocean-one.html`. 67

[Lane et al.1997] Lane, D., O'Brien, D. J., Pickett, M., Davies, J., Robinson, G., Jones, D., Scott, E., Casalino, G., Bartolini, G., Cannata, G., Ferrara, A., angeletti, D., Veruggio, G., Bono, R., Virgili, P., Canals, M., Pallas, R., Garcia, E., and Smith, C. (1997). AMADEUS-Advanced Manipulation for Deep Underwater Sampling. *IEEE Robotics and Automation Magazine*, 4(4):34–45. 58

[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 IFAC NGCUV 2012 - Navigation, Guidance and Control of Underwater Vehicles*. 63

[Latombe1991] Latombe, J.-C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, USA. 11

[Lei and Ming2017] Lei, T. and Ming, L. (2017). Deep-learning in Mobile Robotics - from Perception to Control Systems: A Survey on Why and Why not. *arXiv preprint arXiv:1612.07139*. 19

[Lowe2004] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60(2):91–110. 73

[MacQueen1967] MacQueen, J. B. (1967). Some Methods for Classification and Analysis of MultiVariate Observations. In Cam, L. M. L. and Neyman, J., editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press. 33

[Mallios et al.2014a] Mallios, A. et al. (2014a). *Sonar scan matching for simultaneous localization and mapping in confined underwater environments*. PhD thesis, Universitat de Girona. 13

[Mallios et al.2014b] Mallios, A., Ridao, P., Ribas, D., and Hernández, E. (2014b). Scan matching SLAM in underwater environments. *Autonomous Robots*, 36(3):181–198. 13

[Manschitz et al.2014] Manschitz, S., Kober, J., Gienger, M., and Peters, J. (2014). Learning to sequence movement primitives from demonstrations. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 4414–4421. 40

[Manschitz et al.2015] Manschitz, S., Kober, J., Gienger, M., and Peters, J. (2015). Learning movement primitive attractor goals and sequential skills from kinesthetic demonstrations. *Robotics and Autonomous Systems*, 74, Part A:97–107. 21, 25, 28

[Marani et al.2009] Marani, G., Choi, S., and Yuh, J. (2009). Underwater Autonomous Manipulation for Intervention Missions AUVs. *Ocean Engineering. Special Issue: AUV*, 36(1):15–23. 58, 61

[Matsubara et al.2011] Matsubara, T., Hyon, S.-H., and Morimoto, J. (2011). Learning parametric dynamic movement primitives from multiple demonstrations. *Neural Networks*, 24(5):493–500. 41, 53

[MERBOTS2016] MERBOTS (2016). MERBOTS Project. `http://www.irs.uji.es/merbots`. 65

[Metta et al.2008] Metta, G., Sandini, G., Vernon, D., Natale, L., and Nori, F. (2008). The iCub humanoid robot: an open platform for research in embodied cognition. In *Proceedings of the 8th workshop on performance metrics for intelligent systems*, pages 50–56. ACM. 42

[Nemec et al.2013] Nemec, B., Gams, A., and Ude, A. (2013). Velocity adaptation for self-improvement of skills learned from user demonstrations. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 423–428. 52

[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. 13

[Paduan et al.2009] Paduan, J. B., Caress, D. W., Clague, D. A., Paull, C. K., and Thomas, H. (2009). High-resolution mapping of mass wasting, tectonic, and volcanic hazards using the MBARI Mapping AUV. In *International Conference on seafloor mapping for geohazard assessmen*, volume 7, pages 181–186. 8

[Palomer et al.2016] Palomer, A., Ridao, P., and Ribas, D. (2016). Multibeam 3D Underwater SLAM with Probabilistic Registration. *Sensors*, 16(4):560. 13

[Palomeras et al.2006] Palomeras, N., Carreras, M., Ridao, P., and Hernandez, E. (2006). Mission control system for dam inspection with an AUV. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2551–2556. IEEE. 13

[Palomeras et al.2012] Palomeras, N., El-Fakdi, A., Carreras, M., and Ridao, P. (2012). COLA2: A control architecture for AUVs. *IEEE Journal of Oceanic Engineering*, 37(4):695–716. 13

[Palomeras et al.2013] Palomeras, N., Nagappa, S., Ribas, D., Gracias, N., and Carreras, M. (2013). Vision-based localization and mapping system for AUV intervention. In *OCEANS'13 MTS/IEEE*. 63, 75

[Palomeras et al.2014a] Palomeras, N., Penalver, A., Massot-Campos, M., Vallicrosa, G., Negre, P., Fernandez, J., Ridao, P., Sanz, P., Oliver-Codina, G., and Palomer, A. (2014a). I-AUV docking and intervention in a subsea panel. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2279–2285. 63

[Palomeras et al.2014b] Palomeras, N., Penalver, A., Massot-Campos, M., Vallicrosa, G., Negre, P. L., Fernández, J. J., Ridao, P., Sanz, P. J., Oliver-Codina, G., and Palomer, A. (2014b). I-AUV docking and intervention in a subsea panel. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2279–2285. IEEE. 13

[Prados et al.2014] Prados, R., García, R., and Neumann, L. (2014). *Image blending techniques and their application in underwater mosaicing*. Springer. 13

[Prats et al.2011a] Prats, M., García, J., Fernandez, J., Marín, R., and Sanz, P. (2011a). Advances in the specification and execution of underwater autonomous manipulation tasks. *IEEE Oceans 2011*, pages 10361–10366. 58

[Prats et al.2012a] Prats, M., García, J., Wirth, S., Ribas, D., Sanz, P., Ridao, P., Gracias, N., and Oliver, G. (2012a). Multipurpose autonomous underwater intervention: A systems integration perspective. *Control & Automation (MED), 2012 20th Mediterranean Conference on*, pages 1379–1384. 58

[Prats et al.2012b] Prats, M., Perez, J., Fernandez, J., and Sanz, P. (2012b). 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. 100

[Prats et al.2011b] Prats, M., Ribas, D., Palomeras, N., García, J. C., Nannen, V., Wirth, S., Fernández, J. J., Beltrán, J. P., Campos, R., Ridao, P., Sanz, P. J., Oliver, G., Carreras, M., Gracias, N., Marín, R., and Ortiz, A. (2011b). Reconfigurable AUV for intervention missions: a case study on underwater object recovery. *Intelligent Service Robotics*, 5. To be published. 61

[Rethink-Robotics2016] Rethink-Robotics (2016). Baxter Robot. `http://www.rethinkrobotics.com/products/baxter`. 10

[Reza Ahmadzadeh et al.2013] Reza Ahmadzadeh, S., Kormushev, P., and Caldwell, D. G. (2013). Autonomous Robotic Valve Turning: A Hierarchical Learning Approach. In *IEEE Intl. Conf. on Robotics and Automation (ICRA 2013), Karlsruhe, Germany*. 52

[Ribas et al.2007] Ribas, D., Palomeras, N., Ridao, P., Carreras, M., and Hernandez, E. (2007). Ictineuauv wins the first sauc-e competition. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 151–156. IEEE. 13

[Ribas et al.2012] Ribas, D., Palomeras, N., Ridao, P., Carreras, M., and Mallios, A. (2012). Girona 500 AUV: From Survey to Intervention. *Mechatronics, IEEE/ASME Transactions on*, 17(1):46–53. 13, 61, 69

[Ribas et al.2008] Ribas, D., Ridao, P., Tardós, J., and Neira, J. (2008). Underwater SLAM in Man Made Structured Environments. *Journal of Field Robotics*, 25(11-12):898–921. 13

[Ridao et al.2002] Ridao, P., Batlle, J., and Carreras, M. (2002). O² CA², a new object oriented control architecture for autonomy: the reactive layer. *Control Engineering Practice*, 10(8):857–873. 13

[Rigaud et al.1998] Rigaud, V., Coste-Maniere, E., Aldon, M., Probert, P., Perrier, M., Rives, P., Simon, D., Lang, D., J., K., Casal, A., Amar, J., Dauchez, P., and Chantler, M. (1998). UNION: underwater intelligent operation and navigation. *Robotics & Automation Magazine, IEEE*, 5(1):25–35. 58, 59

[ROBUST2016] ROBUST (2016). ROBUST Project. `http://eu-robust.eu`. 66

[Rosell et al.1999] Rosell, J., Basaniz, L., and Suarez, R. (1999). Compliant-motion planning and execution for robotic assembly. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 4, pages 2774–2779 vol.4. 11

[Rosten and Drummond2006] Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *In European Conference on Computer Vision*, pages 430–443. 73

[Rublee et al.2011] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. 73

[Sacks2003] Sacks, E. (2003). Path planning for planar articulated robots using configuration spaces and compliant motion. *IEEE Transactions on Robotics and Automation*, 19(3):381–390. 11

[Sanz et al.2012] Sanz, P., Marín, R., Sales, J., Oliver, G., and Ridao, P. (2012). Recent advances in underwater robotics for intervention missions. Soller harbor experiments. *Low Cost Books, Spain*. 58, 62

[Schaal and Atkeson1997] Schaal, S. and Atkeson, C. G. (1997). Constructive Incremental Learning From Only Local Information. *Neural Computation*, 10:2047–2084. 37

[Shihavuddin et al.2013] Shihavuddin, A., Gracias, N., Garcia, R., Gleason, A. C., and Gintert, B. (2013). Image-based coral reef classification and thematic mapping. *Remote Sensing*, 5(4):1809–1841. 13

[Silver et al.2012] Silver, D., Bagnell, J. A., and Stentz, A. (2012). Active learning from demonstration for robust autonomous navigation. In *ICRA*, pages 200–207. IEEE. 23

[Toris et al.2012] Toris, R., Suay, H. B., and Chernova, S. (2012). A practical comparison of three robot learning from demonstration algorithms. In Yanco, H. A., Steinfeld, A., Evers, V., and Jenkins, O. C., editors, *HRI*, pages 261–262. ACM. 20

[Ude et al.2010] Ude, A., Gams, A., Asfour, T., and Morimoto, J. (2010). Task-Specific Generalization of Discrete and Periodic Dynamic Movement Primitives. *IEEE Transactions on Robotics*, 26(5):800–815. 21

[Ude et al.2014] Ude, A., Nemec, B., Petrić, T., and Morimoto, J. (2014). Orientation in Cartesian space dynamic movement primitives. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2997–3004. 52

[Universal-Robots2017] Universal-Robots (2017). UR5 Robot. `https://www.universal-robots.com/products/ur5-robot`. 10

[Vallicrosa et al.2014] Vallicrosa, G., Ridao, P., Ribas, D., and Palomer, A. (2014). Active range-only beacon localization for auv homing. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2286–2291. IEEE. 13

[Wang et al.1995] Wang, H. H., Rock, S. M., and Lees, M. J. (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. 58

[Williams et al.2010] Williams, S. B., Pizarro, O., Jakuba, M., and Barrett, N. (2010). AUV benthic habitat mapping in south eastern Tasmania. In *Field and Service Robotics*, pages 275–284. Springer. 8

[Wilson and Bobick1999] Wilson, A. D. and Bobick, A. F. (1999). Parametric Hidden Markov Models for Gesture Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(9):884–900. 21, 41

[Yoerger et al.1998] Yoerger, D. R., Bradley, A. M., Walden, B. B., Singh, H., and Bachmayer, R. (1998). Surveying a subsea lava flow using the Autonomous Benthic Explorer (ABE). *International Journal of Systems Science*, 29(10):1031–1044. 8

[Yoo et al.2015] Yoo, S.-y., Jun, B.-H., Shim, H., and Lee, P.-m. (2015). Design and analysis of carbon fiber reinforced plastic body frame for multi-legged subsea walking robot, Crabster. *Ocean Engineering*, 102:78–86. 67

[Zandara et al.2013] Zandara, S., Ridao, P., Ribas, D., Mallios, A., and Palomer, A. (2013). Probabilistic surface matching for bathymetry based slam. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 40–45. IEEE. 13

Declaration:

I hereby declare that his thesis contains no material which has been accepted for the award of any other degree or diploma in any university. To the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference has been made.

Girona, May 2017

Arnau Carrera Viñas