EDINBURGH CENTRE FOR ROBOTICS

University of Edinburgh

Heriot-Watt University

Learning and Generalisation of Primitive Skills for Robust Dual-arm Manipulation

MSc BY RESEARCH THESIS

Author:
Èric Pairet

Supervisors:
Dr. Frank Broz
Dr. Michael Mistry

A thesis submitted for the degree of MSc by Research

in the

Edinburgh Centre for Robotics

August, 2018

EDINBURGH CENTRE FOR ROBOTICS
UNIVERSITY OF EDINBURGH
HERIOT-WATT UNIVERSITY

Abstract

Learning and Generalisation of Primitive Skills for Robust Dual-arm Manipulation

by Èric Pairet

Robots are becoming a vital ingredient in society. Some of their daily tasks require dual-arm manipulation skills in the rapidly changing, dynamic and unpredictable real-world environments where they have to operate. Given the expertise of humans in conducting these activities, it is natural to study humans motions to use the resulting knowledge in robotic control. With this in mind, this work leverages human knowledge to formulate a more general, real-time, and less task-specific framework for dual-arm manipulation. Particularly, the proposed architecture first learns the dynamics underlying the execution of different primitive skills. These are harvested in a one-at-a-time fashion from human demonstrations, making dual-arm systems accessible to non-roboticists-experts. Then, the framework exploits such knowledge simultaneously and sequentially to confront complex and novel scenarios.

Current works in the literature deal with the challenges arising from particular dual-arm applications in controlled environments. Thus, the novelty of this work lies in (i) learning a set of primitive skills in a one-at-a-time fashion, and (ii) endowing dual-arm systems with the ability to reuse their knowledge according to the requirements of any commanded task, as well as the surrounding environment. The potential of the proposed framework is demonstrated with several experiments involving synthetic environments, the simulated and real iCub humanoid robot. Apart from evaluating the performance and generalisation capabilities of the different primitive skills, the framework as a whole is tested with a dual-arm pick-and-place task of a parcel in the presence of unexpected obstacles. Results suggest the suitability of the method towards robust and generalisable dual-arm manipulation.

Declaration of Originality

I Èric Pairet declare that this dissertation is my own original work that is being submitted to Heriot-Watt University, Scotland in partial of the Degree of Master of Science in Robotics and Autonomous Systems. I acknowledge that the original work that is being submitted to Heriot-Watt University has properly been cited and referenced. Some elements of this work may have already been submitted to Heriot-Watt University as part of the dissertation preparatory work under Robotics Research Report (B31AP) and/or Robotics Research Proposal (B31AT). It has not been submitted to any other university or institute of higher learning.

Signature: 

Èric Pairet

17th of August 2018

Acknowledgments

The work presented throughout this manuscript becomes a reality with the unconditional help and support of many incredible people. Their continuous encouragement and discussions have been essential to complete this work. I would like to give my gratitude to all of them.

Foremost, my most profound gratitude to my supervisors Dr Frank Broz and Dr Michael Mistry, for their time, dedication and invaluable contribution to this thesis. Their advice has been there whenever it was needed, and they have always ensured that this work was coming to fruition.

I would like to extend my sincere thanks to the members of the Robotics Lab in the Heriot-Watt University, especially Ingo, Eli, and Dr Katrin, for the countless fruitful discussions and guidance. I would also like to thank Bence and Josh for the many ideas they have given me. Even the most spontaneous conversation in a coffee break has positively influenced on this work.

I would not have the privilege of writing these words without the support of Prof. Yvan Petillot. I am deeply indebted to him, for bringing me the chance of joining the Edinburgh Centre for Robotics, for his constant guidance, commitment and faith in me.

Finally, the warmest gratitude to my family and friends. Despite the distance, they are always there to guide me through important decisions, looking after me, and ready to enjoy a beer when the chance comes. Last but not least, to Paola for being there in the good and bad times, and for the insatiable source of patience she is to discuss and peer-review my work. Their all daily encouragement and moral support have always pushed me to give my best.

Contents

List of figures iv

List of tables vii

Acronyms viii

1 Introduction 1

1.1 Context . . . 1
1.2 Motivation . . . 2
1.3 Objectives and Scope . . . 3
1.4 Research Impact . . . 4

2 Related Work 5

2.1 Motion Adaptation Against Obstacles . . . 5
2.2 Frameworks for Generalisable Manipulation . . . 7
2.3 Discussion . . . 8

3 System Requisites and Modelisation 9

3.1 Learning for a Dual-arm Manipulator . . . 9
3.2 Dual-arm Primitive Skills Taxonomy . . . 10
3.3 Dual-arm System Modelisation . . . 10
3.3.1 Positional Dynamics . . . 11
3.3.2 Orientational Dynamics . . . 12
3.3.3 Coupling Terms . . . 12
3.4 Dual-arm Grasping Geometry . . . 14

4 Learning Primitive Skills 15

4.1 Goal-oriented Dynamics . . . 15
4.2 Obstacle Avoidance . . . 17

4.3 Force Interaction . . . 19

5 Framework for Robust Dual-arm Manipulation 20

5.1 Learning Module . . . 21
5.2 Roll-out Module . . . 21
5.3 Evaluation Module . . . 22

6 Results and Evaluation 24

6.1 Experimental Setup . . . 24
6.1.1 iCub Humanoid Robot . . . 25
6.1.2 Pick-and-Place Showcase . . . 26
6.1.3 Workspace and Manipulability Analysis . . . 26
6.1.4 Demonstration Recording and Learning . . . 29
6.1.5 Framework Deployment on iCub Humanoid . . . 31
6.2 Goal-oriented Skill . . . 33
6.2.1 Positional Dynamics . . . 33
6.2.2 Orientational Dynamics . . . 34
6.3 Obstacle Avoidance Skill . . . 35
6.4 Framework Evaluation . . . 37
6.4.1 Evaluation on Synthetic Environments . . . 37
6.4.2 Evaluation on a Simulated iCub Humanoid . . . 40

7 Final Remarks and Future Work 43

A Apprenticeship Learning: A Survey 45

B iCub Kinematics 58

B.1 Physical Platform . . . 58
B.1.1 Arms Constitution . . . 59
B.2 Software Architecture . . . 60
B.2.1 Arms Control . . . 61

Bibliography 63

List of Figures

3.1 Dual-arm manipulator modelled as a closed-chain system. Its dynamics are approximated to those of a spring-damper system actuating in the Cartesian space. 11

3.2 Skill of drawing the letter G represented in the force level according to a spring-damper system modelisation. (a) Resulting G-shapes on a two-dimensional (2D) plane. (b)-(c) External forces in the x_1 and x_2 dimensions, respectively. The dashed line is a zero-force reference. 13

4.1 dynamic movement primitive (DMP)-based modelisation and generalisation of G-shapes on a 2D plane. (a) Given demonstration (polka dotted trajectory), learnt G-shape (red trajectory) and generalisaton from different start and goal positions (blue and green trajectories). (b)-(c) Learnt dynamics (red line) in the x_1 dimension. They are the result of a weighted combination of ten radial basis function (RBF). The corresponding weights are the ten segments in (b), and the set of RBF are depicted in (c). 16

4.2 Obstacle avoidance primitive skill proposed in [Fajen and Warren, 2003]. (a) Manipulated object (brown prism) and obstacle (grey circle). (b) Change of steering angle $\dot{\theta}$ of the original formulation in Equation (4.5) with $\gamma = 1000$ and $\beta = 20/\pi$. 17

4.3 Change of steering angle $\dot{\theta}$ and dead zone issue. (a) Absolute representation of Figure 4.2b (black curve), and the proposed alternative in Equation (4.6) with $a = 66.07$, $c = 0.4732$ and $k = 0$ (red curve). (b) Following the same colour code, both methods confronting an obstacle (grey circle) in a 2D environment. The original formulation does not react against imminent collision, instead the proposed alternative provides a smooth and coherent behaviour. 18

5.1 Scheme of the three stages involved in the proposal. Learning: a human demonstrator teaches some primitives behaviours to a system. Roll-out: the robot exploits (generalises and combines accordingly to the environment awareness) the acquired knowledge. Evaluation: an evaluator inspects the system's performance and decides whether reteaching is necessary. 20

6.1 iCub humanoid robot. 25

6.2 Pick-and-place of a parcel (brown prism) in the presence of obstacles (grey prism). 26

6.3 iCub's left (top row) and right (bottom row) end-effector's workspaces. From left to right: left, front, right and top view. 27

6.4 Analysis of iCub's dual-arm workspace contrained by pick-and-place task requirements. Top row: box plot reflecting the constrained workspace subject to different parcel widths d. Second row: workspace for parcel width $d = 100 \pm 5mm$. Bottom row: parcel width $d = 250 \pm 5mm$. From left to right: left, front, right and top view. All presented information is also contrained by the aforementioned orientation error of 0 ± 3 degrees. 28

6.5 iCub's dual-arm workspace with external management of the torso's degree of freedom (DoF) roll. Aforementioned pick-and-place task constraints apply: parcel width $d = 250 \pm 5mm$ and orientation error of 0 ± 3 degrees. From left to right: left, front, right and top view. 29

6.6 Experimental setup of the pick-and-place task and the iCub humanoid in Gazebo. 31

6.7 Layout of the framework deployment on the iCub humanoid. Note: inverse kinematics (IK), grasping geometry (GG), primitive skill (PS). 32

6.8 DMP generalisation capabilities. Given a demonstration (red trajectory), rolling-out the model in Equation (3.5)-(3.6) with a DMP as coupling term $\mathbf{f}_{o_x}(\cdot)$ lets the system move the box (brown prism) to new goal states by mean of dynamics generalisation (blue trajectories). 34

6.9 Orientational dynamics consisting on rotating a free-floating parcel from the most left configuration $e_s = [0 \;\; 0 \;\; 0]^T$ degrees to the most right configuration $e_g = [90 \;\; 90 \;\; 0]^T$ degrees. Note that no rotation is executed at the first and last part of the demonstration. 34

6.10 Framework's orientational capabilities analysis. Demonstrated orientational dynamics (red lines), inference to undemonstrated orientations in the range ± 25 degrees around the demonstration (blue lines). (a)-(d) Profile of each variable of the quaternion $q = [w \;\; x \;\; y \;\; z]^T$. 35

6.11 iCub humanoid robot learning the primitive skill of obstacle avoidance with two different behaviours: reckless (first column) and convervative (second column). (a)-(b) Human demonstrations to avoid an obstacle (red sphere). (c)-(d) iCub's proprioception data. (e)-(f) Processed proprioception data (red trajectory) and learnt behaviour (blue trajectory). 36

6.12 Generalisation capabilities to multiple obstacles and in three-dimensional (3D) scenarios of the learnt reckless (magenta trajectory) and conservative (green trajectory) obstacle avoidance behaviours. 36

6.13 Dual-arm pick-and-place of a parcel (brown prism) in the presence of obstacles (grey prism). Demonstrated task (red trajectory), infered task (blue trajectory), infered task with obstacle avoidance (black trajectory). The composition of primitive skills lets the system generalise to unfamiliar environments. (a) Perspective, (b) lateral, (c) top, and (d) front view. 38

6.14 Framework analysis in Figure 6.13 scenario. System's natural dynamics (dashed lines), demonstrated task (red lines), infered task (blue lines), obstacle avoidance (green lines), and infered task with obstacle avoidance (black lines). First column: primitive skills at the force level. Second column: forces affect at the Cartesian space. Top to bottom: x, y and z-axis. 39

6.15 iCub humanoid robot exploiting the demonstrated pick-and-place task (green trajectory) to succeed (blue trajectory) in *Scenario-1* which has an obstacle (red sphere) at $x_o = [0.3 \;\; 0 \;\; 0.6]^T$ metres. (a) Parcel initial state, (b)-(d) grasping parcel laterally, (e)-(g) simultaneously exploiting some primitive skills to successfully conduct the pick-and-place task in an undemonstrated scenario, and (h) overview of the trajectory adapted in real-time. 40

6.16 iCub humanoid robot exploiting the demonstrated pick-and-place task (green trajectory) to succeed (blue trajectory) in *Scenario-2* which has an obstacle (red sphere) at $x_o = [0.25 \;\; 0 \;\; 0.55]^T$ metres. (a) Parcel initial state, (b)-(d) grasping parcel laterally, (e)-(g) simultaneously exploiting some primitive skills to successfully conduct the pick-and-place task in an undemonstrated scenario, and (h) overview of the trajectory adapted in real-time. 41

B.1 iCub's composition and reference frames. (a) iCub's global reference frame, (b) iCub's kinematic tree, (c) reference frames of iCub's joints. 58

B.2 iCub's composition and reference frames. (a) iCub's global reference frame, (b) iCub's kinematic tree, (c) reference frames of iCub's joints. 59

B.3 iCub's end-effector kinematic chain. (a) Right arm and (b) left arm. 60

List of Tables

6.1 Parametrisation of the closed-chaing dual-arm system modelled in Section 3. Note: positional dynamics (PD), orientational dynamics (OD). 30

6.2 Parametrisation of the goal-oriented skill dynamics modelled in Section 4.1. Note: to be learnt (TBL). 30

6.3 Parametrisation of the obstacle avoidance behaviour modelled in Section 4.2. Note: obstacle avoidance (OA), to be learnt (TBL). 30

6.4 Parametrisation of the force interaction skill modelled in Section 4.3. Note: to be learnt (TBL). 30

6.5 Summary of start and goal configurations (3D position and Euler XYZ orientation) of the demonstrated and unfamiliar scenarios reported in Figure 6.13. . . . 38

B.1 Denavit-Hartenberg paramters for iCub's right end-effector. The first three links are from the torso. The last seven links are from the right arm. 61

B.2 Denavit-Hartenberg paramters for iCub's left end-effector. The first three links are from the torso. The last seven links are from the left arm. 61

Acronyms

2D two-dimensional

3D three-dimensional

AI artificial intelligence

AI-HRI Artificial Intelligence for Human-Robot Interaction

CAN controller area network

DMP dynamic movement primitive

DoF degree of freedom

FK forwad kinematics

HRI human-robot interaction

IIT Italian Institute of Technology

IK inverse kinematics

ILC iterative learning control

KL Kullback-Leibler

LbD learning by demonstration

LMS least mean squares

RBF radial basis function

RL reinforcement learning

ROS robot operating system

YARP yet another robotic platform

Chapter 1

Introduction

The last decades have witnessed a drastic increase in the use of robots in industry, professional and domestic environments. Among the countless competences that robots have acquired, some of the most outstanding are automating repetitive and exhausting tasks in manufacturing plants, working in hazardous scenarios unreachable to humans, assisting doctors in challenging surgical operations, and taking responsibility for household chores. In the achievement of these promising capabilities, biologically-inspiring the design of the morphological and behavioural aspects of robots has played a significant role [Pfeifer et al., 2007].

1.1 Context

In an attempt to confer robots with more human-like capabilities, dual-arm anthropomorphic manipulation has become an important research topic in the robotics community [Smith et al., 2012]. Bi-manual arrangements extend the systems competences to efficiently perform tasks that involve manipulating large objects and ensemble multi-component elements without the need for external assistance. All these tasks require an accurate synchronisation between arms to avoid breaking or exposing the handled object to stress.

Traditional approaches, such as control and planning-based methods, governing these dual-arm systems depend upon an excellent understanding of the model underlying the systems behaviour [Smith et al., 2012]. Even though deriving an accurate model is possible for some complex systems, approximations are commonly used to make the calculations computationally tractable, despite the trade-off of the models uncertainty [Pairet et al., 2018]. Furthermore, some of these methods lack scalability and generalisation capabilities along and across tasks:

hand-defining all possible scenarios, movements, behaviours, and extensive manual tuning of the systems control architecture might be required [Billard et al., 2008]. Therefore, they need an expert programmer and usually involve high computational resources [Argall et al., 2009].

The growth of artificial intelligence (AI) has popularised more natural techniques for robot learning, reducing the laborious task of coding every possible scenario and thus, increasing modularity and flexibility on the systems. This allows non-robotics-experts to interact, teach and modify the robots behaviours [Nicolescu and Mataric, 2003], and, consequently, to obtain more human-like behaviours with enhanced acceptability and compatibility to the human workspaces [Ajoudani et al., 2017]. In an attempt for these systems to work in a more human-like manner, they are expected to learn from (and as) humans, especially when learning motions, i.e. the kinematics, dynamics and constraints describing a task.

In the recent years, adopting human knowledge for the robot control has shown an incredible performance in a wide range of robotic tasks. Despite the encouraging possibilities offered by the learning realm, teaching complex systems, such as dual-arm manipulators, to respond and adapt to a broad case of scenarios is yet an unsolved challenge.

1.2 Motivation

Given the expertise and dexterity of humans in using both arms for manipulation purposes, it is natural to study humans motions to use the resulting knowledge in robotic control. In this context, imitation learning or learning by demonstration (LbD) has shown to be a promising alternative to let robots learn from human demonstrations [Argall et al., 2009]. LbD is a supervised learning-based technique which allows transferring knowledge from a human expert to a machine, rather than manually programming the desired behaviour.

Teaching a robot from human demonstrations can be challenging. The different anatomical characteristics between the teacher and the learner produce the correspondence problem, i.e. the issue of identifying a mapping between the teacher and the learner which allows transferring of information from one to the other [Dautenhahn and Nehaniv, 2002]. Moreover, complex motions involve a mixture of human intentions, which are difficult to learn when following an all-at-once learning baseline [Bajcsy et al., 2018]. On top of that, teaching a dual-arm system can suppose a high endeavour for non-robotics-experts [Akgun et al., 2012].

Learning by demonstration offers some generalisation capabilities, yet limited to similar scenarios as the demonstrated one [Billard et al., 2008]. This restriction is not realistic to the rapidly

changing, dynamic and unpredictable environments where robots have to operate. Extended robustness can be obtained by letting the system to improve and adapt the learnt task to new scenarios iteratively [Guenter et al., 2007]. This leads to the well-known exploration-exploitation dilemma and comes at the cost of needing to fail to learn and consequently, at the risk of causing harm to the robot during the self-learning process [Pairet and Broz, 2018].

The recent trend on imitation learning has taken inspiration from the behavioural and neuroscientific processes of animal imitation [Pastor et al., 2009]. This paper presents a neurobiologically-inspired framework that seeks to jointly overcome the aforementioned issues, namely (i) the complex and ambiguous teaching procedures and (ii) the limited generalisation capabilities.

1.3 Objectives and Scope

The main goal of this thesis is to develop a framework which endows a dual-arm system with a more general and less task-specific method for real-time and robust manipulation in unfamiliar environments. The proposed framework (i) leverages human knowledge to create a library of primitive skills, which are learnt one-at-a-time from human demonstrations, and (ii) endows dual-arm systems with human-like manipulation capabilities by combining (sequentially and simultaneously) the primitive skills. Thus, during this dissertation, the objectives are:

- Give an overview and discuss the state-of-the-art on learning-based algorithms and frameworks which endow a system with generalisable manipulation skills (see Chapter 2).
- Analyse the requirements arising from this work's objective and model the system, so it meets the learning, modularity and dual-arm compatibility requisites (see Chapter 3).
- Establish the fundamentals to learn different primitive skills, such as goal-oriented (position and orientation), obstacle avoidance and force interaction purposes (see Chapter 4).
- Formulate a high-level framework manager which integrates the previously designed components while taking into account the requirements of a dual-arm system (see Chapter 5).

In the scope of this thesis, the potential of the proposed framework is demonstrated with a set of experiments involving synthetic environments, the simulated and real iCub humanoid robot. The showcase is a dual-arm pick-and-place task of a parcel in the presence of unexpected obstacles. This experimental evaluation and its required setup are detailed in Chapter 6. The modularity of the proposal allows exploiting objects and environmental semantic features to broaden its generalisation capabilities in front of a wide variety of scenarios. Although this feature remains out of this thesis scope, it is an interesting direction for future work (see Chapter 7).

1.4 Research Impact

The main contribution of this work is the formulation of a framework which (i) learns primitive skills from human demonstrations in a one-at-a-time fashion, thus easing the complexity and ambiguity involved in the human-robot teaching procedures, and (ii) exploits the acquired knowledge for robust and generalisable dual-arm manipulation purposes in novel environments. Such an architecture extends the capabilities of the method presented in [Pastor et al., 2009] to handle the requirements of a dual-arm system. This leads to a framework which reuses its knowledge to generalise its behaviour accordingly the environment awareness, differently from the current state-of-the-art architectures for dual-arm manipulation which are limited to the highly controlled industrial environments [Topp, 2017; Zöllner et al., 2004].

Alongside the novelty of the proposed framework, this work also contributes to the field of learning coupling terms for obstacle avoidance. This skill initially proposed for single-arm manipulation in [Hoffmann et al., 2009] and later improved in [Rai et al., 2014, 2017], is further enhanced by (i) reformulating it as a bell-shaped function to address its dead zone issue, (ii) learning its behaviour from human demonstrations, and (iii) using it in the dual-arm context.

Chapter 2

Related Work

Strategies which let robots autonomously perform a wide range of tasks in unstructured environments have always been of great interest in the robotics community. State-of-the-art techniques aiming to endow robots with these capabilities have mainly been presented under the pure control theory [Bristow et al., 2006a; Nguyen-Tuong and Peters, 2011; Schaal and Atkeson, 2010] and the learning realm [Argall et al., 2009; Goodrich and Schultz, 2007; Kober et al., 2013; Taylor and Stone, 2009]. Alternatively, a growing area of research exploits the advantages of hybrid learning techniques. An extensive review of this field called apprenticeship learning is under preparation as result of this year's research [Pairet and Broz, 2018] (see Appendix A).

Avoiding to repeat such a generic an extensive state-of-the-art review, this chapter exclusively presents those works in the literature which are strongly relevant to the contributions of this manuscript: (i) techniques which adjust a learnt motion to overcome unexpected obstacles in the trajectory, and (ii) frameworks which re-use, combine, and/or sequentially exploit a set of primitive skills to create complex behaviours. For the sake of completeness, this review considers not only dual-arm systems but also remarkable works involving single-arm manipulators. A final discussion wraps up the pros and cons of all overviewed methods from the literature.

2.1 Motion Adaptation Against Obstacles

In the learning realm, input datasets are considered to be exemplary demonstrations of the desired behaviour. The reality, however, is that datasets might show poor robot-human correspondence, suboptimal performance, ambiguous examples, or lack the actions to take in certain states [Argall et al., 2009]. For that reason, being able to adapt the learnt behaviour to overcome

these issues is critical. This section focuses on the state-of-the-art works that shape motions in a new environment, particularly in the presence of unexpected obstacles. Three main approaches are identified: (i) reward executions, (ii) advise actions, and (iii) adapt online.

The strategy behind rewarding the executed motions lies in using reinforcement learning (RL) algorithms to explore in the parameter space a set of weights that minimise a user-defined reward function. As discussed in [Pairet and Broz, 2018], such an approach needs many iterations to converge to a successful motion. Thus, its use has been primarily for enhancing suboptimal demonstrations. Guenter et al. needed approximately 3,000 iterations to successfully reshape the parameters of an initially learnt policy for a 4 degrees of freedom (DoFs) arm. The manipulator had to grasp objects in arbitrary positions and put them into a box, even with the presence of unseen obstacles in the middle of the demonstrated trajectories [Guenter et al., 2007]. This approach's limitation is that each new scenario involves retraining the model from the base.

Alternatively, the system can be advised on how to proceed in unfamiliar scenarios. One option is using iterative learning control (ILC), which imitates the ability of humans to quickly re-adapt to new situations. ILC consists in feed-forwarding the committed error in the current trial into the next one [Norrlof, 2002]. Bristow et al. reviews the many ILC-based works that have succeeded re-adapting motions. However, it also states that first ILC iterations still involve a significant error, and this cannot be afforded in many real-world robotics applications [Bristow et al., 2006b]. Thus, a natural alternative which does not require to iterate is providing the system with more demonstrations. Stulp et al. provided a total of 55 demonstrations to obtain a probabilistic representation of a pick-and-place task over an obstacle of varying height [Stulp et al., 2013]. Even though the performance of this procedure is exemplary, providing such an amount of demonstrations is usually extremely time-consuming.

The last strategy to adapt motions to new scenarios is doing it in real-time. Online motion adaptation has been extensively studied in motion planning. In this field, Khatib presented the well-known potential fields [Khatib, 1986], later reformulated in [Park et al., 2008] to adjust the output behaviour accordingly to the relative agent-obstacle velocity and heading. This dynamic approach became an excellent source of inspiration for the biologically-inspired obstacle avoidance formulated in [Fajen and Warren, 2003], which makes a robot steer around an obstacle in the same manner as humans do. This strategy was later validated in a pick-and-place task in the presence of obstacles using a single-arm setup [Hoffmann et al., 2009; Rai et al., 2014, 2017]. Analogously to potential fields, this analytical approach emulates the modelling of obstacles as sources of repulsive forces, but without the curse of local minima and the high computational expenses. On the downside, its modelling capabilities are limited to point-like obstacles.

2.2 Frameworks for Generalisable Manipulation

Endowing robots with the ability of adapting to novel scenarios is essential. However, it is equally important to provide them with the capability of grasping, holding, moving and rotating an object, among others. This lets robots to use this knowledge to perform a wide range of tasks in less structured environments. To this aim, the past years have seen a growing interest in developing learning-based frameworks. These architectures are usually end-to-end applications, which deal with the acquisition of data from human demonstrations, learn from such data, and exploit such knowledge to carry out the commanded task.

Zöllner et al. presented a framework which learns from human demonstrations the sequence of actions composing a task involving dual-arm manipulation. After classifying the movement as coordinated (symmetric or asymmetric) or uncoordinated, the framework encodes the set of observed actions in a high-level using Petri Nets [Zöllner et al., 2004]. The idea of sequencing primitive actions to obtain complex behaviours was also exploited to combine manipulation and grasping requirements within the same task [Felip et al., 2013; Lioutikov et al., 2016]. Such an approach lets a robot to re-use the primitive actions across different tasks.

The complexity of a task can also be given by the uncertainty and high dynamism of the environment. To cope with this challenge, some works get inspiration from the neuroscientific belief of a vast repertoire of actions being the basis of any complex human task [Montesano et al., 2008]. As an example, Pastor et al. proposed to consider different primitive motions at the same time. They exemplified this idea with a single-arm pick-and-place task which had to overcome unexpected obstacles. To this aim, their architecture was endowed with two primitives: the pick-and-place dynamics and an obstacle avoidance behaviour. In execution time, both primitive skills were accordingly merged [Pastor et al., 2009].

Indifferently of the usage of primitive motions, either for sequencing or combining them at the same time, they need to be smartly chosen to confront a specific task. Industrial applications usually attach a semantic meaning on the previously demonstrated primitive skills so an end-user can easily reprogram a robot [Makris et al., 2014; Stenmark et al., 2018; Topp, 2017]. However, such an approach is not doable to equip humanoid robots with autonomous manipulation capabilities. Instead, given a description of the task, surrounding environment and affordances of the object to manipulate, a high-level manager can select and trigger the required primitives among the ones available in the library.

2.3 Discussion

There are different learning-based works in the literature which aim to endow a system with robust manipulation skills. They tackle this challenge at different levels, from the particular case where an obstacle needs to be avoided (see Section 2.1) to more generic manipulation scenarios where multiple skills are required (see Section 2.2). Both areas of research suffer from a critical lack of datasets, standardised metrics and scenarios, difficulting a comparison across methods [Grunwald et al., 2008]. This fact makes it hard to analyse the pros and cons of the different approaches quantitatively. Instead, the essence of them is discussed next.

Iterative methods have been extensively used to adapt motions to undemonstrated scenarios, such as the presence of unexpected obstacles. In this fashion, techniques such as RL and ILC use a user-defined reward function to guide the exploration of actions which make the system successful on a particular task. Despite the generalisation capabilities of these techniques, shaping the reward function is not always obvious, and many iterations are needed to converge to successful behaviours. Trials before convergence risk the integrity of the robot due to the unpredictability during the self-learning process. Such a procedure needs to be repeated if the scenario changes since iterative methods cannot reuse knowledge across tasks. Against all these limitations, adapting motions online using a repertoire of primitive skills seems to be a viable alternative to empower robots with human-like manipulation skills.

Most successful learning-based frameworks which handle the manipulation requirements in uncertain and unconstrained environments have been built on top of the online motion adaptation concept. They not only exploit different primitive skills at the same time but also concatenate them sequentially to produce complex and composed behaviours. To the best of the author's knowledge, such a strategy has not been extended to the requirements of dual-arm manipulators. Instead, the existing frameworks for dual-arm systems are limited to the highly controlled industrial environments and focus on including an end-user within the framework's loop to customise the system's behaviour. Halfway between these two approaches is where the novelty of the proposed framework lies: a unified learning-based and modular framework for robust dual-arm manipulation yet customisable by end-users. The strategic formulation of the architecture based on composable primitive skills (i) reduces the complexity of the demonstration process by teaching each skill in a one-at-a-time fashion and (ii) offers generalisation to novel scenarios.

Chapter 3

System Requisites and Modelisation

This thesis pursues an end-to-end learning-based framework that allows real-time autonomous dual-arm manipulation in unfamiliar environments. To this aim, the architecture needs to meet the following requirements: (i) to be able to adapt its plan to achieve a task according to the surrounding environment, while ensuring full synchronisation between both end-effectors, and (ii) to be easily programmable, making a dual-arm platform customizable and accessible even to non-robotics-experts. Bearing these problem requirements in mind, this chapter firstly analyses the challenges that arise when learning actions from human demonstrations. It then describes the typology and diversity of possible actions in a dual-arm system. Finally, this chapter puts the previous pieces together to formulate the modelisation of a dual-arm system and its grasping.

3.1 Learning for a Dual-arm Manipulator

Learning by demonstration (LbD) provides a large set of recording techniques and mathematical supports for encoding a demonstrated skill. However, learning a particular task from human demonstrations raises some challenges, namely (i) clearly understanding the intentions of a demonstration and (ii) establishing a teacher-learner communication channel. Both issues can drastically affect the learning outcome if they are not well adressed [Argall et al., 2009].

The demonstration clarity issue is tackled by leveraging the belief of a vast repertoire of primitive skills being the basis of any complex behaviour. With this in mind, this work avoids demonstrating a task itself but, instead, teaches the robot the involved primitive skills. This task factorisation provides similar benefits as the work in [Bajcsy et al., 2018]: it allows the user to show one feature of the task at a time, and, if required, to correct them individually.

Factorising a complex behaviour into primitive actions reduces the number of DoF to focus on during demonstration time. As an example, the desired position and orientation of a task can be encoded in separate primitive skills and thus, demonstrated one-at-a-time. This fact becomes handy to ease the complicated process of teaching a dual-arm system [Akgun et al., 2012]. This work employs kinesthetic guiding to establish a teacher-learner communication channel which does not suffer from the correspondence problem.

3.2 Dual-arm Primitive Skills Taxonomy

Skills for single-arm manipulation have been well analysed in the robotics community. While some of this knowledge can be extrapolated to dual-arm manipulators, the complexity of these systems requires close attention to their control actions [Grunwald et al., 2008]. This work builds upon the belief in [Montesano et al., 2008], which states that any complex behaviour is composed of a vast repertoire of actions or primitive skills. Then, in the context of manipulation via a dual-arm system, a possible classification of any primitive skill falls into these two groups:

- Absolute skills $\mathcal{S}_a$: imply a change of configuration of the manipulated object in the Cartesian space. Example: move, place or turn an object in a particular manner.
- Relative skills $\mathcal{S}_r$: exert an action on the manipulated object in the object space. Example: the opening of a bottle's screw cap, or hold a parcel employing force contact.

Each type of primitive skill uniquely produces movement in its space. In other words, the absolute and relative skills lie in orthogonal spaces. It is natural to expect from a dual-arm system to simultaneously carry out, at least, one absolute and one relative skill to accomplish a task. Let us analyse the task of moving a bottle to a particular position while opening its screw cap. Both end-effectors synchronously move to reach the desired configuration (absolute skill). At the same time, the left end-effector is constrained to hold the bottle upright (relative skill), while the right end-effector unscrews the cap (relative skill).

3.3 Dual-arm System Modelisation

Given the variety of primitive skills that a dual-arm system can execute, this work seeks to model the robotic platform in a generalisable yet modular fashion, which accounts for both absolute and relative skills. To this aim, let us consider the closed kinematic chain depicted in Figure 3.1 operating in a three-dimensional (3D) workspace $\mathcal{W} = \mathbb{R}^3 \times \mathrm{SO}(3)$. Each arm i, where $i = \{L,\ R\}$, interacts with the same object $\mathcal{O}$. In this context, the absolute skill explains

the movement of the object $\mathcal{O}$ in the workspace $\mathcal{W}$, while the relative skill describes the actions of each end-effector i with respect to the object's reference frame $\{\mathcal{O}\}$. Note that $\{\mathcal{O}\}$ is the centre of the closed-chain dual-arm system. Thus, the remaining of this manuscript uses this notation as the object's and the system's frame indistinguishably.

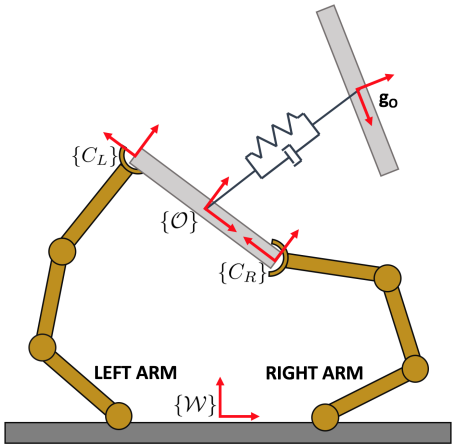


Figure 3.1: Dual-arm manipulator modelled as a closed-chain system. Its dynamics are approximated to those of a spring-damper system actuating in the Cartesian space.

The state of a closed-chain dual-arm system can be described by the position/orientation, linear/angular velocities and accelerations of $\{\mathcal{O}\}$. These variables are subjected to the positional and orientational system's dynamics. As illustrated in Figure 3.1, this work approximates such dynamics to the ones of a spring-damper system acting between the objects's frame $\{\mathcal{O}\}$ and its goal configuration $\mathbf{g}_o$, which accounts for a desired goal position $\mathbf{g}_{o_x}$ and orientation $\mathbf{g}_{o_q}$.

3.3.1 Positional Dynamics

Let the current positional state of the closed-chain dual-arm system be defined by the position, linear velocity and acceleration of its frame $\{\mathcal{O}\}$ in each dimension $N = 3$ of the workspace $\mathcal{W}$, i.e. $(x_o,\ \dot{x}_o,\ \ddot{x}_o)_n\ \forall\ n \in [1,\ N]$. The positional dynamics of the spring-damper system are individually described in each dimension by the following set of nonlinear differential equations:

$$\tau \dot{z}_o = \alpha_x(\beta_x(g_o - x_o) - z_o), \tag{3.1}$$

$$\tau \dot{x}_o = z_o, \tag{3.2}$$

where τ is a scaling factor for time, z_o and $\dot{z}_o$ respectively are the scaled velocity and accelaration, α_x and β_x are constants defining the positional system's dynamics, and g_{o_x} is the model's attractor $\mathbf{g}_{o_x}$ in the n-dimension. The system will converge to g_{o_x} with critically damped dynamics and null velocity when $\tau > 0$, $\alpha_x > 0$, $\beta_x > 0$ and $\beta_x = \alpha_x/4$ [Ijspeert et al., 2013].

3.3.2 Orientational Dynamics

To describe the orientational system's dynamics is of interest a representation which contains no singularities and that its differentiation is numerically stable. However, there is not any minimal representation of orientation such that it lies in $\mathbb{R}^3$ [Ude, 1999]. An alternative consists on using rotation matrices $\mathbf{R} \in \mathrm{SO}(3)$ and individually describing the change (dynamics) in each of the nine numerical values of $\mathbf{R}$ as presented in Equation (3.1). However, this method does not guarantee that the orthogonality requirements for rotation matrices are met at any time.

Another possible representation of orientations is unit quaternions $\mathbf{q} \in \mathbb{R}^4 = \mathbb{S}^3$ [Kramberger et al., 2016; Ude et al., 2014]. They address the strong assumption of independence between numerical values by encoding the rotation as a whole, at the cost of more complex and computationally expensive calculations. Let the current orientational state of the closed-chain dual-arm system be defined by the orientation, angular velocity and acceleration of its system's frame $\{\mathcal{O}\}$ in the workspace $\mathcal{W}$, i.e. $(\mathbf{q}_o,\ \dot{\mathbf{q}}_o,\ \ddot{\mathbf{q}}_o) \in \mathbb{R}^4 = \mathbb{S}^3$. The quaternion-based spring-damper modelisation is described by the following set of nonlinear differential equations:

$$\tau \dot{\boldsymbol{\eta}}_o = \alpha_q(\beta_q\, 2\log(\mathbf{g}_o * \bar{\mathbf{q}}_o) - \boldsymbol{\eta}_o), \tag{3.3}$$

$$\tau \dot{\mathbf{q}}_o = \frac{1}{2}\boldsymbol{\eta}_o * \mathbf{q}_o, \tag{3.4}$$

where $\boldsymbol{\eta}_o$ and $\dot{\boldsymbol{\eta}}_o$ respectively are the scaled angular velocity and accelaration, α_q and β_q are constants defining the system's orientational dynamics, and $\mathbf{g}_{o_q} \in \mathbb{S}^3$ is the model's orientation attractor. The operators $\log(\cdot)$, $*$, and $\bar{\mathbf{q}}_o$ denote the logarithm, multiplication and conjugate operations for quaternions, respectively.

3.3.3 Coupling Terms

The positional and rotational dynamics respectively described in Equation (3.1)-(3.2) and in Equation (3.3)-(3.4) generate a linear continuous displacement between any initial and goal state $\mathbf{g}_o$. Any other dynamical behaviour can be encoded by extending these models with coupling terms, i.e. virtual external force acting on the system's frame $\{\mathcal{O}\}$. For a manipulator in a workspace $\mathcal{W} = \mathbb{R}^3 \times \mathrm{SO}(3)$, the positional dynamics are defined as:

$$\tau \dot{\mathbf{z}}_o = \alpha_x(\beta_x(\mathbf{g}_o - \mathbf{x}_o) - \mathbf{z}_o) + \mathbf{f}_{o_x}(\cdot), \tag{3.5}$$

$$\tau \dot{\mathbf{x}}_o = \mathbf{z}_o, \tag{3.6}$$

where $(\mathbf{x}_o,\ \dot{\mathbf{x}}_o,\ \ddot{\mathbf{x}}_o) \in \mathbb{R}^3$ is the system's positional state and $\mathbf{f}_{o_x}(\cdot) \in \mathbb{R}^3$ is the coupling force.

For the same manipulator in a workspace $\mathcal{W} = \mathbb{R}^3 \times \mathrm{SO}(3)$, the system's orientational state is described by $(\mathbf{q}_o,\ \dot{\mathbf{q}}_o,\ \ddot{\mathbf{q}}_o) \in \mathbb{R}^4$, the dynamics of which are dfined as:

$$\tau \dot{\boldsymbol{\eta}}_o = \alpha_q(\beta_q\, 2\log(\mathbf{g}_o * \bar{\mathbf{q}}_o) - \boldsymbol{\eta}_o) + \mathbf{f}_{o_q}(\cdot), \tag{3.7}$$

$$\tau \dot{\mathbf{q}}_o = \frac{1}{2}\boldsymbol{\eta}_o * \mathbf{q}_o, \tag{3.8}$$

where $\mathbf{f}_{o_q}(\cdot) \in \mathbb{R}^4$ is the corresponding coupling term.

The coupling terms describe the profile of the external force affecting the natural positional and orientational dynamics of the system, respectively. In other words, $\mathbf{f}_{o_x}(\cdot)$ and $\mathbf{f}_{o_q}(\cdot)$ characterise the system's behaviour, thus being useful to encode and retrieve any primitive skill. For the sake of clarity, an example is provided to illustrate the spring-damper modelisation altogether with the use of coupling terms.

Figure 4.1 examplifies the coupling terms concept with the skill of drawing the letter G on a two-dimensional (2D) plane. To this aim, the system is modelled in $\mathcal{W} = \mathbb{R}^2$ as two spring-damper systems describing the change in position in each dimension. The starting and goal positional configurations are set at the most top-right and inner part of the letter G, respectively. According to this modelisation, different G-shapes (see Figure 3.2a) are defined by the corresponding external forces (coupling terms) actuating on each dimension (see Figure 3.2b and Figure 3.2c). The letter's silhouette is uniquely conditioned by these external forces (coupling terms); a smoother set of forces leads to a rounder G-shape (highlighted in red).

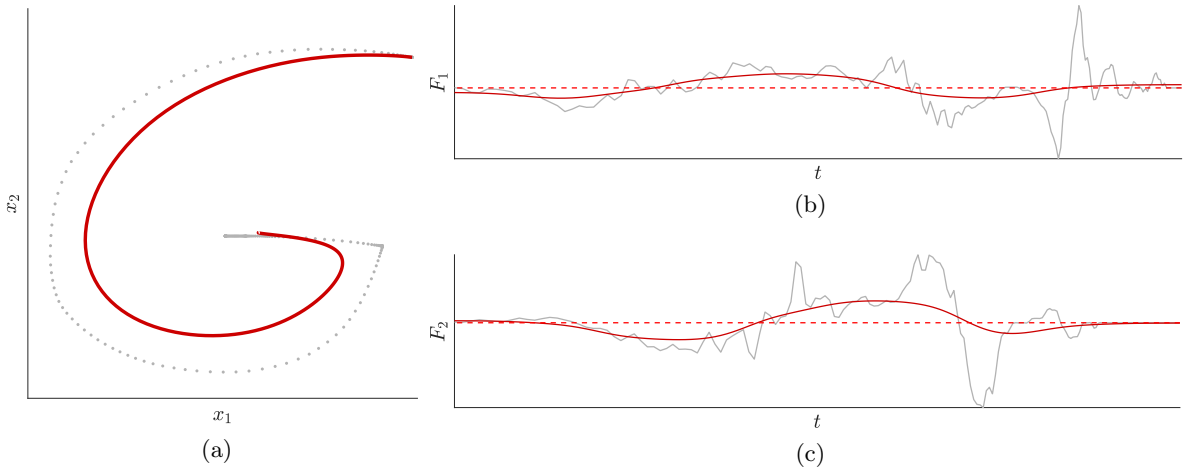


Figure 3.2: Skill of drawing the letter G represented in the force level according to a spring-damper system modelisation. (a) Resulting G-shapes on a 2D plane. (b)-(c) External forces in the x_1 and x_2 dimensions, respectively. The dashed line is a zero-force reference.

3.4 Dual-arm Grasping Geometry

Any action referenced to the object's frame $\{\mathcal{O}\}$ can be projected to the end-effectors using the grasping geometry of the manipulated object. This allows computing the required end-effector control commands to achieve a particular absolute task. For the end-effector i there is a transformation map or grasping matrix $\mathbf{G}_i$ which establishes a velocity relation between the contact point C_i and the systems reference frame $\{\mathcal{O}\}$ as:

$$\dot{\mathbf{y}}_{C_i} = \mathbf{G}_i^T \, \dot{\mathbf{y}}_o, \tag{3.9}$$

where, for a workspace $\mathcal{W} = \mathbb{R}^3 \times \mathrm{SO}(3)$, $\dot{\mathbf{y}} \in \mathbb{R}^6$ is the concatenation of the system's positional velocity $\dot{\mathbf{x}} \in \mathbb{R}^3$ and angular velocity in Euler format $Euler(\dot{\mathbf{q}}) \in \mathbb{R}^3$.

The grasping matrix of the end-effector i is defined as:

$$\mathbf{G}_i \in \mathbb{R}^{6\times 6} = \begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{O}_{3\times 3} \\ \mathbb{S}(\mathbf{r}_i) & \mathbf{I}_{3\times 3} \end{bmatrix}, \tag{3.10}$$

where $\mathbf{I}_{3\times 3}$ is the identity matrix, and $\mathbb{S}(\mathbf{r}_i) \in \mathbb{R}^{3\times 3}$ is the skew-symmetric matrix performing the cross product:

$$\mathbb{S}(\mathbf{r}_i) = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}, \tag{3.11}$$

where $\mathbf{r}_i$ is the distance from the object's reference frame $\{\mathcal{O}\}$ to the contact point C_i.

A global grasp map $\mathbf{G}$ for the dual-arm manipulator can be defined by horizontally concatenating the grasp matrix of each end-effector, i.e. $\mathbf{G} = [\mathbf{G}_L \; \mathbf{G}_R] \in \mathbb{R}^{6\times 12}$ where $\mathbf{G}_L$ and $\mathbf{G}_R$ are the left and right arm grasp matrix, respectively.

Chapter 4

Learning Primitive Skills

Humans master a significant number of primitive skills. The modelisation of the dual-arm manipulator in the Cartesian space as a spring-damper system let us exploit coupling terms to make the robot behave in a more human-like manner. In other words, coupling terms can be used to encode and reproduce any primitive skill. This chapter presents some generic mathematical formulation under which many primitives can be encoded, namely: goal-oriented motions (both for position and orientation), obstacle avoidance and force interaction.

4.1 Goal-oriented Dynamics

The non-linear dynamical behaviour of any task can be represented using dynamic movement primitives (DMPs). This mathematical encoding support has proven to be a versatile tool for modelling and learning complex motions, given that: (a) any movement can be efficiently learned and generated, (b) a unique demonstration is already generalisable, (c) convergence to the goal is guaranteed, and (d) their representation is translation and time-invariant [Ijspeert et al., 2013; Pastor et al., 2009]. Following up with the example depicted in Section 3.3.3, Figure 4.1a shows some of these DMP-inherent generalisation capabilities applied to positional dynamics.

The system modelisation defined in Equation (3.5)-(3.8) can integrate DMPs as the coupling terms $\mathbf{f}_{o_x}(\cdot)$ and $\mathbf{f}_{o_q}(\cdot)$. Regardless of the different formulation of the positional and orientational dynamics, DMPs are applied in the same fashion. For each spring-damper defining the perturbationless system's dynamics, there is one DMP-based coupling term which shapes the dynamics. That is, for a workspace $\mathcal{W} = \mathbb{R}^3 \times \mathrm{SO}(3)$, a total of seven DMPs are required: three for the positional and four for the orientational information.

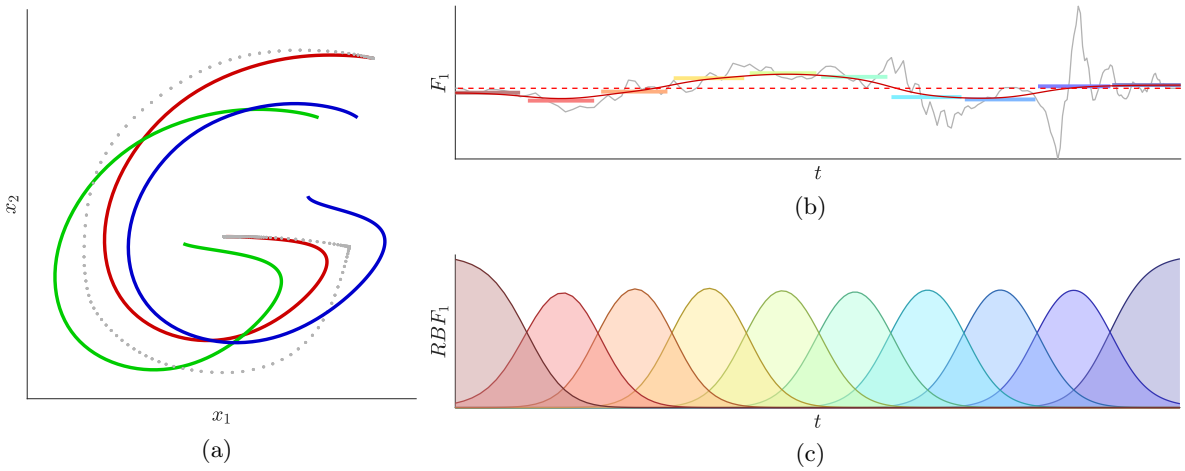


Figure 4.1: DMP-based modelisation and generalisation of G-shapes on a 2D plane. (a) Given demonstration (polka dotted trajectory), learnt G-shape (red trajectory) and generalisaton from different start and goal positions (blue and green trajectories). (b)-(c) Learnt dynamics (red line) in the x_1 dimension. They are the result of a weighted combination of ten RBF. The corresponding weights are the ten segments in (b), and the set of RBF are depicted in (c).

Formally, a DMP is a weighted linear combination of non-linear RBFs [Ijspeert et al., 2013; Pastor et al., 2009]. The value of such non-linear $\mathbf{f}(\cdot)$ function when evaluated at a specific numerical value $k \in \mathbf{k}$ is defined as:

$$f(k) = \frac{\sum_{i=1}^{N} w_i \Psi_i(k)}{\sum_{i=1}^{N} \Psi_i(k)}\, k, \tag{4.1}$$

$$\Psi_i(k) = \exp\Big(-h_i(k - c_i)^2\Big), \tag{4.2}$$

where c_i and $h_i > 0$ are the centres and widths, respectively, of the $i \in [1,\ N]$ RBFs distributed along the trajectory. Each RBF is weighted by w_i. The phase variable $\mathbf{k}$ avoids direct dependency of $\mathbf{f}(\cdot)$, and thus, the coupling terms, on time. The dynamics of $\mathbf{k}$ are defined as:

$$\tau \dot{\mathbf{k}} = -\alpha_k \mathbf{k}, \tag{4.3}$$

where the initial value of the canonical system $\mathbf{k}(0) = 1$ and α_k is a positive constant.

The learning of the DMPs relies on adjusting the set of RBF, i.e. the weight vector $\mathbf{w}$, composed of all weights w_i, which makes the previously modelled system in Equation (3.5)-(3.8) adjust to a recorded skill propioception information $\{(\mathbf{x},\ \dot{\mathbf{x}},\ \ddot{\mathbf{x}}) \in \mathbb{R}^3\ (\mathbf{q},\ \dot{\mathbf{q}},\ \ddot{\mathbf{q}}) \in \mathbb{R}^4\}_k\ \forall\ k \in [1\ T]$, where k represents time $t = k\Delta t$ and T is the total duration of the demonstrated primitive skill.

Figure 4.1b and Figure 4.1c follow up with the example introduced in Section 3.3.3 to illustrate this learning procedure. Finding the weights w_i (line segments in Figure 4.1b) which make the set of ten RBF (see Figure 4.1c) adjust to the dynamics of the recorded G-shape (red trajectory).

4.2 Obstacle Avoidance

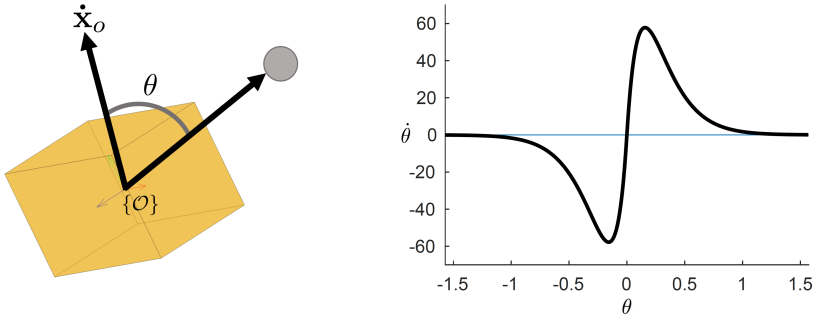
An analytical description of how humans steer around an obstacle was first presented in [Fajen and Warren, 2003]. Later on, such biologically-inspired formulation was used in [Hoffmann et al., 2009] for single-arm manipulation purposes. Let $\mathbf{x}_o$, $\dot{\mathbf{x}}_o$, and θ_o be respectively the system's $\{\mathcal{O}\}$ position, velocity and orientation in the workspace $\mathcal{W}$ (see Figure 4.2a). In order to avoid an obstacle, the positional dynamics in Equation (3.5)-(3.6) need to change accordingly to:

$$\mathbf{f}_{o_x}(\cdot) \sim \mathbf{f}_o(\mathbf{x}_o,\ \dot{\mathbf{x}}_o) = \mathbf{R}\ \dot{\mathbf{x}}_o\ \dot{\theta}, \tag{4.4}$$

where $\mathbf{R} \in \mathrm{SO}(3)$ is a $\pi/2$ rotation matrix with respect to the vector $\mathbf{r} = (\mathbf{x}_{obstacle} - \mathbf{x}_o) \times \dot{\mathbf{x}}_o$, and $\dot{\theta}$ is the desired turning velocity:

$$\dot{\theta} = \gamma\ \theta \exp\big(-\beta\ |\theta|\big), \tag{4.5}$$

where γ and β are tuning constants. Their effect can be best understood in Figure 4.2b: γ sets the abruptness of the obstacle avoidance behaviour, and β determines its sensitivity.



(a) (b)

Figure 4.2: Obstacle avoidance primitive skill proposed in [Fajen and Warren, 2003]. (a) Manipulated object (brown prism) and obstacle (grey circle). (b) Change of steering angle $\dot{\theta}$ of the original formulation in Equation (4.5) with $\gamma = 1000$ and $\beta = 20/\pi$.

The original formulation of Fajen and Warren experiences some limitations, namely: (i) the dead zone that makes the system less reactive as the heading towards an obstacle tends to zero (see black curve in Figure 4.3a and Figure 4.3b), (ii) the lack of distance awareness to the obstacles,

and (iii) the fact of being parameter dependant [Rai et al., 2014, 2017]. To address these issues, this work reformulates Equation (4.5) as:

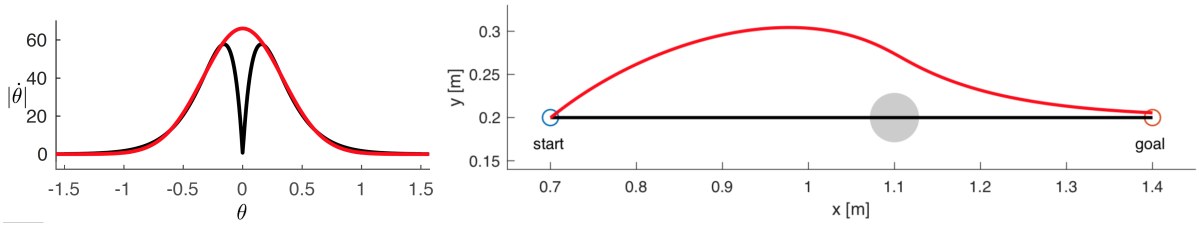
$$\dot{\theta} = a\,\mathrm{sign}(\theta)\exp\left(-\frac{\theta^2}{c^2}\right)\exp\left(-k\,d^2\right), \tag{4.6}$$

where $a\,\mathrm{sign}(\theta)\exp\left(-\theta^2/c^2\right)$ addresses the aforementioned (i)-issue by shapping the absolute change of steering angle as a bell-shaped function (see red curve in Figure 4.3a and red trajectory in Figure 4.3b), and $\exp\left(-k\,d^2\right)$ tackles the (ii)-issue by vanishing the effect of the previous term according to the distance d to the obstacle. The (iii)-issue is solved by learning the parameters a, c and k from human demonstrations, which control the abruptness, sensitivity, and anticipation of the obstacle avoidance behaviour, respectively.

Learning the parameters a, c and k from human demonstrations avoids blindly hand-tunning the behaviour of the obstacle avoidance skill. This is achieved using least mean squares (LMS) after log-linearising Equation (4.6) and arranging it as:

$$\log\dot{\boldsymbol{\theta}} = \begin{bmatrix}\log a & \dfrac{1}{c^2} & k\end{bmatrix}\begin{bmatrix}\mathbf{1}\\ -\boldsymbol{\theta}^2\\ -\mathbf{d}^2\end{bmatrix}, \tag{4.7}$$

where the training data $\dot{\boldsymbol{\theta}}$, $\boldsymbol{\theta}$ and $\mathbf{d}$ contain the periodically sampled value of $\dot{\theta}$, θ and d experienced during the obstacle avoidance demonstration. $\dot{\theta}$ is retrieved from Equation (4.4), where $\mathbf{f}_o(\mathbf{x}_o,\ \dot{\mathbf{x}}_o) = \mathbf{f}_{o_x}(\cdot)_{obs} - \mathbf{f}_{o_x}(\cdot)$, i.e. the difference on the dynamics between a perturbationless task $\mathbf{f}_{o_x}(\cdot)$ and one with obstacles $\mathbf{f}_{o_x}(\cdot)_{obs}$ is only motivated by the presence of an obstacle.



(a) (b)

Figure 4.3: Change of steering angle $\dot{\theta}$ and dead zone issue. (a) Absolute representation of Figure 4.2b (black curve), and the proposed alternative in Equation (4.6) with $a = 66.07$, $c = 0.4732$ and $k = 0$ (red curve). (b) Following the same colour code, both methods confronting an obstacle (grey circle) in a 2D environment. The original formulation does not react against imminent collision, instead the proposed alternative provides a smooth and coherent behaviour.

4.3 Force Interaction

Manipulation of a rigid object via a dual-arm system requires each end-effector to be in contact with the object. This arises the need of controlling the force applied by each end-effector on the object, thus preventing damaging it or the system itself. A particular case is manipulation by force contact (without grasping the object), which not only requires each end-effector to be in contact with the object but also to apply the sufficient forces to ensure grasp maintenance, i.e. prevention of contact separation and unwanted contact sliding [Lin et al., 2018].

The complexity of this task usually requires modelling the necessary coupling forces as a dynamical function subjected to the complete state of the system, i.e. $\mathbf{f}(\mathbf{x}_o,\ \dot{\mathbf{x}}_o,\ \ddot{\mathbf{x}}_o)$. However, learning this complex model from human demonstrations can be challenging and might require many demonstrations. An alternative for applications with low-dynamical requirements was presented in [Gams et al., 2014]. They approximated the previous dynamical function with a force tracking controller defined as:

$$\dot{\mathbf{y}}_{C_i} = \mathbf{K}(\mathbf{F}_{d_i} - \mathbf{F}_{r_i}), \tag{4.8}$$

where, for a workspace $\mathcal{W} = \mathbb{R}^3 \times \mathrm{SO}(3)$, $\dot{\mathbf{y}}_{C_i} \in \mathbb{R}^6$ contains the linear and angular (in Euler format) velocity commands for the end-effector i to correct the errors in force and torque contact, $\mathbf{K} \in \mathbb{R}^{6\times6}$ is an error multiplying constant, $\mathbf{F}_{d_i} \in \mathbb{R}^6$ is the desired coupling force and $\mathbf{F}_{r_i} \in \mathbb{R}^6$ is the current coupling force retrieved from the robot's sensors. Thus, the learning of this primitive skill resides on learning from demonstrations which $\mathbf{F}_{d_i}$ ensures grasp maintenance.

Chapter 5

Framework for Robust Dual-arm Manipulation

In order to endow robots with real-time, robust and autonomous dual-arm manipulation, while letting non-robotics-experts to program and customise the system's behaviour easily, this work presents the learning-based framework depicted in Figure 5.1. Such an architecture jointly addresses the aforementioned requirements with three sequential parts: (i) the learning module that learns a set of primitive skills from human demonstrations, (ii) the roll-out module that combines those primitive skills to plan a trajectory which makes the system succeed at a task, even in unfamiliar environments and (iii) the evaluation module that lets a human-in-the-loop supervise the robot's behaviour and reteach a specific skill, if required.

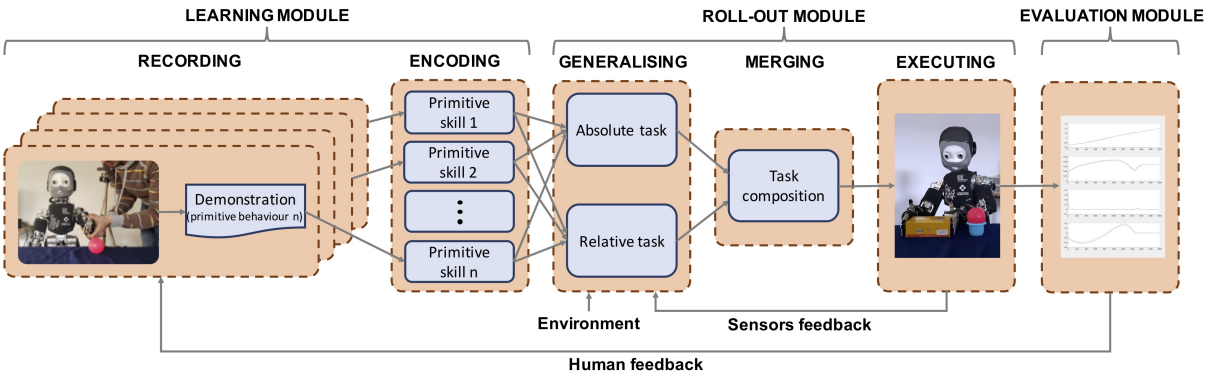


Figure 5.1: Scheme of the three stages involved in the proposal. Learning: a human demonstrator teaches some primitives behaviours to a system. Roll-out: the robot exploits (generalises and combines accordingly to the environment awareness) the acquired knowledge. Evaluation: an evaluator inspects the system's performance and decides whether reteaching is necessary.

5.1 Learning Module

A complex task can be represented by a limited repertoire of simple behaviours, i.e. primitive skills [Montesano et al., 2008; Pastor et al., 2009]. Motivated by the hindrance of demonstrating a behaviour through kinesthetic guiding and the challenge of deducing the human intentions behind such demonstration (see Section 3.1), the proposed framework learns a set of primitive skills individually, i.e. in a one-at-a-time fashion [Bajcsy et al., 2018]. In the context of dual-arm manipulation, this results in a library of independent primitive skills, some of which are represented in the absolute skill space and some others in the relative skill space (see Section 3.2).

Following up with the modelisation of the dual-arm system introduced in Section 3.3 and later formalised in Equation (3.5)-(3.8), the learning of any primitive skill is based on describing the virtual external forces affecting the natural positional and orientational dynamics of the modelled spring-damper system. This is, given a kinesthetic demonstration represented by the acquired propioception information $\{(\mathbf{x},\ \dot{\mathbf{x}},\ \ddot{\mathbf{x}}) \in \mathbb{R}^3\ (\mathbf{q},\ \dot{\mathbf{q}},\ \ddot{\mathbf{q}}) \in \mathbb{R}^4\}_k\ \forall\ k \in [1\ T]$, find and learn the coupling terms $\{\mathbf{f}_{o_x}(\cdot) \in \mathbb{R}^3\ \mathbf{f}_{o_q}(\cdot) \in \mathbb{R}^4\}_k\ \forall\ k \in [1\ T]$, where k represents time $t = k\Delta t$ and T is the total duration of the demonstrated primitive skill.

Within the framework, the profiles of the coupling forces are characterised by isolating such terms from the system's model in Equation (3.5)-(3.8). Then, each retrieved coupling force is learned accordingly to the demonstrated primitive skill; as discussed in Chapter 4, different primitive skills have a different mathematical representation. Thus, the framework needs to be aware of which action is being learnt at demonstration time.

5.2 Roll-out Module

Given a library containing a repertoire of absolute and relative primitive skills, these motions need to be properly combined to confront dual-arm tasks in a wide range of scenarios. The framework addresses this challenge in a twofold procedure: (i) retrieves the effect of each primitive skill at the velocity level, and (ii) selects and activates the effects of those primitives to succeed in the commanded task. Formally, for a workspace $\mathcal{W} = \mathbb{R}^3 \times \mathrm{SO}(3)$, this is defined as:

$$\begin{bmatrix} \dot{\mathbf{y}}_L \\ \dot{\mathbf{y}}_R \end{bmatrix} = \mathbf{G}^T \sum_{j=1}^{J} w_j\ \dot{\mathbf{y}}_{o_j} + \sum_{k=1}^{K} w_k \begin{bmatrix} \dot{\mathbf{y}}_{C_L} \\ \dot{\mathbf{y}}_{C_R} \end{bmatrix}, \tag{5.1}$$

where, considering $i = \{L,\ R\}$, $\dot{\mathbf{y}}_i \in \mathbb{R}^6$ describes the linear and angular velocity commands for the i end-effector which satisfies the set of activated primitive skills, and $\dot{\mathbf{y}}_{o_j} \in \mathbb{R}^6$ and $\dot{\mathbf{y}}_{C_i} \in \mathbb{R}^6$

are the velocities of the $j \in [1,\ J]$ absolute and $k \in [1,\ K]$ relative primitive skill available in the library. Note that all these velocity vectors belong in $\mathbb{R}^6$ because their components representing angular velocities are in Euler format. Absolute and relative primitive skill selection is conducted with the weights w_j and w_k, respectively.

The velocity for each primitive skill is retrieved from the system's model defined in Equation (3.5)-(3.8). Bearing in mind that each primitive skill is described by its coupling terms, this model provides the desired system acceleration subject to a primitive skill. By means of integration, the corresponding desired velocity and position can be obtained. All this process is known as roll-out in the LbD community. According to the design of the proposed framework, this process uniquely needs to be extended until the velocity level (see Equation (5.1)).

Dealing with the action challenge, i.e. which set of primitive skills from the library needs to be considered accordingly to the desired task and object affordances, is currently not the focus of this research. In fact, this question constitutes the main motivation of a particular line of research. Such an alternative is considered for future work (see Chapter 7). In the scope of this thesis, the desired task and object to manipulate are known in advance, which lets us loading the framework's library with the essential primitive skills. Given this context, all weights w_j and w_k in Equation (5.1) are all set to one. This assumption is further detailed and exemplified in the different evaluation scenarios reported in Chapter 6.

5.3 Evaluation Module

There is a critical lack of standardised metrics for motion evaluation purposes, difficulting an objective analysis of the trajectory conducted by a system. This becomes even more critical when dealing with dynamic and unconstrained environments. This fact has led to qualitative evaluation being the predominant assessment protocol of motions in the LbD literature [Grunwald et al., 2008]. In some cases, this qualitative analysis goes along with ad-hoc metrics to numerically represent a particular feature of the system's performance. Below follows a brief discussion of some procedures or metrics that can be employed to evaluate the resulting outcome of the proposed framework. However, their usage is dependant on the user's interest.

As an example, the human-robot interaction (HRI) community is interested in enhancing the acceptability and compatibility of robots in human workspaces [Ajoudani et al., 2017]. At some extent, the proposed framework can be used for this purpose; learning all primitive skills from human demonstrations arise expectations about the degree of similarity that a robot's final performance might have with the demonstrator's behaviour under the same conditions.

A possible metric to quantitatively evaluate this feature consists on computing the human-like similarity of the framework's outcome. An alternative for conducting this study consists of recording some samples of both the robotic and human approach in a particular scenario to quantify their deviation with the Kullback-Leibler (KL) divergence statistic. The lower this indicator is, the higher the chances are that these two agents have similar behaviours.

An overall analysis might also provide useful information about the system's performance. Supervising and qualitatively assessing the robot's performance can help to detect a wrongly learnt primitive skill. For instance, two main primitives are required for hitting a tennis ball: keeping a proper orientation of the tennis racket and performing the motion to push the ball. The failure on correctly inferring any of these two skills to a novel situation should be easily detectable. In this context, the demonstrator could reteach the misleading characteristic of the robot's behaviour while keeping the rest of primitive skills unmodified. This feature comes in handy to avoid the laborious process of loading the framework's library from the ground up.

Note that being able to reteach a particular aspect of the robot's behaviour results from the strategic formulation of the system, primitive skills and framework presented throughout this manuscript. As previously discussed in Section 5.1, the proposed architecture exploits the one-at-a-time teaching fashion introduced in [Bajcsy et al., 2018] for a twofold benefit: (i) from the learning and manipulation point of view, harvesting primitive skills lets a robot adapt its behaviour to confront novel scenarios (see Section 5.2), and (ii) equitably relevant for the HRI community, learning each primitive skill in isolation eases the hindrance of teaching a robot through kinesthetic guidance, which is an extremely critical issue in the dual-arm context. All in all, these two advantages are gathered under the realm of the proposed framework, thus allowing non-robotics-experts to interact, teach and modify the behaviour of a dual-arm system endowed with enhanced generalisation capabilities to novel scenarios.

Chapter 6

Results and Evaluation

Experimental evaluation has been carried out to demonstrate the applicability of the two main contributions of this work: (i) an online obstacle avoidance skill which reacts even against imminent collisions, and (ii) a framework for generalisable dual-arm manipulation which learns from human demonstrations. Such an evaluation has been subjected to some limitations inherent from the evaluation platform, namely: (i) reduced dual-arm workspace, and (ii) lack of realistic simulated force/torque sensors. Moreover, some assumptions have been made to ease the evaluation of the proposal: (i) the required set of primitive skills to confront each scenario is defined beforehand (see Chapter 5), and (ii) the location of the obstacles and the object to manipulate is directly retrieved from the simulator. This information could also have been extracted by visual perception but has been avoided as it is currently not the focus of this research.

This chapter firstly introduces the experimental setup used to evaluate the framework and elaborates on the aforementioned limitations and assumptions. It then validates the generality of the goal-oriented encoding as well as the suitability of the reformulated obstacle avoidance behaviour for humanoid robots. Finally, this chapter analyses the applicability of the entire framework to conduct dual-arm pick-and-place tasks with the presence of obstacles. The experiments have involved synthetic environments, the simulated and real iCub humanoid robot.

6.1 Experimental Setup

The generality of the proposed framework is narrowed down to provide an application case. The designed showcase has to be feasible for the dual-arm robotic platform available in the Edinburgh Centre for Robotics. Thus, this section firstly gives an overview of the iCub humanoid robot

and the designed pick-and-place task. Bearing in mind the robot's capabilities and the tasks requirements, this section analyses the workspace and object's manipulability. It then details the undertaken strategy to record and teach a robot from human demonstrations. Finally, this section gives an overview of the proposal's deployment on a simulated iCub humanoid robot.

6.1.1 iCub Humanoid Robot

iCub is an open source humanoid robot testbed for research into human cognition and artificial intelligence applications [Metta et al., 2008] (see Figure 6.1). It was first designed back in 2008 by the Italian Institute of Technology (IIT). The dimensions of this platform are similar to that of a 3.5-year-old child (104cm). It can see, hear, and has the sense of proprioception and movement (using encoders, accelerometers and gyroscopes). It also has the sense of touch, and it senses how much force exerts on the environment. The robot is not designed for autonomous mobility, consequently not being equipped with onboard batteries or the required processors. Instead, it has an umbilical cable to provide power and a network connection via Gigabit Ethernet.

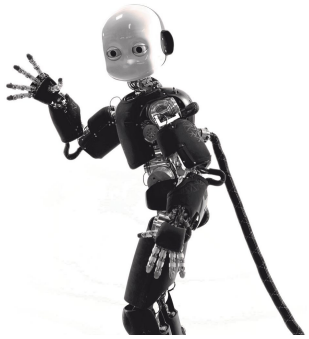


Figure 6.1: iCub humanoid robot.

Since the first robots were constructed, the design has undergone several revisions and improvements. The current version in the Edinburgh Centre for Robotics has 53 actuated DoFs organised as follows: three in the torso, six in the head, seven in each arm, nine in each hand, and six in each leg. The head has stereo cameras in a swivel mounting where eyes would be located on a human and microphones on the side. It is mainly covered by an elastic fabric simulating the face skin, which lets the robot make some facial expressions by moving its mouth. All this equipment is connected with controller area network (CAN) bus to an on-board PC104, which centralises all control of the humanoid.

The constitution of the iCub's seven-DoF manipulators, its software architecture which operates under the YARP middleware, and its inverse kinematic control are detailed in Appendix B.

6.1.2 Pick-and-Place Showcase

The suitability of the proposed framework to endow a dual-arm manipulator with enhanced autonomy is evaluated with a dual-arm pick-and-place of a parcel. This task does not involve dexterity with the fingers but, instead, manipulation by force contact, i.e. each end-effector needs to be in contact with the object and apply the sufficient forces to ensure grasp maintenance [Lin et al., 2018]. Moreover, maintaining a parallel orientation between the end-effectors promotes a larger area of contact, and thus, more friction. Not satisfying these synchronisation requirements may lead to unwanted contact sliding or risking the handled object to stress.

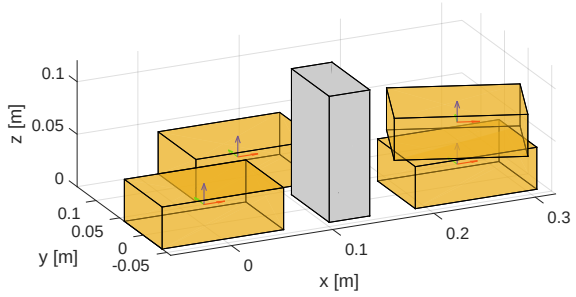


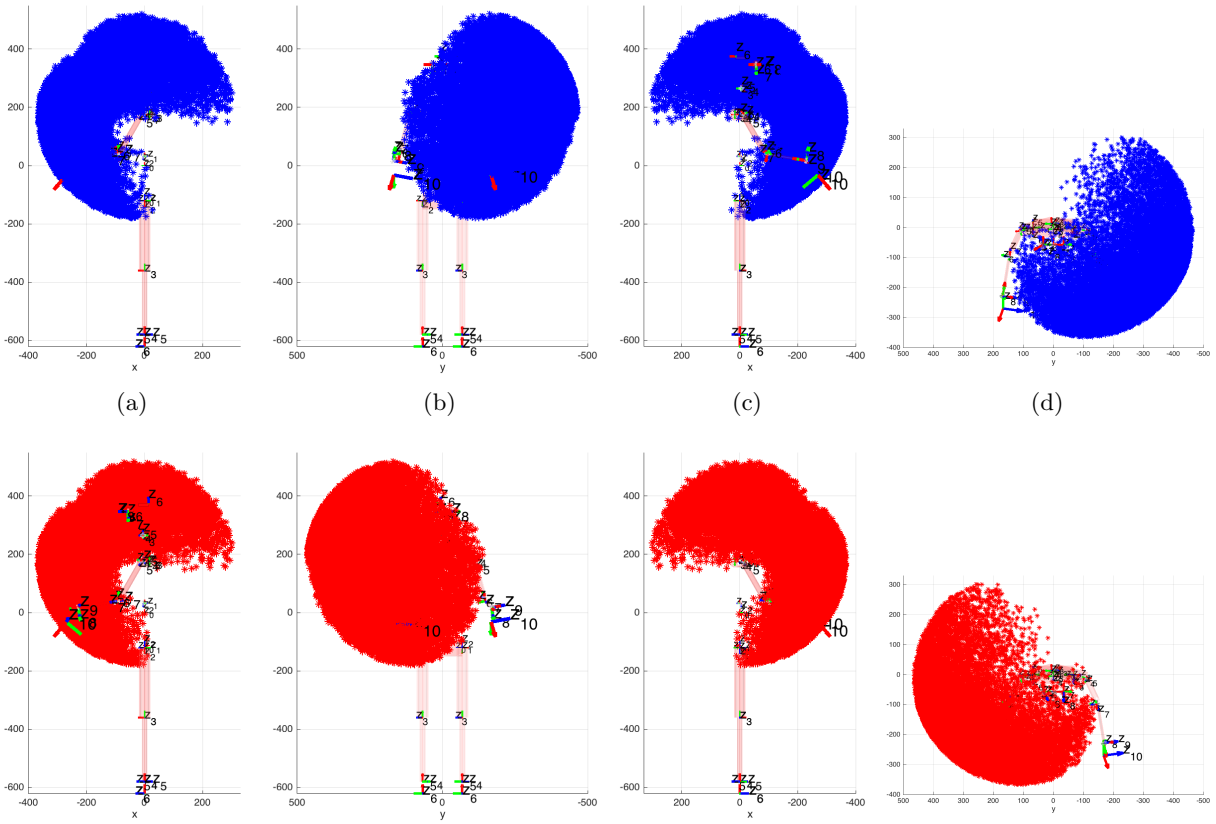
Figure 6.2: Pick-and-place of a parcel (brown prism) in the presence of obstacles (grey prism).

The designed pick-and-place task is sketched in Figure 6.2. Parcels (brown prisms) are meant to be taken and placed in different configurations of the workspace, adjusting the behaviour of the dual-arm whether there is an obstacle or not (grey prism). To this aim, the library of primitive skills is loaded with: underlying dynamics of a pick-and-place task, obstacle avoidance and grasp maintenance (force interaction) behaviours. The dimensions of the parcel and obstacle are dependant on the workspace and manipulability analysis conducted next.

6.1.3 Workspace and Manipulability Analysis

Exploiting iCub's whole control body dynamics requires great expertise with the platform and many calibration routines. In the scope of this project, the control of the humanoid robot has been initially limited to the two seven-DoF manipulators. This leads to a more restricted workspace where the platform can operate. Thus, it is essential to set a showcase task which lies within the common (dual-arm) workspace between end-effectors.

iCub's dual-arm workspace has been determined using the Monte Carlo sampling approach proposed in [Alciatore and Ng, 1994]. This method uniformly samples one arm's joint space to compute the corresponding end-effector poses. Such a method has been implemented in MATLAB. Figure 6.3 illustrates iCub's left and right arm workspace when using this approach.



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

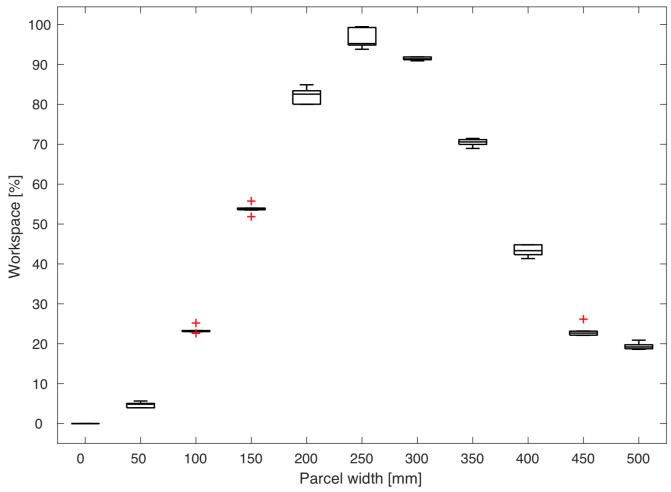
(e) (f) (g) (h)

Figure 6.3: iCub's left (top row) and right (bottom row) end-effector's workspaces. From left to right: left, front, right and top view.

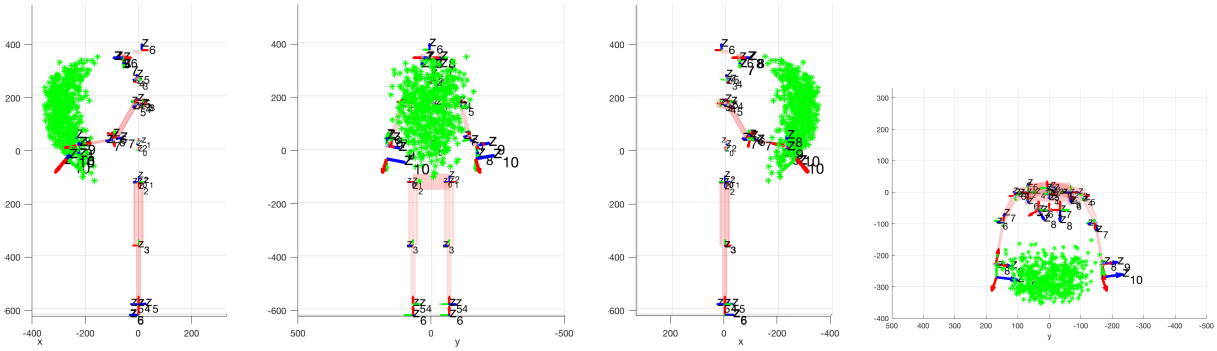
A total of 100,000 joint configuration sets have been sampled. The iCub's forwad kinematics (FK) used to perform this evaluation is detailed in Appendix B.2.1.

Applying the Monte Carlo sampling approach individually for each arm leads to two point clouds which do not consider the constraints of a closed-chain dual-arm system subjected to a specific task. As discussed in Section 6.1.2, the kinematic requirements of a pick-and-place task are constant distance and parallel orientation between both end-effectors. These constraints have been imposed on the previous point clouds depicted in Figure 6.3, seeking for pairs of end-effector configurations (coming from different arms) which are separated the parcel's size d with $\pm 5mm$ tolerance, and similar orientation in the workspace with ± 3 degrees of tolerance at any axis.

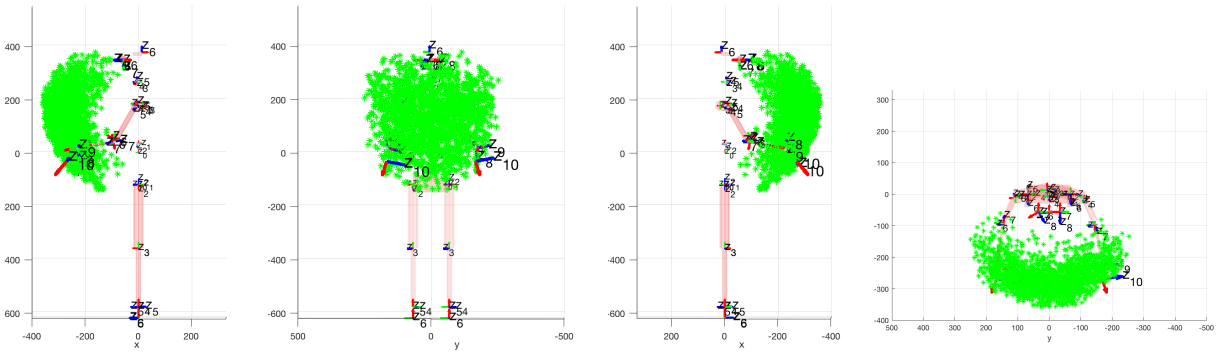
The dual-arm workspace is analysed subject to the parcel's characteristics. This aims to find the parcel's width which leads to a larger manipulability of the parcel during the experiments. Keeping the aforementioned orientation constraints, the considered parcels widths range from $0mm$ (end-effectors in flat clapping configuration) to $500mm$, in increment steps of $50mm$.



(a)



(b) (c) (d) (e)

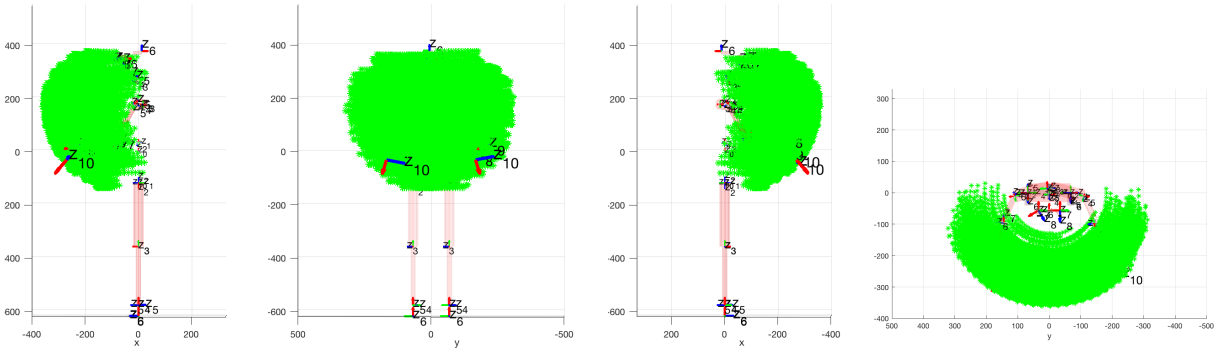


(f) (g) (h) (i)

Figure 6.4: Analysis of iCub's dual-arm workspace contrained by pick-and-place task requirements. Top row: box plot reflecting the constrained workspace subject to different parcel widths d. Second row: workspace for parcel width $d = 100 \pm 5mm$. Bottom row: parcel width $d = 250 \pm 5mm$. From left to right: left, front, right and top view. All presented information is also contrained by the aforementioned orientation error of 0 ± 3 degrees.

Bearing in mind the randomness of the used Monte Carlo sampling approach, a total of 10 trials per parcel size were conducted to determine the probabilistic significance of the analysis. The obtained data is presented as a box plot and exemplified for the parcel widths of $d = 100mm$ and $d = 250mm$ in Figure 6.4. After verifying probabilistic independence between clusters, it can be concluded that the parcel size constraining the less iCub's workspace is $d = 250mm$. Thus, the experiments reported in the remainder of this thesis are for parcels of this size.

To further increase iCub's dual-arm workspace, the torso's DoFs can be used. However, as explained in Appendix B.2.1, the built-in YARP implementation of iCub's inverse kinematics (IK) does not cope with the complexity of internally solving the closed-chain problem, thus not managing the corresponding DoFs of the torso. Following the recommendations in the YARP documentation, an external manager has been implemented to control the torso's roll according to a heuristic which minimises the distance between iCub's chest and the object to be manipulated. Figure 6.5 depicts iCub's constrained workspace when managing the torso's roll in the range of ±30 degrees.



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

Figure 6.5: iCub's dual-arm workspace with external management of the torso's DoF roll. Aforementioned pick-and-place task constraints apply: parcel width $d = 250 \pm 5mm$ and orientation error of 0 ± 3 degrees. From left to right: left, front, right and top view.

6.1.4 Demonstration Recording and Learning

Section 6.1.2 has introduced the requirements to succeed on a dual-arm pick-and-place task in the presence of unexpected obstacles. Such a task could be demonstrated in an all-at-once fashion, i.e. a particular demonstration exploiting all skills in order to succeed. This raises concerns about the legibility of the human intentions and hinders teaching the robot through kinesthetic guiding [Bajcsy et al., 2018]. The proposed framework has been designed to address all these challenges by learning the different skills one-at-a-time. Notably, for the commanded

task, three primitive skills are learnt: the underlying dynamics of a pick-and-place task, obstacle avoidance and grasp maintenance (force interaction).

Two strategies have been used to record human demonstrations: kinesthetic guiding on the real iCub humanoid and trajectory demonstration using a trackpad as a haptic device. To conduct kinesthetic teaching, the systems' joints are set in gravity compensation allowing the teacher to physically manoeuvre the robot through the desired skill. During the demonstrations,

Description	**Parameter**	**Equation**	**Value**
PD overall gain	α_x	(3.5)	10
PD configuration gain	β_x	(3.5)	2.5
OD overall gain	α_q	(3.7)	10
OD configuration gain	β_q	(3.7)	2.5
Time scaling factor	τ	(3.5)-(3.8)	1
Grasping geometry left arm	r_L	(3.10)-(3.11)	$[0\ 0.1\ 0]^T$
Grasping geometry right arm	r_R	(3.10)-(3.11)	$[0\ -0.1\ 0]^T$

Table 6.1: Parametrisation of the closed-chaing dual-arm system modelled in Section 3. Note: positional dynamics (PD), orientational dynamics (OD).

Description	**Parameter**	**Equation**	**Value**
Number of RBF	N	(4.1)	35
Canonical system	α_k	(4.1)-(4.3)	1
Weight vector	$\mathbf{w}$	(4.1)	TBL

Table 6.2: Parametrisation of the goal-oriented skill dynamics modelled in Section 4.1. Note: to be learnt (TBL).

Description	**Parameter**	**Equation**	**Value**
OA abruptness	a	(4.6)-(4.7)	TBL
OA sensitivity	c	(4.6)-(4.7)	TBL
OA anticipation	k	(4.6)-(4.7)	TBL

Table 6.3: Parametrisation of the obstacle avoidance behaviour modelled in Section 4.2. Note: obstacle avoidance (OA), to be learnt (TBL).

Description	**Parameter**	**Equation**	**Value**
Error multiplying constant	$\mathbf{K}$	(4.8)	$10\,\mathbf{I}_{6\times6}$
Desired coupling force	$\mathbf{F}_d$	(4.8)	TBL

Table 6.4: Parametrisation of the force interaction skill modelled in Section 4.3. Note: to be learnt (TBL).

proprioception information is retrieved via YARP ports using the built-in `yarpdatadumper` module. The recorded data is already described with respect to the 3D space of the robot.

Limiting the demonstrations to be recorded exclusively with the robot hampers acquiring new data. Alternatively, a MATLAB script has been implemented to read 2D trajectories drawn on a trackpad. Two or more of these demonstrations can be synthesised to obtain 3D demonstrations. To use such information on iCub's architecture, it needs to be scaled to real-world units and referenced with respect to the robot's frame. A parser implemented in C++ lets the robot's YARP-based architecture access to these demonstrations stored as MATLAB variables.

Indifferently from the acquisition method used to record a demonstration, the learning of the different primitive skills and its retrieval has been done with the same parametrisation and at a discretisation frequency of 100 Hz. Table 6.1 details the parameters for the system modelisation. Table 6.2, Table 6.3 and Table 6.4 detail the parameters for the goal-oriented dynamics, obstacle avoidance and force interaction primitive skills, respectively. Such a parametrisation has been used for all experiments reported in the remainder of this chapter.

It is worth highlighting that the intrinsic parameters of the framework weighting each of the primitive skills within its library are all equal to one (see Chapter 5). In the scope of this project, the library is uniquely loaded with the primitive skills required to succeed on the showcase. As it has been previously discussed, the high-level reasoning module according to environment and object affordances is out of this thesis scope but left for future research.

6.1.5 Framework Deployment on iCub Humanoid

Further experimental setup has been required to gather data, test and evaluate the proposal on a real/simulated iCub humanoid robot. First of all, as shown in Figure 6.6, an environment has been set up in the Gazebo simulator according to the pick-and-place task described in Section 6.1.2 and the workspace analysis conducted in Section 6.1.3.

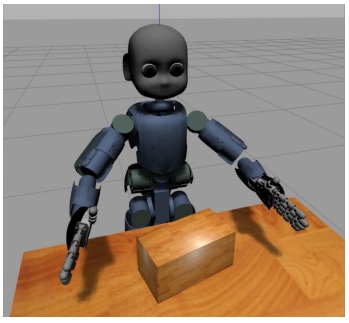


Figure 6.6: Experimental setup of the pick-and-place task and the iCub humanoid in Gazebo.

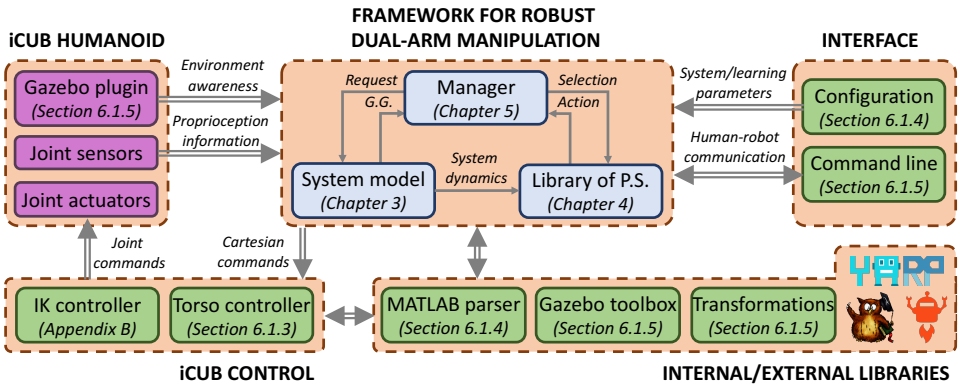


Figure 6.7: Layout of the framework deployment on the iCub humanoid. Note: inverse kinematics (IK), grasping geometry (GG), primitive skill (PS).

The deployment of the entire framework on the iCub humanoid robot has required the design, implementation and integration of some extra components. A layout of this integration is schematised in Figure 6.7. Mainly, three big functional modules can be distinguished: the proposed framework (blue rectangles), the real/simulated platform (magenta rectangles), and the extra components (green rectangles). The theoretical basis of the proposed framework has been detailed throughout Chapter 3, Chapter 4 and Chapter 5. Practically, it also acts as a coordinator of all external elements required for integrating the framework to the iCub platform.

iCub has a built-in YARP architecture which can be exploited to simulate the robot's dynamical behaviour in the Gazebo world. Among all others functionalities of iCub's architecture, there are three that are of the framework's particular interest: the joint sensors of the arms (for learning (see Section 6.1.4) and control purposes), the control of the end-effectors (see Appendix B and Section 6.1.3), and the environment awareness. Instead of retrieving the environment status from the robot's sensors, such information is directly extracted from the Gazebo simulator. For this purpose, it has been implemented a C++ plugin for Gazebo which reports the state of the parcel and any obstacle in the simulated world.

The HRI nature of the proposal requires an interface where the system's model and learning can be parametrised according to Section 6.1.4, and where a human-robot communication can be established. Such duplex communication is set via command line using a YARP `RPC port`. Some of the implemented functionalities via this channel are: retrieval of the system's information, configure different start and goal configurations, check whether a configuration is reachable, change the parcel's size, or, among others, simulate the planned task before execution. This visualisation is displayed in the Gazebo simulator itself, thus addressing the lack of a visualiser in the YARP architecture. For that purpose, an entire toolbox has been implemented to represent trajectories, grasping setpoints, obstacles and the simulated movement of the parcel.

Fully integrating all these modules has only been possible after implementing a library to handle the different requirements regarding data types, reference frames and orientation representations. This comes motivated by Gazebo working with the library Ignition, iCub having its math library implemented within the YARP architecture, and Eigen libraries being an efficient approach to conduct matrices management and computation. Additionally, this self-implemented library also provides all required transformations between frames, e.g. from world to robot coordinates, as well as from the end-effectors' or the object's frame to any other. Moreover, it takes care of the conversions between the representation of rotations in Euler format (required for making the framework understandable and accessible to humans), quaternions (needed for the system modelisation), and axis (required for the iCub control).

Early experimentation on the simulated iCub robot unveiled the lack of realistic simulated force/torque sensors. This fact complicated the control of the exerted force on the carried object. Alternatively, following the approach undertaken in [Gams et al., 2014] for simulated environments, the grasp maintenance skill primitive in Equation (4.8) has been reformulated as:

$$\dot{\mathbf{x}}_{C_i} = \mathbf{K}(\mathbf{D}_{d_i} - \mathbf{D}_{r_i}), \tag{6.1}$$

where, for a workspace $\mathcal{W} = \mathbb{R}^3 \times \mathrm{SO}(3)$, $\mathbf{D}_{d_i} \in \mathbb{R}^6$ is the desired relative distance and orientation between the end-effector i and the carried object and $\mathbf{D}_{r_i} \in \mathbb{R}^6$ is the current state of such features retrieved from the robot's sensors. As in Equation (4.8), $\dot{\mathbf{y}}_{C_i} \in \mathbb{R}^6$ is a vector of velocity commands for the end-effector i to correct the pose errors and $\mathbf{K} \in \mathbb{R}^{6\times6}$ is an error multiplying constant. All in all, instead of learning the required coupling force $\mathbf{F}_{d_i}$ which ensures grasp maintenance, the reference $\mathbf{D}_{d_i}$ has been defined accordingly to the grasping geometry.

6.2 Goal-oriented Skill

This work has modelled the closed-chain dual-arm manipulator as a spring-damper system (see Section 3.3). Such a modelisation let us encode any goal-oriented skills with the DMP-based formulation presented in Section 4.1. The generalisation capabilities of this encoding approach for positional and orientational dynamics are presented next.

6.2.1 Positional Dynamics

The encoding and retrieval of the 3D positional dynamics are done with three DMPs. As introduced in Section 4.1, this approach already offers some generalisation capabilities. Figure 6.8 demonstrates such generability in the pick-and-place context. The demonstrated dynamics (red

trajectory in Figure 6.8) translate the parcel from its initial position at $x_s = [0 \;\; 0 \;\; 0.02]^T$ metres to the configuration $x_g = [0.4 \;\; 0 \;\; 0.03]^T$ metres. As it can be observed, the dynamics move the parcel along the x and z-axis simultaneously until a certain height is achieved. Then, the parcel exclusively moves in the xy-plane until the system is close to the desired xy-configuration. Finally, the parcel is taken down to reach the desired 3D configuration.

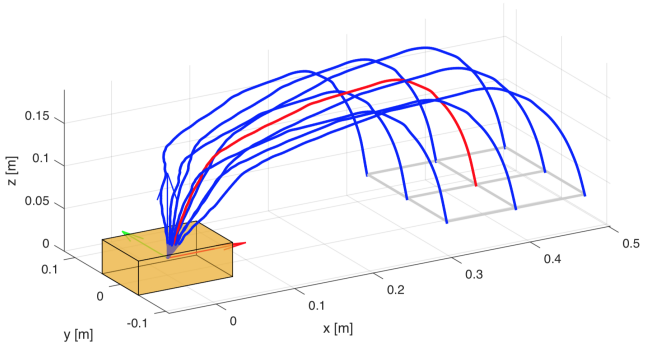


Figure 6.8: DMP generalisation capabilities. Given a demonstration (red trajectory), rolling-out the model in Equation (3.5)-(3.6) with a DMP as coupling term $\mathbf{f}_{o_x}(\cdot)$ lets the system move the box (brown prism) to new goal states by mean of dynamics generalisation (blue trajectories).

The encoded dynamics have been used to infer the demonstration to novel goal configurations, the position of which is within the range of ± 0.1 metres around the provided demonstration. This has lead to the eight roll-outs depicted in Figure 6.8 (blue trajectories). These generalisation capabilities also apply to different start configurations.

6.2.2 Orientational Dynamics

The encoding and retrieval of orientational dynamics is done with quaternion-based DMPs. This approach can handle dynamics involving many DoFs at once and great rotations, such as the ones depicted in Figure 6.9. The demonstrated dynamics rotate the parcel from its initial orientation at $e_s = [0 \;\; 0 \;\; 0]^T$ degrees to the configuration $e_g = [90 \;\; 90 \;\; 0]^T$ degrees. Specifically,

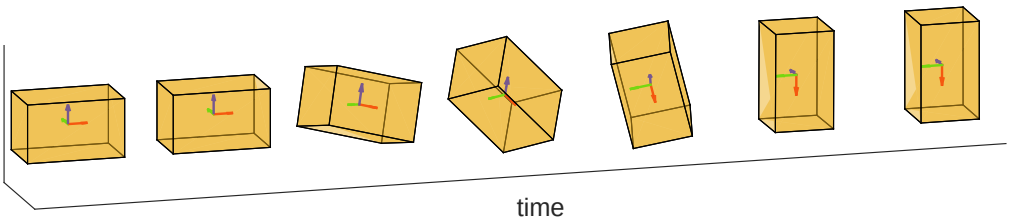


Figure 6.9: Orientational dynamics consisting on rotating a free-floating parcel from the most left configuration $e_s = [0 \;\; 0 \;\; 0]^T$ degrees to the most right configuration $e_g = [90 \;\; 90 \;\; 0]^T$ degrees. Note that no rotation is executed at the first and last part of the demonstration.

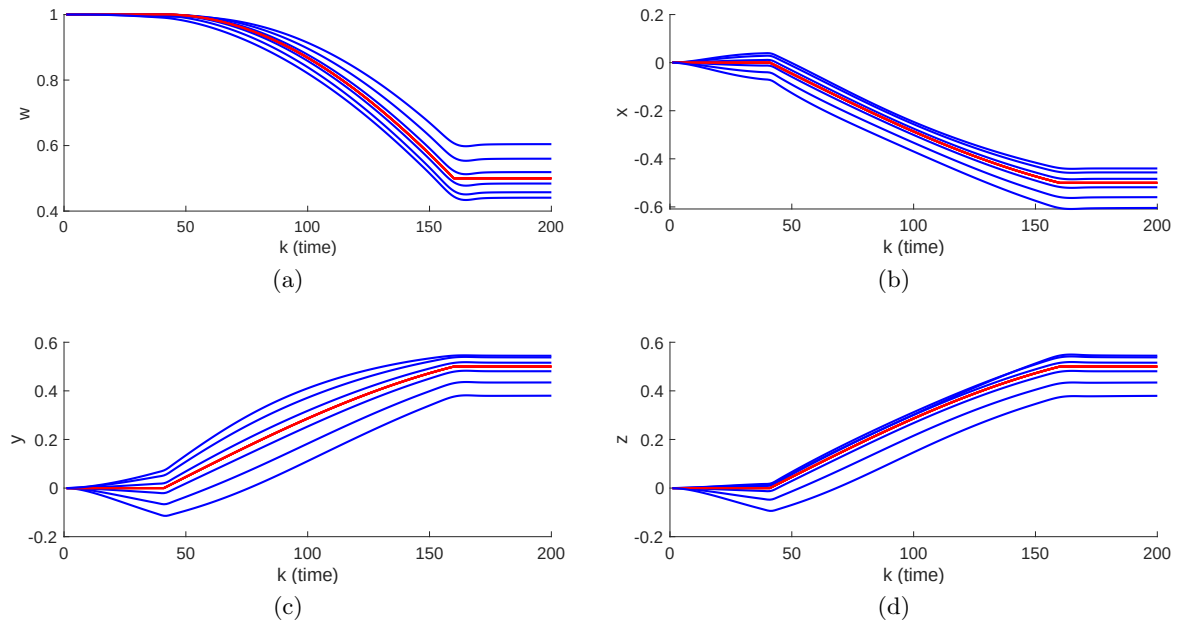


Figure 6.10: Framework's orientational capabilities analysis. Demonstrated orientational dynamics (red lines), inference to undemonstrated orientations in the range ±25 degrees around the demonstration (blue lines). (a)-(d) Profile of each variable of the quaternion $q = [w \;\; x \;\; y \;\; z]^T$.

such rotation uniquely occurs during the middle part of the demonstration corresponding to the 60% of the task's time. This is, the parcel does not change of configuration during the first and last 20% of the task's time. Such orientational dynamics are depicted as red lines in Figure 6.10.

The learnt model has allowed inferring the demonstrated dynamics to novel goal orientations within the range of ±25 degrees around the provided demonstration (see blue lines in Figure 6.9). These generalisation capabilities also apply to different start configurations.

6.3 Obstacle Avoidance Skill

The theoretical formulation and the corresponding advantages of the proposed obstacle avoidance skill have already been demonstrated in Section 4.2. This section aims to show its suitability to reproduce obstacle avoidance behaviours learnt from humans demonstrations. To this aim, such a primitive skill has been taught to the real iCub with two different behaviours: reckless (see Figure 6.11a) and conservative (see Figure 6.11b). While the former steers around the obstacle (red sphere) closely, the latter keeps a larger distance to it. The recorded raw proprioception data of these two kinesthetic demonstrations is respectively portrayed in Figure 6.11c and Figure 6.11d. As it can be observed, the retrieved trajectories are noisy and not smooth.

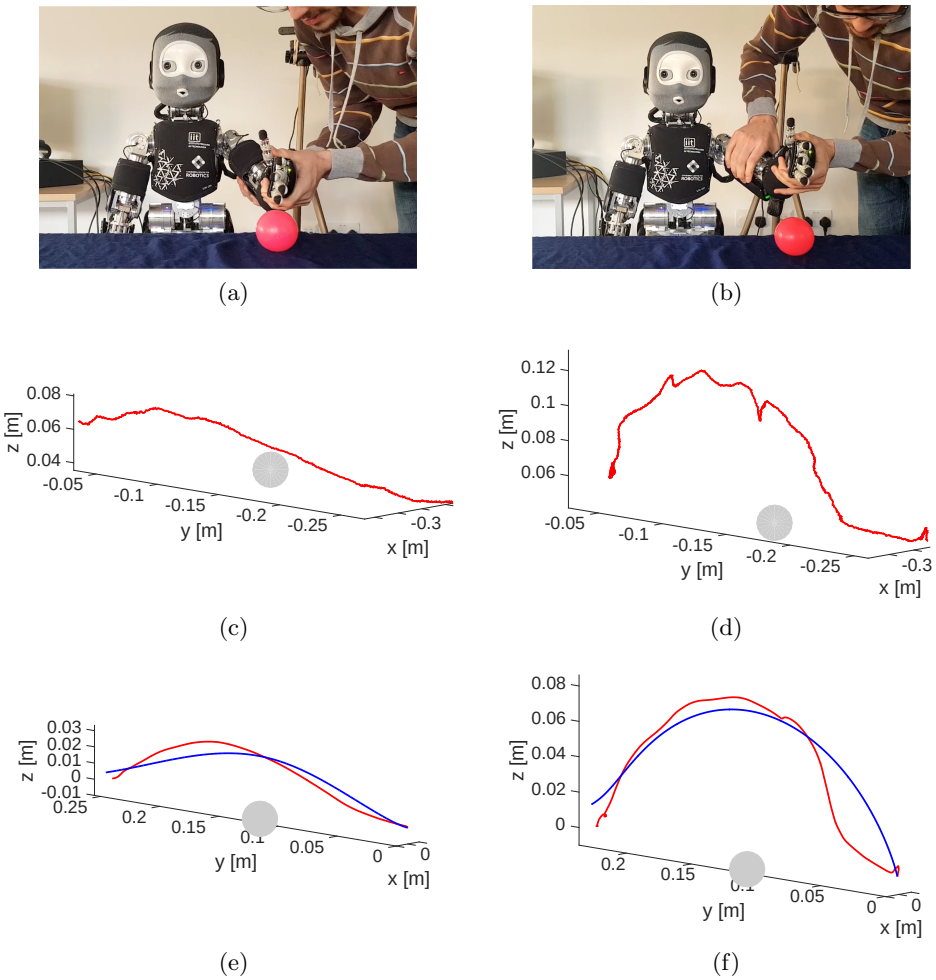


Figure 6.11: iCub humanoid robot learning the primitive skill of obstacle avoidance with two different behaviours: reckless (first column) and convervative (second column). (a)-(b) Human demonstrations to avoid an obstacle (red sphere). (c)-(d) iCub's proprioception data. (e)-(f) Processed proprioception data (red trajectory) and learnt behaviour (blue trajectory).

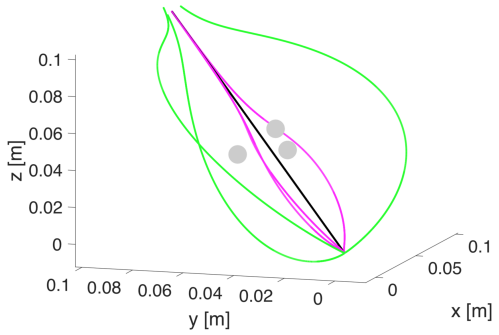


Figure 6.12: Generalisation capabilities to multiple obstacles and in 3D scenarios of the learnt reckless (magenta trajectory) and conservative (green trajectory) obstacle avoidance behaviours.

To learn from these demonstrations, the data has been preprocessed in two steps: (i) filtering to remove outliers and high-frequency noise, and (ii) projecting the resulting information to the 2D space defined by the two principal components of the data. Figure 6.11e and Figure 6.11f show the preprocessed data (red trajectories), which has been used in Equation (4.7) to learn the parameters defining the demonstrator's obstacle avoidance behaviour. The encoded reckless and conservative styles are respectively depicted in Figure 6.11e and Figure 6.11f (blue trajectories). Note that learning the parameters instead of the motion itself lets the robot generalise such behaviour under different conditions and multiple obstacles (see Figure 6.12).

Figure 6.11 and Figure 6.12 point out the suitability of the proposed mathematical formulation to encapsulate, reproduce and generalise the demonstrator's style on avoiding obstacles. Discrepancies between the demonstrated and retrieved behaviours are attributed to the noise in the proprioception data, which increases the variance in the learning stage. Alternatively, a high-precision tracking system such as the one used in [Rai et al., 2014] shall be considered. Because the proposed approach extracts the parameters of a demonstrated obstacle avoidance behaviour, other approaches than kinesthetic guiding can be employed for teaching this primitive skill.

6.4 Framework Evaluation

The entire framework conducting dual-arm pick-and-place tasks with the presence of obstacles has been evaluated in both synthetic and simulated environments. The former scenario lets testing the concept of the framework without the inherent robotics-constraints, such as reduced workspace and the uncertainties present in the proprioception data and control. The latter case is to demonstrate the applicability of the proposed framework in the iCub humanoid robot.

6.4.1 Evaluation on Synthetic Environments

Evaluation of the framework on the pick-and-place setup has been first carried out in a synthetic environment to avoid the inherent robotics-constraints. The lack of world-like physics do not allow simulating interaction forces, hence limiting the test and analysis of the framework on three absolute skills: goal-oriented dynamics (both positional and orientational) and obstacle avoidance behaviour. The resulting response of the framework is depicted in Figure 6.13.

Initially, a pick-and-place demonstration (red trajectory) is given to the system using a trackpad as a haptic device (see Section 6.1.4). It consists of moving the parcel from the left to the right of the workspace, without generating any rotation and without the presence of the obstacle (grey prism). As discussed in Section 4.1 and later depicted in Figure 6.8, encoding the positional

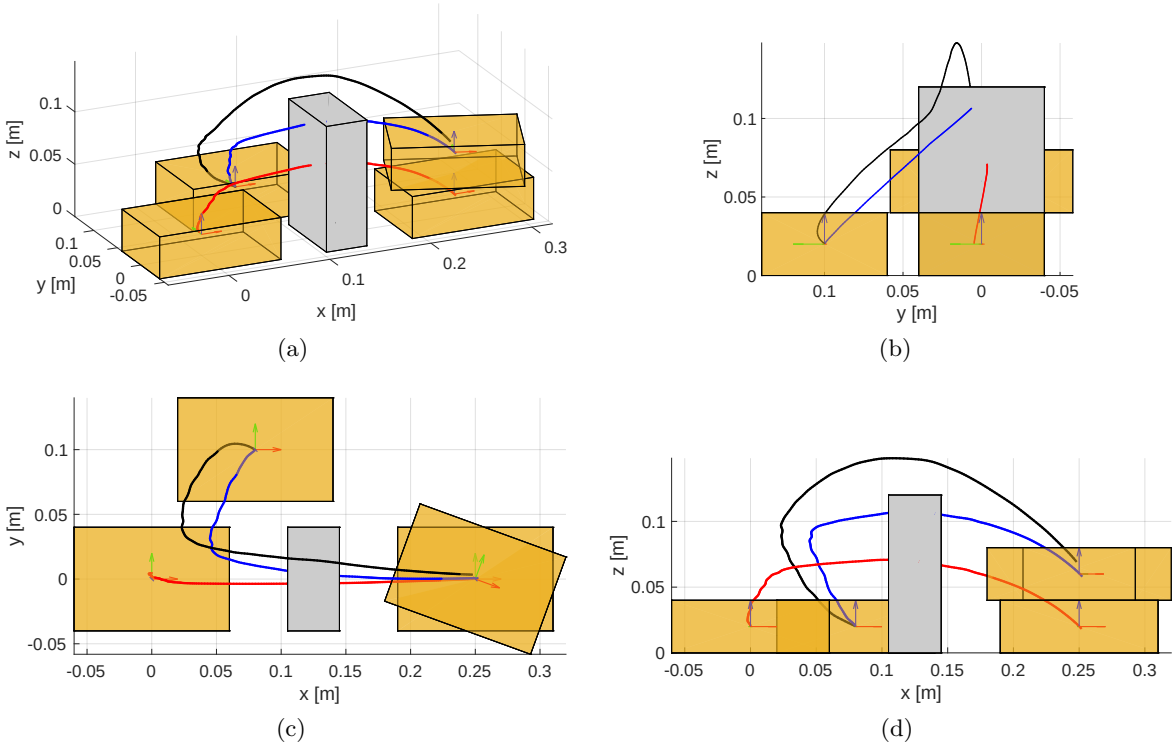


Figure 6.13: Dual-arm pick-and-place of a parcel (brown prism) in the presence of obstacles (grey prism). Demonstrated task (red trajectory), infered task (blue trajectory), infered task with obstacle avoidance (black trajectory). The composition of primitive skills lets the system generalise to unfamiliar environments. (a) Perspective, (b) lateral, (c) top, and (d) front view.

and orientation dynamics underlying the execution of this task in a DMP formulation already endows the system with some inherent generalisation capabilities (blue trajectory). This lets the robot to infer the pick-and-place with small variations in starting and goal positions and orientation (see Table 6.5). However, the generalisation capabilities are yet limited and unable to generalise to the presence of obstacles. It is only after coupling the previously learnt obstacle avoidance behaviour (see Section 6.3) and the pick-and-place dynamics together that the system can generalise in real-time to the presence of unexpected obstacles (black trajectory).

	Start configuration	**Goal configuration**
Demonstrated scenario	$x_s = [0 \;\; 0 \;\; 0.02]^T$ [m] $e_s = [0 \;\; 0 \;\; 0]^T$ [deg]	$x_g = [0.25 \;\; 0 \;\; 0.02]^T$ [m] $e_g = [0 \;\; 0 \;\; 0]^T$ [deg]
Unfamiliar scneario	$x_s = [0.08 \;\; 0.1 \;\; 0.02]^T$ [m] $e_s = [0 \;\; 0 \;\; 0]^T$ [deg]	$x_g = [0.25 \;\; 0 \;\; 0.06]^T$ [m] $e_g = [0 \;\; 0 \;\; -20]^T$ [deg]

Table 6.5: Summary of start and goal configurations (3D position and Euler XYZ orientation) of the demonstrated and unfamiliar scenarios reported in Figure 6.13.

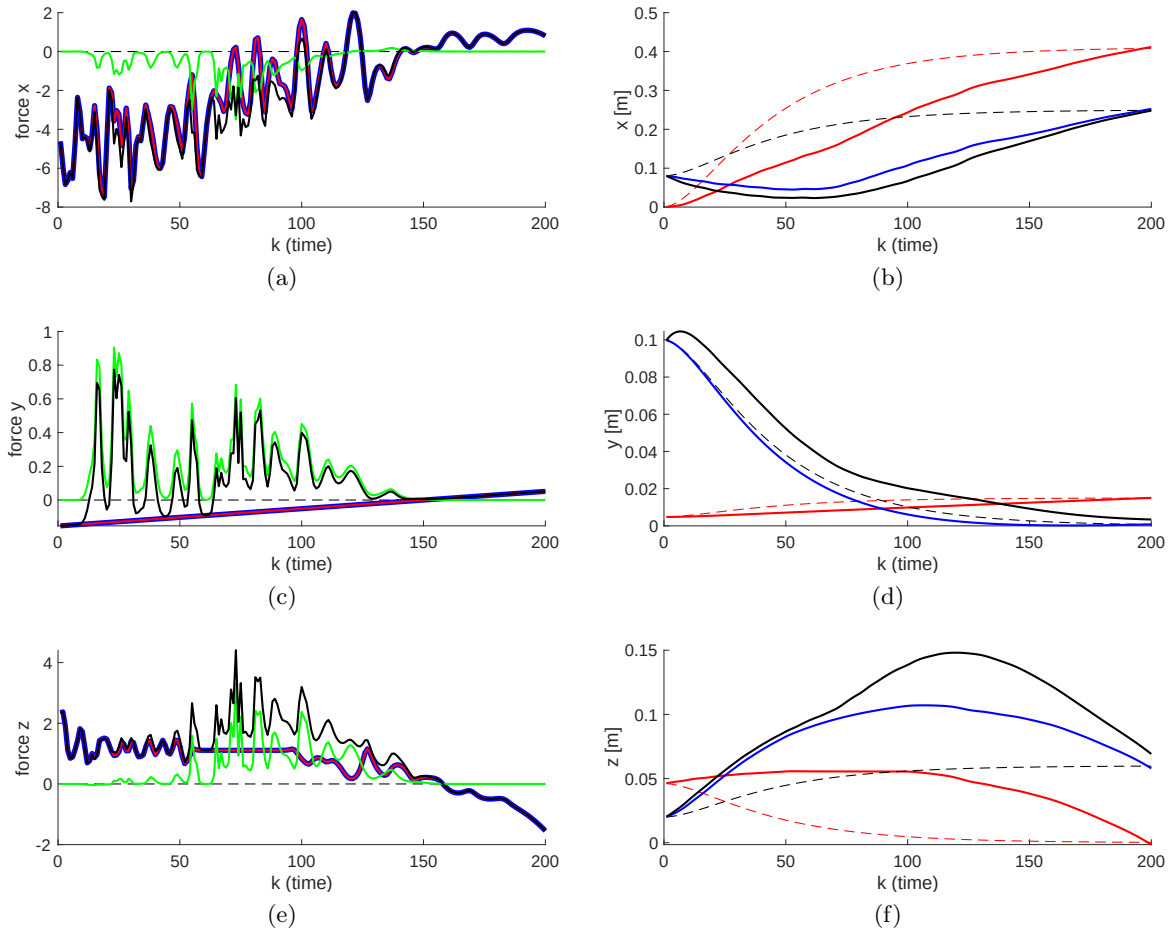
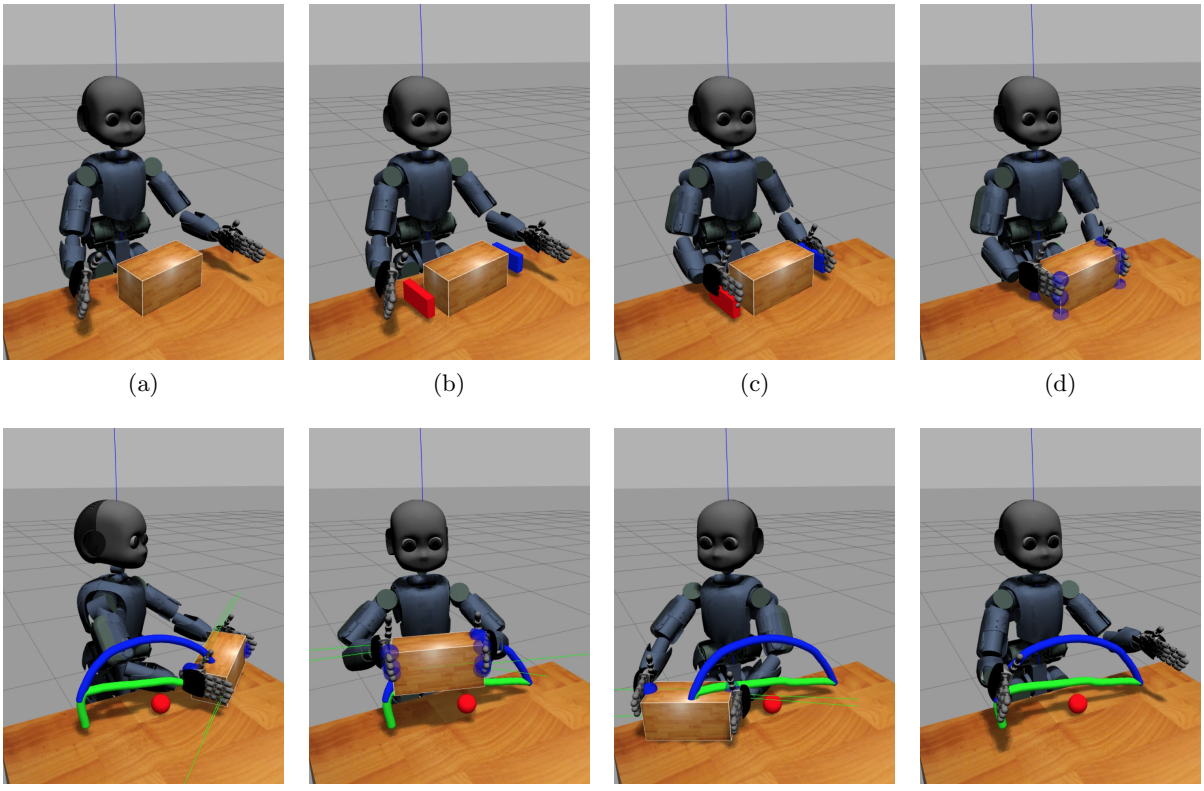


Figure 6.14: Framework analysis in Figure 6.13 scenario. System's natural dynamics (dashed lines), demonstrated task (red lines), infered task (blue lines), obstacle avoidance (green lines), and infered task with obstacle avoidance (black lines). First column: primitive skills at the force level. Second column: forces affect at the Cartesian space. Top to bottom: x, y and z-axis.

The framework's behaviour observed in Figure 6.13 is the result of different primitive skills acting at the same time. Figure 6.14 depicts their interaction at the force level (first column) and affect in the 3D Cartesian space (second column) along the aforementioned task. The colour code of these plots is as in Figure 6.13. When the natural dynamics of the system are not modified, i.e. the total virtual external force (coupling term) is null, the system reaches the goal configuration with spring-damper dynamics (dashed lines). Since the demonstrated behaviour does not follow such dynamics, the virtual force is not null (red lines). This force profile has been encoded as a DMP to learn the demonstrated primitive skill. This knowledge already lets the system to generalise the demonstrated dynamics to different start and goal configurations (blue lines). The successful roll-out which generalises in front of obstacles (black lines) is obtained after considering the obstacle avoidance primitive skill available in the framework (green lines).

6.4.2 Evaluation on a Simulated iCub Humanoid

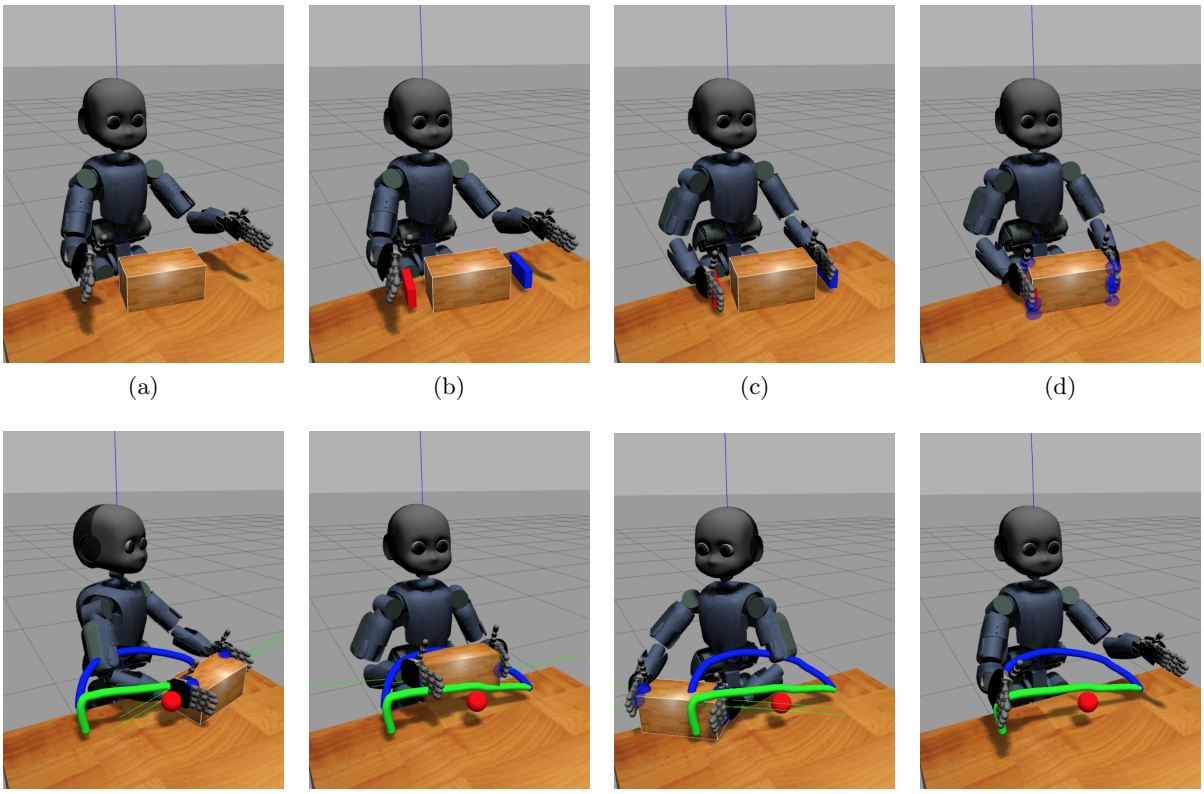
The framework's robustness in undemonstrated scenarios has already been analysed in the context of synthetic environments in Section 6.4.1. This section shows the applicability of the proposal in a robotic platform, particularly on the simulated iCub robot. After the set up procedure reported in Section 6.1.5, iCub has been taught in a one-at-a-time fashion four primitive skills: (i) pick-and-place in a sawtooth shape fashion starting at $x_s = [0.3 \;\; 0.25 \;\; 0.56]^T$ and finishing at $x_g = [0.3 \;\; -0.25 \;\; 0.56]^T$ (encoded as a positional goal-oriented dynamics), (ii) rotational motion of 20 degrees around the z-axis with dead-ends as in Section 6.4.1 (encoded as an orientational goal-oriented dynamics), (iii) obstacle avoidance with conservative behaviour as in Section 6.3, and finally, (iv) grasp maintenance by means of the parcel's grasping geometry as defined in Section 6.1.5. All these primitive skills have been loaded in the framework's library to be later exploited according to the tasks requirements introduced next.



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

(e) (f) (g) (h)

Figure 6.15: iCub humanoid robot exploiting the demonstrated pick-and-place task (green trajectory) to succeed (blue trajectory) in *Scenario-1* which has an obstacle (red sphere) at $x_o = [0.3 \;\; 0 \;\; 0.6]^T$ metres. (a) Parcel initial state, (b)-(d) grasping parcel laterally, (e)-(g) simultaneously exploiting some primitive skills to successfully conduct the pick-and-place task in an undemonstrated scenario, and (h) overview of the trajectory adapted in real-time.



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

(e) (f) (g) (h)

Figure 6.16: iCub humanoid robot exploiting the demonstrated pick-and-place task (green trajectory) to succeed (blue trajectory) in *Scenario-2* which has an obstacle (red sphere) at $x_o = [0.25 \;\; 0 \;\; 0.55]^T$ metres. (a) Parcel initial state, (b)-(d) grasping parcel laterally, (e)-(g) simultaneously exploiting some primitive skills to successfully conduct the pick-and-place task in an undemonstrated scenario, and (h) overview of the trajectory adapted in real-time.

Initially, the parcel is located on the table in configuration $x_p = [0.3 \;\; 0 \;\; 0.56]^T$ metres (with a random variation of ± 5 centrimetres on the xy-plane) and $e_p = [0 \;\; 0 \;\; 0]^T$ degrees (with variation of ± 10 degrees around the z-axis). This leads the parcel in front of the robot with a random pose but still accessible to the robot workspace (see Figure 6.15a and Figure 6.16a). Given this environment setup, the commanded task consists of (i) pick the parcel regardless its initial random configuration, (ii) move it to the configuration $x_s = [0.3 \;\; 0.25 \;\; 0.56]^T$ metres $e_s = [0 \;\; 0 \;\; 18]^T$ degrees (iCub's left side) and (iii) place the parcel to the configuration $x_g = [0.3 \;\; -0.25 \;\; 0.56]^T$ metres $e_g = [0 \;\; 0 \;\; -18]^T$ (iCub's right side). Only the latter stage of the task requires iCub to avoid an obstacle located at $x_o = [0.25 \;\; 0 \;\; 0.55]^T$ metres for *Scenario-1* (see Figure 6.15), and located at $x_o = [0.3 \;\; 0 \;\; 0.6]^T$ metres for *Scenario-2* (see Figure 6.16).

The former stage of the task, i.e. grasping the parcel, is completed by first retrieving the parcel's random configuration, then use the grasping geometry to compute the desired grasping points,

and finally approach them laterally via the middle-setpoints displayed as red and blue prisms for the right and left end-effector, respectively (see Figure 6.15a-Figure 6.15d for *Scenario-1* and Figure 6.16-Figure 6.16d for *Scenario-2*). From this stage on, the primitive skill for grasp maintenance ensures that both end-effectors are in flat contact with the box. Contact points between these elements are displayed with the small blue spheres visible in Figure 6.15d-Figure 6.15g and Figure 6.16d-Figure 6.16g for *Scenario-1* and *Scenario-2*, respectively.

The two latter stages of the task are completed exploiting the learnt sawtooth-shaped pick-and-place while, at the same time, ensuring grasp maintenance. During the first movement of the task, i.e. moving from the configurations depicted in Figure 6.15d and Figure 6.16d to the ones in Figure 6.15e and Figure 6.15e, there is not any obstacle. Consequently, the built-in DMPs generalisation capabilities shown in earlier sections of this manuscript are sufficient to address this pick-and-place task. However, the latter movement of the task involves adapting the learnt dynamics to avoid an obstacle. In *Scenario-1*, since the obstacle (red sphere) is collinear with the start and goal positions, i.e. below the demonstrated task (green trajectory), the framework makes the robot circumnavigate the obstacle by the top of the obstacle (see Figure 6.15e-Figure 6.15g). Instead, since the location of the obstacle in *Scenario-2* is further from the robot than the previous obstacle, the framework guides the system through a collision-free trajectory near iCub's chest (see Figure 6.16e-Figure 6.16g).

These two experiments executed with the simulated iCub humanoid robot have demonstrated various of the aforementioned framework's features. Having a repertoire of primitive skills available in the framework's library allows exploiting them both simultaneously and sequentially to obtain composed and complex tasks such as the ones reported in this section. The considered scenarios involved real-time adaptation capabilities and using the self-implemented robot's torso controller to reach distant configurations. In all examples, the robot's behaviour was successful in performing the commanded dual-arm pick-and-place task, while avoiding obstacles and ensuring grasp maintenance and synchronisation. All in all, these experiments have shown the applicability and suitability of the designed framework for humanoid robots.

Chapter 7

Final Remarks and Future Work

This work has presented a novel end-to-end framework which endows a dual-arm system with real-time, robust and less task-specific manipulation capabilities. Such an architecture is twofold: (i) learns from human demonstrations to create a library of primitive skills, and (ii) combines such knowledge to confront challenging unfamiliar scenarios with human-like manipulation capabilities. Unlike the framework of motion primitives in [Pastor et al., 2009], the proposed approach handles primitive skills for dual-arm manipulation purposes while still being able to combine different primitives at the same time. This feature is what differentiates the current work from other state-of-the-art dual-arm oriented frameworks [Topp, 2017; Zöllner et al., 2004]. The evaluation conducted on the iCub humanoid suggest the proposal's suitability for robust dual-arm manipulation, yet with some room for improvement.

The framework is not restricted to the presented experimental evaluation nor platform. Any system able to retrieve proprioception information can benefit from this work. Moreover, any primitive skill that might be required for dual-arm manipulation can be included in the framework's library. The application case reported in this manuscript exemplifies this fact by considering, four types of primitive skills: positional and orientational goal-oriented skill dynamics, obstacle avoidance behaviour and force interaction for grasp maintenance. The obstacle avoidance behaviour which steers around obstacles in real-time was originally presented in [Fajen and Warren, 2003], later reformulated in [Rai et al., 2017], and further enhanced in this work to address the dead zone issue and discard distant obstacles. The desired performance of this skill is learnt from human demonstrations using kinesthetic guiding on the real iCub humanoid.

The execution of this work has approximately followed the a priori planning presented in the research proposal. Achieving the proposal's goal has been possible after working around some

unforeseen events happening during this master thesis. Firstly, the chance of presenting the overall idea of this work in the 2018 AAAI Artificial Intelligence for Human-Robot Interaction (AI-HRI) symposium, which involved writing a paper while conducting the experiments for this master thesis. Secondly, the short notice of iCub's reduced dual-arm workspace forced extending the system's modelisation and learning to also contemplate orientational information. Even though this feature was not in the original proposal, it allowed exemplifying the pick-and-place task even in iCub's high-constrained workspace.

Future work will significantly extend the library of primitive skills such that more tasks and scenarios involving challenging dual-arm manipulation behaviours can be addressed within the framework. In this regard, imminent efforts will focus on exploiting the force/torque sensors of the iCub humanoid robot to learn force-dependant primitive skills, such as the grasp maintenance one, or other actions requiring complex synchronisation between end-effectors, such as the opening of a bottle's screw cap or succeeding in the peg-in-a-hole tasks. Then, action selection will be integrated to automatically select from the framework's library the necessary set of skills to conduct a particular task subject to the environment and object affordances. In these lines, Ardón et al. jointly exploits the objects and environmental semantic features to infer the best grasping point [Ardón et al., 2018]. This idea can be extended to deal with the aforementioned action selection requirements of the proposed framework.

Another potential direction is the evaluation of the framework from a HRI perspective. For that purpose, non-robotics-experts will assess through questionnaires their experience with teaching the iCub humanoid robot in a one-at-a-time versus all-at-once fashion. The expected outcome is to qualitatively determine the ease of endeavour that the one-at-a-time demonstration baseline supposes to naive users. This social experiment will also require the participants and the robot to conduct a task in a novel scenario. This data will allow to quantitatively evaluate the human-like similarity of the framework's outcome using the aforementioned introduced KL divergence statistic. All in all, this social study seeks to assess the effect of learning composable skills to increase the acceptability and compatibility of robots in human workspaces.

The integration and evaluation of all these components into the framework constitutes the primary roadmap for the PhD thesis.

Appendix A

Apprenticeship Learning: A Survey

The manuscript attached in the next pages is an extensive survey on apprenticeship learning. It is still under preparation, but expected to be presented in *The International Journal of Robotics Research* after the completion of the present master thesis.

Apprenticeship Learning: A Survey

Èric Pairet[1], and Frank Broz[1]
ep18@hw.ac.uk, f.broz@hw.ac.uk

***Abstract*— Real-world robots are becoming a vital ingredient in society. Not only they are required to live in environments originally designed for humans but also to perform human tasks. Hence, it is reasonable to expect robots to learn and master some skills as we humans do: from demonstrations and/or sharing information (social learning) and with practice (self-learning). In the last decades, many efforts have been focused on exploiting the previous approaches. However, in an attempt to have more intelligent agents, combining both approaches becomes essential. This is called apprenticeship learning, which attempts to endow robots with enhanced learning, generalisation and scalability capabilities. No surveys on apprenticeship learning reflecting their breakthroughs in the learning field have been conducted yet. In this paper, we use the human learning paradigm to motivate the review of the current status of learning in robotics. Specifically, we focus on apprenticeship learning methods and its strengths in opposition to traditional learning approaches. By analysing some of these works we exhibit its potential for providing robots with enhanced autonomous capabilities. To conclude we discuss current limitations in the apprenticeship learning development and mention promising areas for future research.**

***Index Terms*— Robot Learning, Intelligent Robots, Autonomous Systems, Reinforcement Learning, Learning by Demonstration, Inverse Reinforcement Learning, Transfer Learning.**

I. INTRODUCTION

Real-world robots are required to perform their tasks in human-like environments. This fact has lead to an increase of interest in the development of systems such as robotic legs, prosthetic hands, robotic arms and, on a larger scale, humanoid robots. Nonetheless, with humanoid robots arise many technological challenges. Bohren et al. discusses that endowing a humanoid robot with autonomy requires the integration of many complex subsystems such as perception, reasoning, navigation, motion planning and grasping, among others [10]. Even though these components have been extensively validated individually during the last decades, integrating them into a robust functional system is still an active area of research.

Traditional approaches governing these complex systems require a great understanding of the model underlying the system's behaviour. Even though deriving an accurate model is possible for some systems, approximations are commonly used in order to make the calculations computationally tractable, despite the trade-off of the model's accuracy. Furthermore, these models might also require hand-defining all possible scenarios, movements, tasks and extensive manual tuning of the system's control architecture [8]. Therefore, this traditional approach lacks scalability and generalisation.

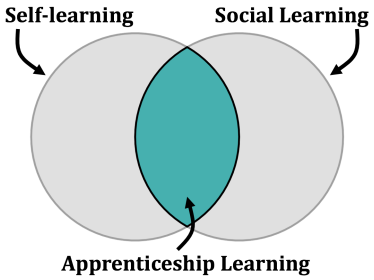


Fig. 1: This work focuses on apprenticeship learning, i.e. the interaction between self-learning and social learning techniques for robotics.

With the popularisation of artificial intelligence (AI), more natural methods for robot learning have been adopted, reducing the laborious task of coding every possible scenario and thus, increasing modularity and flexibility on the systems. This allows non-robotics-experts to interact, teach and modify the robot's behaviours [38]. In an attempt for these systems to work in a more human-like manner they involve learning from (and as) humans, especially when learning motions, i.e. the kinematics, dynamics and constraints describing the functionality of the robot's actuators. In the recent years, this has become an important research topic in the robotics community.

Given the expertise of humans in interacting with the environment, it is natural to study humans' motions to use the resulting knowledge in robotic control. In this survey, we review different approaches for robot task learning using as motivation and analogy the human learning paradigm: demonstrations, practice and sharing [19]. We generalise this paradigm with two learning alternatives: social learning and self-learning. The integration of both approaches on an agent results on apprenticeship learning (see Figure 1). This analogy makes this survey different from machine learning-based papers that exclusively review the state-of-the-art on general human-robot interaction (HRI) [15], specific methods for programming intelligent systems [7], self-learning techniques (reinforcement learning (RL) [23, 25]), social learning techniques (learning by demonstration (LbD) [2, 8], and transfer learning (TL) [49]), among others. Furthermore, despite the difficulty of defining a boundary between machine learning and the control theory, modelisation theory and purely control-based approaches are outside the scope of this review. Excellent surveys in these subjects can be

[1]Robotics Lab, School of Mathematical and Computer Sciences at Heriot-Watt Univeristy, UK.

found in [11, 37, 42]. Instead, this article emphasises the contributions and advantages of hybrid learning techniques, i.e. apprenticeship learning, to provide an overview of a new, growing area of research.

The remainder of the paper is structured as follows. The human learning paradigm that motivates this paper as an analogy of the current state-of-the-art about apprenticeship learning in robotics is stated in Section II. Then, Section III introduces the foundations of apprenticeship learning, focussing on their individual strengths and flaws. Section IV reviews and categorises existing approaches on apprenticeship learning. Finally, some conclusions and interesting future works are discussed in Section V.

II. LEARNING PARADIGM

The current state-of-the-art of learning in robotics keeps some similarities to the way in which humans and animals learn. The latest has been extensively studied in many fields such as psychology [13], pedagogy [30], biology [31] and neuroscience [33], among others. At the same time, each field has different theories about the learning in biological systems. In this section we introduce the analogy of the learning concept from a biological point of view (see Section II-A) with the current state-of-the-art about learning in robotics (see Section II-B). Furthermore, we introduce some notation to formalise the paradigm in the robotics context.

A. Learning in Animalia

Humans and animals (agents) evolve over the years thanks to the inherent capacity of learning. From a biological understanding, an agent acquires knowledge by interacting with other agents and develops knowledge from own experiences [19]. These learning capabilities let an agent learn what other agents know and also, to master a specific skill by practising it. Formally, we can distinguish the following individual approaches of learning:

- **Self-learning:** experimentation with the environment provides agents with some feedback about the suitability of their actions. It is also referenced as associative learning, asocial learning and individual learning.

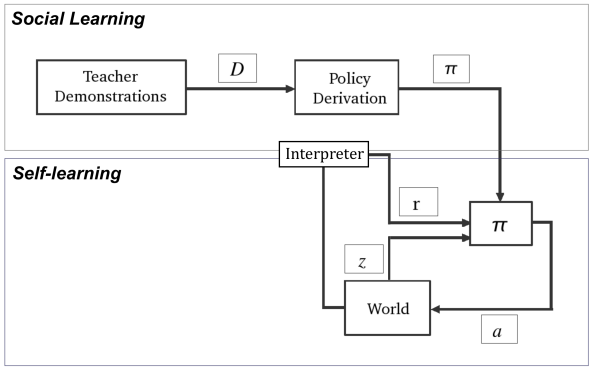


Fig. 2: Learning paradigm in robotics. Both social and self-learning can have an influence on the robot learning, i.e model of the task encoded in a policy π. Adapted from [2].

- **Social learning:** interaction with external agents and observation of other's actions provide agents with examples of what, when, how, and why to do a task. It is also referenced as observational learning and group learning. Therefore, there are two concepts in social learning:
  - **Adoption:** perceived knowledge might be sufficient to use it as it is. Thus, it is adopted without the need of understanding.
  - **Imitation:** perceived knowledge needs to be understood in order to replicate it. Thus, it needs to be processed and generalised.

Any of the previous individual learning approaches can allow an agent to obtain the skills perform a task. However, it is known that humans and animals use these approaches in conjunction. This modular structure broadens the alternatives for learning, being able to make any possible combination of the individual learning resources. In order to clarify this idea, let's analyse a typical real-world scenario: *A coach teaches to three students how to score in basketball, let's call them A, B and C. After some demonstrations, each student builds an understanding of the basketball rules and technique. Student A grasps concepts correctly and is able to score straightaway from any part of the court. Student B got the rules and an intuition of how to technically proceed, but he/she needs a bit of practice before being skilled enough. Instead, student C neither understood the rules nor the technique, and even after some hours of self-study, is unable to score. Because of that, he/she asks for help from classmates and as a result builds an understanding of the rules allowing him/her to progress during self-study. To settle the new ideas, he/she asks to validate the technique with the coach.*

As exemplified in the previous scenario, the process of learning and teaching among humans can be extremely complex and challenging: lack of examples, misunderstandings, loss of information and incapacity of self-learning are just some of the difficulties that humans might face in the learning process. Considering that the human learning process is of such sophistication that it took million of years of evolution to develop, it is understandable that replicating this process in robotics is such a struggle. Actually, because learning in robotics has taken great inspiration from learning in humans and animals, some of the previously mentioned challenges are also present in the field.

B. Learning in Robotics

Similarly to the learning paradigm in humans and animals, robotics agents also interact with other agents (social learning) and practice (self-learning) to gather and improve their skills. In a formal way, the learning paradigm in robotics (see Figure 2) consists mainly of three components:

- **World:** the environment is composed by states $\mathbf{s} \in \mathcal{S}$. Any agent can interact with it through actions $\mathbf{a} \in \mathcal{A}$. A mapping between states by way of actions is defined by a probabilistic transition function $\mathbb{T}(\mathbf{s}' \,|\, \mathbf{s}, \mathbf{a})$: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$. This transition function $\mathbb{T}$ determines the dynamics of the system being controlled [42].

- **Agent:** the agent's actions given a state are commanded by a model or policy π: $\mathcal{S} \rightarrow \mathcal{A}$. The new state $\mathbf{s}'$ is assumed to not be fully observable, but an agent instead has access to the observed state $\mathbf{z} \in \mathcal{Z}$, through the mapping $\mathbb{M}$: $\mathcal{S} \rightarrow \mathcal{Z}$. The agent might also get a reinforcement signal $\mathbf{r} \in \mathcal{R}$ which evaluates the suitability of its actions, i.e. the policy π.
- **Teacher:** the teacher might provide to the agent a set of demonstrations $\mathbf{d}_j \in \mathcal{D}$ consisting on k_j pairs of observations and actions: $\mathbf{d}_j = \left(\mathbf{z}_j^i, \mathbf{a}_j^i\right)$, where $\mathbf{z}_j^i \in \mathcal{Z}$, $\mathbf{a}_j^i \in \mathcal{A}$, $i = 0 \dots k_j$. This step involving HRI might raise some challenges depending on the technique used to teach the robot [2].

Further to these three components, there is an additional one called the **interpreter**. It reports with a reinforcement signal $\mathbf{r} \in \mathcal{R}$ the suitability of the new state $\mathbf{s}'$ (or new observation $\mathbf{z}'$) as a result of applying $\mathbf{a}$ in $\mathbf{s}$. The interpreter could be an external observer or an internal function from the agent. Thus, the observability of the states might differ depending on the nature of the interpreter. An interesting discussion of reward function origins can be found in [44].

The similarity of learning in robotics is not only in the interaction between the environment and the agents, but also in the way that robots can learn. Robots can not only learn from demonstrators and ask for help when needed (social learning) and from their own exploration (self-learning), but also they can make use of both approaches when needed. Then, the previous examples with robots instead of human students would analogously imply: (a) *robot A would satisfactorily learn a policy π from human demonstrations $\mathcal{D}$ using social learning techniques which lets it score from any part from the court*, (b) *robot B would partially derive a suitable policy π from human demonstrations $\mathcal{D}$ using social learning techniques and would improve the policy π with self-learning techniques until the reward $\mathbf{r}$ is favourable, i.e. success rate when trying to score from any position of the court*, and (c) *robot C would derive a useless policy π from human demonstrations $\mathcal{D}$ using social learning techniques, which even after some trial and error on the court (self-study) does not converge to a useful policy, for what it needs is to adopt other's policies and new human demonstrations to correct and generalise the understanding*.

The paradigm set in this section establishes the route for reviewing the state-of-the-art about learning in robotics. While most of the techniques uniquely explore some of the presented interactions between agents [38], there is some work that exploress two or more interactions, i.e. undertaking the apprenticeship learning approach. As it will be seen, this latest approach leads to a more effective teaching and learning techniques.

III. APPRENTICESHIP LEARNING FOUNDATIONS

Apprenticeship learning stands on social and self-learning's shoulders. Thus, all the respective methods' strengths, limitations and challenges become the foundations of apprenticeship learning. Following the already introduced learning paradigm and notation, the principles of self-learning and social learning techniques are respectively explored in Section III-A and Section III-B.

A. Self-learning

An invaluable source of knowledge is one's own experiences. The outcome of one's previous actions when facing a specific situation give an intuition (feedback) of one's performance. This feedback then allows for the reaction method to adapt to successfully achieve a specific goal. The first approaches to emulate this behaviour in machines were using auto-tunable control algorithms back to the 1950s. However, it was not until the late 80s, with the appearance of machine learning, that control theory matured to conceive of adaptive control [12, 45]. Details of this approach are outside of the scope of this paper, but an excellent review can be found in [3].

According to Sutton et al., adaptive control evolved into what is now know as reinforcement learning (RL) [47]. Its beguilement lies in letting agents explore the policy π which leads them to accomplish a task without the need to implicitly specify how it needs to be achieved. Instead, the agent obtains reinforcements $\mathbf{r} \in \mathcal{R}$ (either positive or negative reward) depending on the suitability of its actions within the task meant to be learnt. A more extensive explanation of RL's theoretical basis can be found in [23, 25].

There is an innumerable amount of research using RL that has succeeded in letting robots learn by themselves. However, this approach also faces challenges and limitations such as the following:

- **Lack of generalisation:** RL approaches typically do not generalise their knowledge over the state-action space. Thus, demonstrations must be provided for every discrete state. This lack of generalisation over the state-action space leads to the well-known exploration vs. exploitation dilemma.
- **Struggles with continuous spaces:** RL was originally conceived for discrete spaces. Even though some work deals with continuous tasks, discretisation of the space is the most preferred approach.
- **Curse of dimensionality:** the data and computation needed to cover the complete state-action space increase exponentially as the dimensionality of the system and discretisation resolution increase. The term "curse of dimensionality" was coined by Bellman back in 1957 [4].
- **Exploration with real systems:** RL needs to explore the state-action space. Physically executing all actions from every state is likely to be infeasible, dangerous, and not to scale with continuous state spaces [2]. Moreover, robots might suffer from wear and tear.
- **Reward function modelisation:** defining a reward function to correctly guide the system as to which actions suit the task meant to be learned is not always obvious. Inaccurate reward functions can lead the system to not learn.

A common consequence of all previous issues is the long exploration time when RL is applied on real robotic systems. This is often characterised by high dimensional state

and action spaces, due to the many degrees of freedom of modern robots. Some authors have dealt with this by looking at lower-dimensional space representations of the problem. However, this form of a state dimensionality reduction severely limits the dynamic capabilities of the robot [25].

Learning in a reasonable time-frame is essential to overcome the cost of getting experience on a real physical systems. Therefore, suitable approximations of state, policy, value function, and/or system dynamics are used as an alternative. Another option is to let the system learn in a simulated environment. However, for highly dynamic tasks, small modelling errors can lead to substantially different behaviours. Hence, the algorithms need to be robust against these models that do not capture all the details of the real system, i.e. uncertain models or undermodeled systems.

Defining a reward function that quickly guides the learning system to succeed on a specific task can also help to cope with the time-frame problem and thus, make RL suitable for real-world tasks. This issue is called "reward shaping" [29]. Specifying an effective reward function in robotics can be a non-trivial issue. An alternative to address this is inverse reinforcement learning (IRL) [41], where the reward function is learned rather than hand-defined.

In the context of this survey, IRL is seen as social learning; by leveraging the other's knowledge, a system can extract the concept of what is good or bad, i.e. the reward function. This is not the unique example of how self-learning can benefit from social learning. More hybrid approaches will be reviewed in Section IV.

B. Social Learning

Living creatures also learn from their neighbours. Others' knowledge can help in acquiring a new skill by understanding its utility, physics, rules, etc. Thus, it is natural that the learning interactions in animalia, i.e. social learning, serve as an inspiration in robotics. First approaches emulating this behaviour were using inverse optimal control (IOC) back to the 1960s. Its enchantment lies in finding a metric such that a known trajectory through a state space is optimal under the same metric [24]. Even though these control-based methods are still extensively used [6, 34], the popularisation of machine learning on the late 80s brought some alternatives such as learning by demonstration (LbD), inverse reinforcement learning (IRL) and transfer learning (TL), to ease the manual programming of robots and to automate the tedious parametrisation tweaking.

Social learning in robotics can occur between robots and/or between a human and a robot. This is similar to the human learning paradigm (see Section II), where learning can be conceived in two hierarchical directions: vertically (when a task is learnt from a teacher), and horizontally (when a task is learnt from other learners). In either case, this learning methodology allows learning two concepts from an external source: *what are the physics of a task?* and *what are the accepted actions during the performance of a task?*.

LbD is a subset of supervised learning which attempts to get the physics of a task. LbD is used to transfer knowledge from an expert to a machine, rather than analytically decomposing a problem and manually programming a desired behaviour [8]. A LbD process is threefold: (i) a set of demonstrations $\mathbf{d} \in \mathcal{D}$ of a specific task is acquired from a teacher demonstrator, (ii) a model or policy π is derived to generalise the demonstrations, and (iii) the learnt task is executed by repeatedly performing some action $\mathbf{a} \in \mathcal{A}$, which is dictated by the built policy after getting an observation $\mathbf{z} \in \mathcal{Z}$ of the system's state $\mathbf{s} \in \mathcal{S}$.

Given the increase of robots in common places, LbD brings a set of possibilities for HRI which allow non-robotics-experts to interact/teach robots [2]. Furthermore, teaching through demonstrations becomes more natural and an intuitive medium for communication from humans rather than programmatically defining what is the desired task. This approach has the practical feature of focusing the dataset on areas of the state-space that are actually encountered during the task execution.

IRL [41] was originally designed to deal with "reward shaping" issue [29]. This relates with the hardness of having to hand-define a reward function. Instead of modelling/defining such a function, IRL aims to extract it by observing an external agent performing the task meant to be learnt. This technique can extract the concept of desired and undesired actions when doing a task, i.e. the reward function. Similarly, it can be useful to also extract the parameters for control policies or the models from the demonstrations. Because of the interaction with external agents, in this survey we consider IRL as a type of social learning instead of a subfield of RL [2].

In order to derive the reward function from external demonstrations, IRL relies on the design of a suitable feature extractor. In other words, an algorithm which is able to capture the important aspects of the task in the problem space. Designing such function requires insights not much different from the ones that are required to design the actual reward function. Moreover, when deriving a reward function through IRL, there is no guarantee that the obtained reward function is the same as the one followed by the demonstrator. It will only be one that suits the provided demonstrations.

TL is a technique in machine learning which allows to reuse or adopt any already acquired knowledge. In this work, we distinguish two levels of TL:

- **Across tasks:** when generalization occurs not only within tasks, but also across tasks [49]. Generalisation in LbD is about repeating the same task in different conditions. Instead, TL also cares about transferring the knowledge to similar tasks, i.e. across tasks.
- **Across agents:** traditional learning approaches have focused on the problem of a single robot being taught by a single teacher [2]. However, horizontal learning between learners can benefit the group of robots to learn complex tasks by sharing their own experiences, i.e. models/policies.

As an example, robot A has learnt task A. Robot B is supposed to execute task B, which is similar to task A. In this context, TL is needed in both levels: across tasks and agents.

Because of that, we categorised TL to be as social learning. This approach not only faces the challenges in the other learning techniques, but also in multi-robot coordination: issues of action coordination, communication, noise in shared information, and different physical embodiments between agents. However, letting agents to interexchange information provides an interesting alternative to speeding up the learning process and generalisation across agents.

TL also brings the possibility of learning unobservable features from one's own. Exchanging information between agents with different perception sensors and computational capabilities lets the overall system be more dynamic. This allows ill-equipped robots to acquire knowledge that would not be in their capabilities otherwise. On the downside, the agents grow a dependency on the other's capabilities.

There is a great amount of works benefiting from acquiring knowledge from external observations, either with LbD, IRL or TL. However, those approaches come with some challenges and limitations such as the following:

- **Different agent's morphologies:** some HRI approaches to record the demonstrations might emphasize the anatomical differences between agents, either the teacher and the learner (in vertical transfer knowledge), or between learners (in horizontal transfer knowledge). This is known as the correspondence problem [36]. This issue deals with the identification of a mapping between the teacher and the learner which allows transferring of information from one to the other. An explanation of the required mappings, HRI acquisition approaches, and the generalisation challenge can be found in [2].
- **Lack of task generalisation:** LbD already provides an example of a good approach to undertake a specific task. This, however, might scale poorly to some new environment setup, where the context is dissimilar to the one of the demonstration. Billard et al. exemplify this situation with a dynamical system which is able to readjust a manipulation task with different starting/target locations from the ones shown during the demonstration [8]. However, this approach would not scale when placing a large obstacle in the robot's path.
- **Limited performance:** a model or policy learnt from external sources can only be as good as the one provided by the demonstrations. Consequently, the learner performance is heavily limited by the quality of the dataset. Argall et al. mention that poor learner performance can be due to: (a) dataset sparsity, or under-demonstrated areas of the state space, and (b) poor quality of the demonstrations, which can result from a teachers inability to perform the task optimally [2].

The major limitation of social learning is the need of resemblance between agents or resources to map the agent's capabilities. Related to that, a policy and reward function extracted by observing demonstrations have to be linked in the same way as they are demonstrated [1]. This creates an extra link further from the previously mentioned challenges.

Social learning can benefit from self-learning to ease its limited performance: a policy initially learnt by demonstration can be enhanced using RL. This is not the unique way of how social learning can benefit from self-learning; other hybrid approaches are reviewed in Section IV.

IV. APPRENTICESHIP LEARNING REVIEW

Apprenticeship learning brings together the best of two worlds: self-learning and social learning. This makes robotics closer to the animalia learning paradigm, where a skill can be acquired from an expert and then mastered with practice. Additionally, integrating both paradigms allows each of them to mitigate the others shortcomings at the same time of improving the overall learning quality [16]. Thus, the resulting learning framework becomes enhanced in terms of capabilities, adaptability and generalisation.

Bearing in mind the drawbacks of using some learning techniques (see Section III), this section starts by describing some of the existent methods in apprenticeship learning to deal with different agents' morphologies (see Section IV-A). It then overviews different approaches for avoiding engineer-crafted reward functions (see Section IV-B). Then, it provides an insight of how self-learning can be guided using social-learning techniques (see Section IV-C). Finally, Section IV-D overviews the enhancement on generalisation and performance when using apprenticeship learning frameworks.

A. Agent Differences Adjustment

Letting an agent practice a specific task might help shortening the teacher-learner perception and kinematic differences (see Section III-B). These discrepancies do not only arise because of the different agents' anatomical structure, but also due to the HRI setup used for the communication between agents. This fact is known as the correspondence problem [14, 36] and supposes a challenge to make the collection of state-action pairs recorded during the demonstrations usable by the learner. In fact, this turns to be crucial

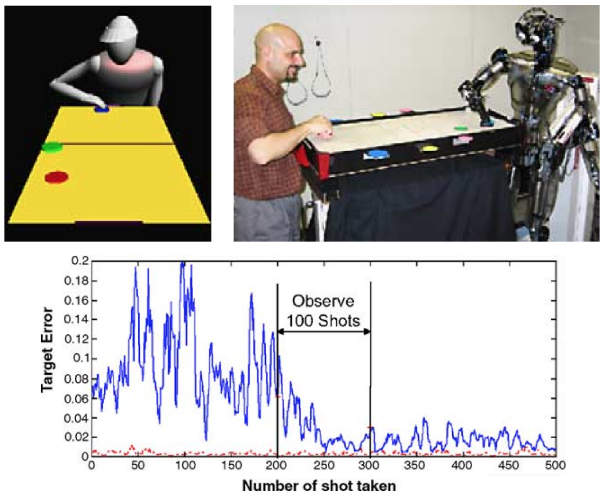


Fig. 3: Bentivegna et al. teaching a humanoid robot to play air-hockey. With the learnt model, the robot does 200 straight shots. When it processes the outcome of the trials (shots 201−300), it eventually improves its skills [5].

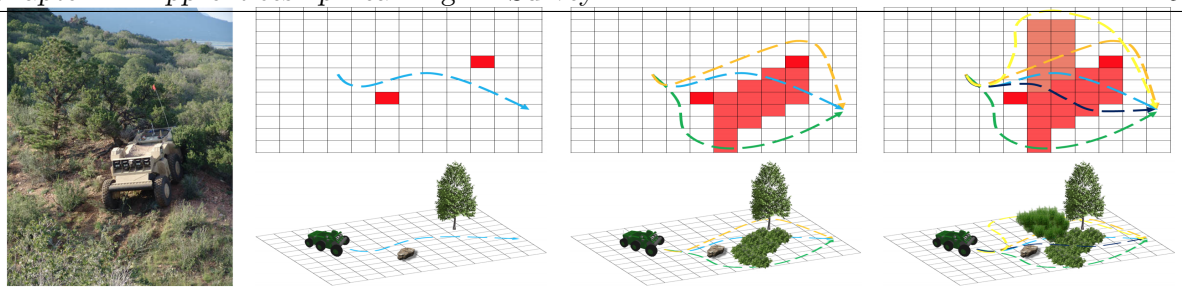


Fig. 4: Silver et al. highlighted the importance of the reward (cost) function on a robots behaviour. As the environment's harshness increases (from left to right), more complex parametrisations are required. In such environments, the tuning of the reward function can drastically affect the robot's behaviour [43].

for being successful in the learning process [2]. In an attempt to overcome this issue, many alternatives have been proposed in the LbD field to facilitate the mapping challenges. These solutions are out of the scope of this survey, nonetheless, an excellent overview of them is available at [2, 36].

An example of this challenge is reported in Nakanishi et al.. They aimed to learn biped locomotion from human demonstrations. However, they did not succeed because the kinematic differences between the teacher and the real humanoid were too noticeable. Thus, they finally got the demonstrations from an already existing successful robot locomotion [35]. Even though they found an alternative to avoid the correspondence problem, this approach might not always be doable. Instead, one could shorten the difference between embodiments with a simulated model of the system, avoiding the cost of self-learning in real robots. However, the differences between simulators and real robots can still suppose a challenge. Bentivegna et al. compensate the kinematic differences between a human demonstrator and the real robot when playing air-hockey (see Figure 3) in two stages. First, simulated system masters the learn task without the cost of running the full robot setup. Second, the resulting model is transferred to the real robot which is used as a basis for another self-learning stage [5]. This approach not only lets the robot to leverage from the human demonstrations but also to master its skills in a safe way.

Because the kinematic differences are strongly coupled with the acquired skills, it is not always obvious when the system is smart enough to generalise any task. Gräve et al. implemented RL and LbD as two alternative control flows for learning and executing grasping behaviours. Whenever there is enough knowledge to safely proceed on the required task, the RL module will drive the system to succeed on the task at the same time of improving the policy in terms of both quality and kinematic differences. Otherwise new demonstrations are required [16]. Similarly, Ross et al. dealt with the assumption that experts always give correct demonstrations, which makes the learning quality dependant from any mistake that the learner might commit, and any recording and/or communication problem between agents. To address this issue, they acknowledge a continuous input flow of demonstrations, allowing them to update the policy with the most suitable state-action pairs [40]. These two novel and alternative approaches are further discussed in Section IV-C.

B. Reward Function Acquisition

Deriving the reward function using IRL [41] becomes an alternative to deal with the "reward shaping" [29] issue introduced in Section III-A. This approach consists in learning from external observations which behaviours are satisfactory when performing a task, i.e. the reward function. Other options in between engineer-crafted and learnt reward functions are reward function adoptions (transferred from another agent) and processing external reward signals. Even though this approach involves multi-robot cooperation or an additional HRI system, successful examples undertaking this former approach can be found in [32], and the latter approach in [16, 17, 22]. There are several successful applications of IRL, some of them being completely decoupled from the reviewed apprenticeship learning literature but still a source of inspiration documented in [25].

In either case, the reward function or signal serves as guidance to shape up the policy during a self-learning process. Even though the reward function might correctly guide the learning process, there are no guarantees that the learnt reward function is exactly the same underlying the expert's demonstrations [1]. Silver et al. discuss the importance of the reward function since it can drastically compromise the system's behaviour (see Figure 4). Thus, special attention must be taken when learning the reward functions from external agents, even more acknowledging that this method is also sensible to the aforementioned correspondence problem [36].

In between control-theory and the apprenticeship learning paradigm, Hwang et al. learnt the whole body dynamics of the Saika-3 humanoid robot [27] to acquire manipulation skills. This was achieved via self-learning cooperative motion using a simple genetic algorithm (SGA). In this context, the main body and the arms are considered different agents, which try to minimise the overall used energy but while cooperatively succeeding on the task [20]. A step towards apprenticeship learning was done by Abbeel and Ng, which formulated an MDP\R (Markov decision process (MDP) without reward function) to iteratively adjust the policy underlying the system's behaviour accordingly to the reward

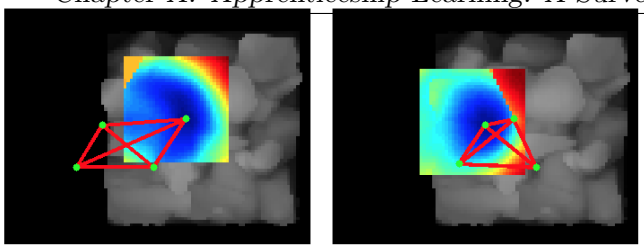


Fig. 5: Ratliff et al. learnt reward functions for making the Boston Dynamics LittleDog quadruped robot walk. The function includes the learnt kinematic feasibility and the learnt terrain costs over a depth-map. Low costs are indicated with bluish shades and high costs with reddish shades (high) [39].

function being guessed from the expert demonstrations. Interesting results were reported in a car driving simulation, where the algorithm was able to learn different driving styles without any previous definition of the reward function [1].

Related to reward function derivation, some works have focused on learning cost functions which could then be used in a learning framework. For instance, Ratliff et al. learnt from human demonstrations how to score each action within a multi-class classification framework. They not only tested their approach to grasping behaviours, but also in quadruped locomotion. For the latter scenario, they learnt the kinematic constraints and the costs related to the features of the terrain, using the overall cost function to plan the next step [39] (see Figure 5). Similarly, Silver et al. learnt terrain cost functions from human demonstrations to reduce development efforts and to increase the overall performance of the system. Their approach was tested in a Crusher autonomous navigation platform [46] to make it navigate at the lowest cost through complex unstructured terrains [43].

C. Learning Exploration Guidance

Guiding the exploration of a self-learning approach might ease its difficulties of dealing with high-dimensional or/and real systems (see Section III-A), and reduce the learning time [16]. The system can be guided with hand-crafted policies or models derived from demonstrations. The former requires expertise and knowledge about the system to model it, becoming a non-trivial task. This issue is similar to the one of engineer-crafted reward functions (see Section III-A). The latter relies on the quality of the demonstrations, making the overall framework vulnerable from any error in the acquired model [40] (see Section IV-A). In between these two approaches, simulating human behaviours using sub-optimal controllers seems to be a suitable alternative [17, 40, 50]. This avoids inconsistencies associated with human users and lets the initial policy be just close enough to the desired task but still imperfect to let the self-learning algorithm improve the model. Similarly, some authors acquired demonstrations from already working systems [28, 35].

In either case, the obtained model can be used at different learning stages: (1) policy initialisation and/or (2) policy's evolution evaluation. These approaches, which might be used together, prevent the self-learning algorithm undertaking a greedy exploration of the entire state-action space.

1) Policy initialisation: knowing an initial model might help to overcome some of the aforementioned limitations. This initial model already contains what actions to perform in the states encountered in policy and allows the learner to improve it by locally optimizing it. However, this implies that only local optima close to the initial policy can be found, making the learning procedure dependant of the provided policy. In other words, this technique becomes only useful when the context for the reproduction is sufficiently similar to that of the initial policy [8]. An alternative to this issue is proposed by Ross et al.; they batch the demonstrations over iterations, providing a no-regret online learning which reduces the learning scope to the most suitable demonstrations while still having strong performance guarantees [40]. However, this approach requires a continuous input of data, i.e. initial policy candidate.

A classic but practical approach to policy initialisation is provided by Silver et al.. They learnt terrain cost functions from human demonstrations to reduce development efforts and to increase the overall performance of the system shown in Figure 4. For that, a maximum margin planning (MMP) was used to learn a cost function from the demonstrations [43]. Yet another real-world example but with a humanoid robot consists on providing some demonstrations to the robot, so it can start a self-learning procedure already knowing how to repeat a task with various starting and goal points [18]. In the same lines, Bentivegna et al. showed a humanoid robot how to play air-hockey so then it could then master its skills (see Figure 3). Results not only in this work but also in Taylor et al. show the importance of providing some initial examples about the task [5, 50].

Improving an initial policy leads to better results when it has been learnt by demonstration rather than hand-defined [50]. However, learning a policy might not always be possible, for what there might be hybrid approaches such as getting information from already functional robots. In this regard, Latzke et al. transferred the knowledge from a similar robot to speed up the learning process, thus reducing the required own trial and error to master a fundamental soccer skill [28]. Likewise, Nakanishi et al. leveraged the experience of another robot with successful robot locomotion in order to learn biped locomotion in a new system. This initial knowledge served as a basis to improve the robot's locomotion capabilities with control-based algorithms [35].

2) Policy clarification: there are situations in which a self-learning approach might require more data. In these scenarios, an algorithm can benefit from actively querying the teacher for additional demonstrations when needed. Such a method guides by reinforcement the policy that is being learnt. At the same time, this approach opens a new window for HRI, where an agent can be reinforced in more natural ways. For example, Grollman and Jenkins avoids autonomous exploration during the learning process by highly interacting with the robot. They use mixed initiative control (MIC) to acquire initial demonstrations, get the feedback

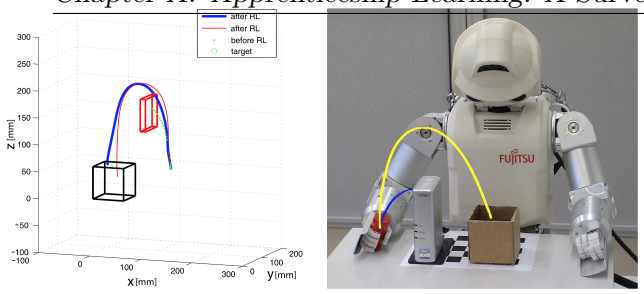


Fig. 6: Guenter et al. explored self-learning techniques to let HOAP3 adapt to a new situation (sudden appearance of an obstacle in the middle of the learnt trajectory) where the demonstrations (green line) do not help to fulfill the task [18].

from a human and incorporate additional examples of the correct control policy. They called this approach dogged learning (DL) [17]. Instead of a continuous interaction, Gräve et al. checks the uncertainty of the derived model to determine if there is sufficient information to safely execute the task. In case of the risk being too high, new examples are asked to a human expert to fulfill the current task [16]. Similarly, Jansen and Belpaeme focused on goal-oriented learning, inferring the policy of a demonstrator by observing its movements, executing the guessed hypothesis and waiting for some feedback about the learning performance. [22].

Alternatively to human guidance, robots can cooperate to the same end by interchanging reward signals to guide the policy learning. An interesting work is the one of Tan, in which they compared the performance of cooperative agents against individual agents. They found out that multi-agent learning initially captures knowledge slower, but at the end, they outperform individual learners [48]. A practical work is the one of Mataric, which used cooperative learning to enhance the performance of the overall population of agents. For that, agents were interchanging local reinforcements and sending perception data to reveal hidden states. Hence, reducing the need for self-exploration when learning the map between perception and actions [32]. Billard and Dautenhahn also showed that behavioural social mechanisms speed up the grounding of exteroceptions [9].

D. Generalisation and Performance Enhancement

Iterating over an initial model lets an agent improve the performance and generalisation of such a skill (see Section III-B). This is remarkably important because the performance of a system which uniquely learns from demonstrations is limited by the capabilities of the teacher [2]. Instead, this approach allows surpassing the abilities of the demonstrator [50]. Apart from that, the generalisation capabilities of an agent are limited to similar scenarios to the demonstrations. For instance, Gräve et al. state that a policy purely learnt through social learning is still too constraint for grasping behaviours [16].

In the context of apprenticeship learning, the generalisation issue is usually tackled using reinforcement learning, e.g. [16, 18, 22]. For instance, Guenter et al. first encodes the model underlying some human demonstrations. In order to make the system adaptable to different scenarios, a RL module is used to adapt the policy to new environments. They tested their approach in a humanoid robot which had to grasp objects in unseen positions and put them into a box, even with the presence of unseen obstacles in the middle of the demonstrated trajectories [18] (see Figure 6). Similarly, Gräve et al. leverage self-learning to generalise the system's knowledge to new grasping scenarios [16]. A slightly different approach is the one adopted in Jansen and Belpaeme. They focused on goal-oriented learning to infer the policy underlying the demonstrator's behaviour. Because generalising intentions can be challenging, after observing the teacher and executing the guessed hypothesis, the system received some external feedback at the end of each trial [22].

The hybridisation of a social learning approach with a self-learning technique not only improves the generalisation capabilities, but can also enhance the overall performance of an already known skill. In this regard, Kober et al. first provided kinaesthetic guiding to a simulated anthropomorphic SARCOS arm to complete the Ball-in-a-Cup game to then let the system practice the new skill. In this work, the self-learning was not only essential to succeed on the game but also to perfect the movement of the ball [26]. Again, the modelisation of the reward function plays an important role on what its being mastered. For example, Yoshikai et al. tweak the reward function so the whole body tendon-driven humanoid Kenta [21] can learn that is supposed to mimic the posture of a demonstrated human hand and then master such an imitation [51]. Also, Bentivegna et al. resort to a self-learning approach to enhance the performance of the demonstrator when playing air-hockey (see Figure 3). Results show that letting the robot practice by its own is equally important as providing some initial knowledge about the task [5]. The same moral is stated in Taylor et al..

In contrast to individual self-learning approaches, enhancement of an initial policy can also be achieved with cooperative systems. Multi-agent learning can speed up the learning process. A population of agents can copy, mimic and share information with each other so that they learn control policies by making experiments themselves and by watching others. Using genetic algorithms, Hwang et al. considered the different extremities of a simulated humanoid as different agents to minimise the overall spent energy when cooperatively succeeding on a manipulation task [20]. Already in the apprenticeship learning field, Mataric evolved the behaviour of an overall population of agents by means of a common reward function [32]. As stated in Tan, the price of this cooperation is worth because, in long term, cooperative agents outperform individual learners. They exemplified this fact by sharing sensations, sharing episodes, and learned policies between hunter agents which sought to capture randomly-moving preys [48]. This performance enhancement is the result of combining some of the previously seen strengths of the apprenticeship learning approach. Rewards are obtained from one own and other's experiences (see Section IV-B) and by letting this signals and other's policies guide (initialise

and clarify) one's own policy (see Section IV-C).

Enhancing the generalisation capabilities and/or system's performance requires practice from a self-learning module. As discussed in Section III-A, this exploration usually becomes prohibitively unsafe on real-world setups [2]. Alternatively, many works in this fields conduct self-learning approaches in simulated environments in order to test the performance and generalisation enhancement of an initial policy [1, 5, 22, 26, 39, 40, 50]. According to Taylor and Stone, ideally, the agent could learn a specific behaviour by exploiting its own model in simulated environments to then transfer the knowledge to the real robot and directly interact with real-world scenarios [49]. This step of learning on the simulated system is often called "mental rehearsal. Even though this promising alternative, such an approach requires extracting a precise model from both the agent and the environment, the challenge of which have been discussed in Section IV-C. Even though Kober et al. provide a full discussion of the challenges of "mental rehearsal" [25], it is important to bear in mind that small model errors due to under-modelling can make the simulated learning diverge from the required one for the real-world system. Under these circumstances, a direct transfer of knowledge might only succeed in limited applications. Acknowledging this fact, letting the real robot practice the new acquired skill is still essential to master a task [18, 28]. In any case, simulation becomes an interesting alternative to go through in order to narrow the gap between biological systems and real robots.

E. Summary of Works

Comparing apprenticeship learning frameworks is challenging because (a) not all works aim for the same improvement with respect to the shortcomings defined in Section III, (b) the testbed applications differ with the requirements, and (c) different robots are used, which have different sensoriception and motion capabilities. Aiming to provide the reader with a useful and quick guide to identify works of potential interest, Table I summarises some of the aforementioned works, for which six features have been selected:

- Learning media: interaction method for social learning.
- Learning support: mathematical encapsulation of the model underlying the application of interest.
- Learning refinement: acquisition of the reward signal for self-learning.
- Learning techniques: highlight of the used techniques, either learning-based (LbD, TL, IRL, RL) or others (control-based approaches, genetic algorithms, etc).
- Application: objective task/skill to be learnt/acquired.
- Learner: learning target, emphasising either in simulation or real-world.

Acknowledging the impossibility of embedding all information in this table, the section where each work is mentioned is indicated underneath the references. The reader will find in there the interest, contribution and peculiarity of each work.

V. DISCUSSION

In this review paper, we have seen that learning in robotics has been strongly inspired by the human learning. Although this paradigm has been pursued for many decades, the current state-of-the-art on robotics learning is the legacy of advanced control-based approaches developed in the early 80s. Mainly, two learning strategies prevail in the literature: self-learning and social learning. Due to the imminent (if not yet) matureness of these techniques, an emerging field of research hybridises social and self-learning to mitigate the others weaknesses, at the same time of improving the overall learning quality. In an attempt to formalise this concept, usually named apprenticeship learning, this paper has first reviewed the shortcomings of traditional learning approaches to then provide an insight to some of the most outstanding works in the apprenticeship learning field.

Current works in this field have done a step forward in AI by obtaining more autonomous systems; fewer modelisation requirements, extended generalisation capabilities and increased learning rate, are only some of the advantages of this novel learning technique. Though apprenticeship learning has proven promising advances, there is a long way to go. As emphasised in this review, many works uniquely meet some of these strengths individually. On top of that, they are presented in particular setups, i.e environments, tasks and robots. This fact makes difficult to disseminate the advances in this field among different robotic domains.

Working towards a generic framework should be the long-term aim of the learning in the robotics community. To achieve this robustness and generalisation it is essential to bear in mind the complete human learning paradigm: getting inspiration from external sources, processing the acquired knowledge and practising to master a task. Equivalently in robotics, this implies first learning from humans or other robots, being cooperatively a natural consideration; second, generalising the concepts not only along the same skill but also across others. Finally, consummate expertise by iteratively testing and evaluating one's own performance. Endowing a robot with all these capabilities is the way to achieve more autonomous systems. However, the majority of works in the field only happen in simulated environments. The comfortableness and safeness of developing self-learning in this context should only be the prior step before transferring the knowledge to real robotic platforms.

Despite the promising route of this emerging field, apprenticeship learning lacks a standard set of tasks, scenarios and evaluation metrics. This complicates comparisons across algorithms and domains. Instead, formalising an evaluation criterion would facilitate the contrast between approaches and implementations, at the same time that it would help driving research and the development of apprenticeship learning towards general-purpose frameworks. Nonetheless, this seems to be still a pending subject on the fundamentals of apprenticeship learning. Thus, interdisciplinary efforts should be focused on addressing this absence, which would benefit the learning community in robotics as a whole.

Reference	Learning media	Learning support	Learning refinement	Learning techniques
Abbeel and Ng [1] (S.IV-B)	Teleoperation in simulation	MDP\R	Via learnt RF	**LbD** TL **IRL** **RL** Oth.
	Application: mimic driving styles.		**Learner:** car driving scene **(simulation)**.	
Kober et al. [26] (S.IV-D)	VICON^{TM} motion-capture setup	Augmented DMP	Via hand-defined RF	**LbD** TL IRL **RL** Oth.
	Application: ball-in-a-cup game.		**Learner:** anthropomorphic SARCOS arm **(simulation)**.	
Gräve et al. [16] (S.IV-A, S.IV-B, S.IV-D)	Motion capture rig and a data glove	DMP and GP	Via external HE	**LbD** TL **IRL** **RL** Oth.
	Application: grasp object in random position.		**Learner:** Dynamaid humanoid robot **(simulation)**.	
Latzke et al. [28] (S.IV-C, S.IV-D)	Similar robot and teleoperation in simulation	MDP	Via hand-defined RF	LbD **TL** IRL **RL** Oth.
	Application: fundamental soccer skills.		**Learner:** RoboSapien humanoid robot **(real)**.	
Ratliff et al. [39] (S.IV-B)	Teleoperation	-	Via learnt RF	**LbD** TL **IRL** RL Oth.
	Application: cost maps and grasping metrics.		**Learner:** Boston Dynamics LittleDog quadruped robot, Barrett Technologies three-fingered hand **(simulation)**.	
Silver et al. [43] (S.IV-B, S.IV-C)	Teleoperation and exteroceptive sensors	-	Via learnt RF	**LbD** TL **IRL** RL Oth.
	Application: autonomous navigation.		**Learner:** Crusher autonomous navigation platform **(real)**.	
Hwang et al. [20] (S.IV-B)	-	-	Genetic algorithm	LbD TL IRL RL **Oth.**
	Application: energy optimisation on pushing task.		**Learner:** Saika-3 humanoid robot **(simulation)**.	
Tan [48] (S.IV-C, S.IV-D)	-	MDP	Via hand-defined RF	LbD **TL** IRL **RL** Oth.
	Application: cooperative prey hunting.		**Learner:** group of hunters **(simulation)**.	
Mataric [32] (S.IV-C, S.IV-D)	-	MDP	Via shared hand-defined RF	LbD **TL** IRL **RL** Oth.
	Application: cooperative box-pushing and sensor-actuator mapping.		**Learner:** two Genghis-II six-legged robots, four IS Robotics R2e robots **(real)**.	
Ross et al. [40] (S.IV-A, S.IV-C, S.IV-D)	Teleoperation or near-optimal planner	N/S	N/S	**LbD** TL IRL RL **Oth.**
	Application: steer a car and succeed in a game.		**Learner:** 3D racing game, Super Mario Bros **(simulation)**.	
Nakanishi et al. [35] (S.IV-A, S.IV-C)	Human walking data	DMP	N/S	**LbD** TL IRL RL **Oth.**
	Application: learn and adapt biped locomotion.		**Learner:** 5-link biped robot **(simulation and real)**.	
Grollman and Jenkins [17] (S.IV-C)	Hand-coded controllers	MIC	Via external HE	**LbD** TL IRL RL **Oth.**
	Application: mimicry and ball seeking.		**Learner:** Sony Aibo **(real)**.	
Jansen and Belpaeme [22] (S.IV-C, S.IV-D)	Human demonstrator and simulation	Set of rules	Via external HE	**LbD** TL IRL RL **Oth.**
	Application: logical policy inference.		**Learner:** two-dimensional 5-by-5 blocks world **(simulation)**.	
Guenter et al. [18] (S.IV-C, S.IV-D)	Kinaesthetic demonstrations	GMM and GMR	Via hand-defined RF	**LbD** **TL** IRL **RL** Oth.
	Application: adaptability on putting an object into a box and grasping a chess queen on a table.		**Learner:** HOAP3 humanoid robot **(real)**.	
Yoshikai et al. [51] (S.IV-D)	-	Sensor-action attention pair	Via hand-defined RF	LbD TL IRL **RL** Oth.
	Application: human hand posture imitation.		**Learner:** Kenta tendon-driven humanoid robot **(real)**.	
Bentivegna et al. [5] (S.IV-A, S.IV-C, S.IV-D)	Teleoperation in simulation	Subgoal-action pair	Via hand-defined RF	**LbD** **TL** IRL RL **Oth.**
	Application: air-hockey and marble maze.		**Learner:** humanoid robot **(real)**, marble maze **(simulation)**.	
Billard and Dautenhahn [9] (S.IV-D)	N/S	Sensor-action pair	-	**LbD** **TL** IRL RL Oth.
	Application: cooperative localisation.		**Learner:** agents **(simulation)**.	
Taylor et al. [50] (S.IV-C, S.IV-D)	Teleoperation or sub-optimal controller	MDP	Via hand-defined RF	**LbD** **TL** IRL **RL** Oth.
	Application: cooperative ball keeping.		**Learner:** agents in Keepaway **(simulated)**.	

TABLE I: Summary of apprenticeship learning frameworks. Oth.: others, usually referring to control-based approaches. RF: reward function. HE: human evaluation. N/A: not applicable. N/S: not specified.

REFERENCES

[1] Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM.

[2] Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robot. Auton. Syst.*, 57(5):469–483.

[3] Åström, K. J. and Wittenmark, B. (2013). *Adaptive control*. Courier Corporation.

[4] Bellman, R. (2013). *Dynamic programming*. Courier Corporation.

[5] Bentivegna, D. C., Atkeson, C. G., and Cheng, G. (2004). Learning tasks from observation and practice. *Robotics and Autonomous Systems*, 47(2-3):163–169.

[6] Berret, B., Chiovetto, E., Nori, F., and Pozzo, T. (2011). Evidence for composite cost functions in arm movement planning: an inverse optimal control approach. *PLoS computational biology*, 7(10):e1002183.

[7] Biggs, G. and Macdonald, B. (2003). A Survey of Robot Programming Systems. *Proceedings of the Australasian conference on robotics and automation*, pages 1–3.

[8] Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). Robot Programming by Demonstration. *Robotics*, pages 1371–1394.

[9] Billard, A. and Dautenhahn, K. (1999). Experiments in learning by imitation-grounding and use of communication in robotic agents. *Adaptive behavior*, 7(3-4):415–438.

[10] Bohren, J., Rusu, R. B., Jones, E. G., Marder-Eppstein, E., Pantofaru, C., Wise, M., Mösenlechner, L., Meeussen, W., and Holzer, S. (2011). Towards autonomous robotic butlers: Lessons learned with the PR2. In *Robotics and automation (ICRA), 2011 IEEE international conference on*, pages 5568–5575. IEEE.

[11] Bristow, D., Tharayil, M., and Alleyne, A. G. (2006). A survey of iterative learning control. *Control Systems, Institute of Electrical and Electronics Engineers*, 26(3):96–114.

[12] Burghes, D. N. and Graham, A. (1982). Introduction to control theory, including optimal control. *SIAM Review*, 24(1):87–89.

[13] Driscoll, M. P. (2000). Psychology of learning. *Boston, Allyn and Bacon*.

[14] Englert, P., Paraschos, A., Peters, J., and Deisenroth, M. P. (2013). Probabilistic Model-based Imitation Learning. *Adaptive Behavior*, 21(5):388—-403.

[15] Goodrich, M. A. and Schultz, A. C. (2007). Human-robot interaction: a survey. *Foundations and trends in human-computer interaction*, 1(3):203–275.

[16] Gräve, K., Stückler, J., and Behnke, S. (2010). Learning motion skills from expert demonstrations and own experience using gaussian process regression. In *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pages 1–8. VDE.

[17] Grollman, D. H. and Jenkins, O. C. (2007). Dogged learning for robots. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2483–2488. IEEE.

[18] Guenter, F., Hersch, M., Calinon, S., and Billard, A. (2007). Reinforcement learning for imitating constrined reaching. *RSJ Advanced Robotics, Special Issue on Imitative Robots*, 21(13):1521–1544.

[19] Heyes, C. M. (1994). Social learning in animals: categories and mechanisms. *Biological Reviews*, 69(2):207–231.

[20] Hwang, Y. K., Choi, K. J., and Hong, D. S. (2006). Self-learning control of cooperative motion for a humanoid robot. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 475–480. IEEE.

[21] Inaba, M., Mizuuchi, I., Tajima, R., Yoshikai, T., Sato, D., Nagashima, K., and Inoue, H. (2003). Building spined muscle-tendon humanoid. In *Robotics Research*, pages 113–127. Springer.

[22] Jansen, B. and Belpaeme, T. (2006). A computational model of intention reading in imitation. *Robotics and Autonomous Systems*, 54(5):394–402.

[23] Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.

[24] Kalman, R. E. (1964). When is a linear control system optimal? *Journal of Basic Engineering*, 86(1):51–60.

[25] Kober, J., Bagnell, J. A., and Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274.

[26] Kober, J., Mohler, B., and Peters, J. (2008). Learning perceptual coupling for motor primitives. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 834–839. IEEE.

[27] Konno, A., Kato, N., Shirata, S., Furuta, T., and Uchiyama, M. (2000). Development of a light-weight biped humanoid robot. In *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 3, pages 1565–1570. IEEE.

[28] Latzke, T., Behnke, S., and Bennewitz, M. (2006). Imitative reinforcement learning for soccer playing robots. In *Robot Soccer World Cup*, pages 47–58. Springer.

[29] Laud, A. D. (2004). Theory and application of reward shaping in reinforcement learning. Technical report.

[30] Looß, M. (2001). Types of learning? *Die Deutsche Schule*, 93(2):186–198.

[31] Marshall, J., Morton, J., Bever, T., Brown, J., Estes, W., Grossmann, K., Huber, W., Petersen, M., Squire, L., and Weinstein, S. (1984). Biology of learning in humans. In *The Biology of Learning*, pages 687–705. Springer.

[32] Mataric, M. J. (1998). Using communication to reduce locality in distributed multiagent learning. *Journal of experimental & theoretical artificial intelligence*, 10(3):357–369.

[33] Meltzoff, A. N., Kuhl, P. K., Movellan, J., and Sejnowski, T. J. (2009). Foundations for a new science of learning. *science*, 325(5938):284–288.

[34] Mombaur, K., Truong, A., and Laumond, J.-P. (2010). From human to humanoid locomotion: an inverse optimal control approach. *Autonomous robots*, 28(3):369–383.

[35] Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., and Kawato, M. (2004). Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems*, 47(2):79 – 91.

[36] Nehaniv, C. and Dautenhahn, K. (2002). The Correspondence Problem. *Imitation in Animals and Artifacts*, pages 1–40.

[37] Nguyen-Tuong, D. and Peters, J. (2011). Model learning for robot control: a survey. *Cognitive processing*, 12(4):319–340.

[38] Nicolescu, M. N. and Mataric, M. J. (2003). Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 241–248. ACM.

[39] Ratliff, N., Bagnell, J. A., and Srinivasa, S. S. (2007). Imitation learning for locomotion and manipulation. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 392–397. IEEE.

[40] Ross, S., Gordon, G., and Bagnell, D. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635.

[41] Russell, S. (1998). Learning agents for uncertain environments. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 101–103. ACM.

[42] Schaal, S. and Atkeson, C. G. (2010). Learning control in robotics. *IEEE Robotics and Automation Magazine*, 17(2):20–29.

[43] Silver, D., Bagnell, J. A., and Stentz, A. (2010). Learning from demonstration for autonomous navigation in complex unstructured terrain. *The International Journal of Robotics Research*, 29(12):1565–1592.

[44] Singh, S., Lewis, R. L., and Barto, A. G. (2009). Where do rewards come from? In *Proceedings of the annual conference of the cognitive science society*, pages 2601–2606.

[45] Stengel, R. F. (1986). *Stochastic optimal control*. New York: John Wiley and Sons.

[46] Stentz, A., Bares, J., Pilarski, T., and Stager, D. The crusher system for autonomous navigation. *AUVSIs Unmanned Systems North America, 3*.

[47] Sutton, R. S., Barto, A. G., and Williams, R. J. (1992). Reinforcement learning is direct adaptive optimal control. *IEEE Control Systems*, 12(2):19–22.

[48] Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pages 330–337.

[49] Taylor, M. E. and Stone, P. (2009). Transfer Learning for Reinforcement Learning Domains : A Survey. *Journal of Machine Learning Research*, 10:1633–1685.

[50] Taylor, M. E., Suay, H. B., and Chernova, S. (2011). Integrating reinforcement learning with human demonstrations of varying ability. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 617–624. International Foundation for Autonomous Agents and Multiagent Systems.

[51] Yoshikai, T., Otake, N., Mizuuchi, I., Inaba, M., and Inoue, H. (2004). Development of an imitation behavior in humanoid kenta with reinforcement learning algorithm based on the attention during imitation. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 2, pages 1192–1197. IEEE.

Appendix B

iCub Kinematics

Knowing the robot's kinematic structure is crucial to understand the robot's control capabilities. This appendix firstly details the most relevant physical parts of the iCub humanoid robot. After the hardware description, this appendix gives an overview of the iCub's software and control.

B.1 Physical Platform

iCub's kinematic tree is rooted at the middle of the torso (see Figure B.1), where its reference frame is oriented as follows: x-axis points backwards the robot, the y-axis points laterally to the right, and the z-axis is parallel to gravity but pointing upwards.

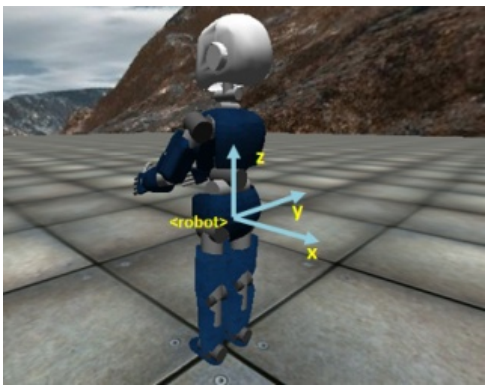
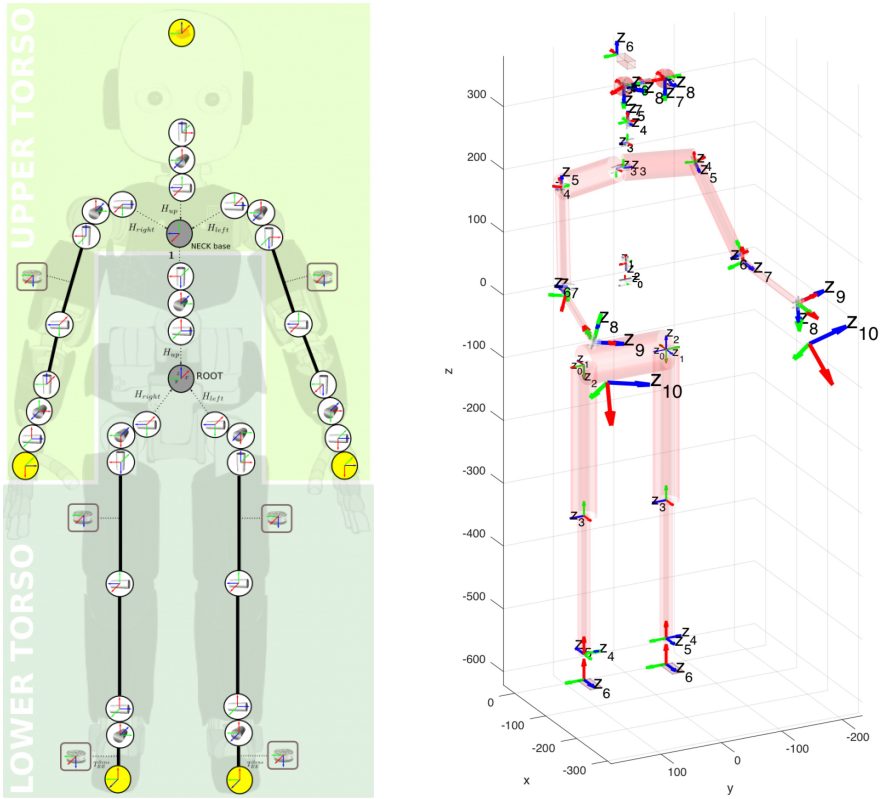


Figure B.1: iCub's composition and reference frames. (a) iCub's global reference frame, (b) iCub's kinematic tree, (c) reference frames of iCub's joints.

Figure B.2a and Figure B.2b show the kinematic chain of all the components of iCub attached to the previously mentioned iCub's root (see Figure B.1). Not all hardware elements are represented



(a) (b)

Figure B.2: iCub's composition and reference frames. (a) iCub's global reference frame, (b) iCub's kinematic tree, (c) reference frames of iCub's joints.

with respect to this root, but also all control commands, as well as all retrieved data from the iCub sensors. Describing in detail the whole iCub's kinematics is out of the scope of this project. Instead, an understanding of the kinematic composition of the iCub's arms described below is fundamental for the research conducted in this thesis.

B.1.1 Arms Constitution

The kinematic chain of the iCub's right and left end-effector are depicted in Figure B.3a and Figure B.3b, respectively. As it can be seen, the location of both end-effectors is dependant on the three torso's DoFs and the corresponding seven arm's DoFs. Even though the composition of both end-effectors looks symmetric, they are exactly not.

The Denavit-Hartenberg convention proposed in [Hartenberg and Denavit, 1964] is considered to describe the aforementioned end-effector kinematic chains (see Table B.1 and Table B.2 for the right and left end-effector, respectively). This standard uses four parameters to describe a

link i, i.e. the relative location of the j^{th} joint frame with respect to the $(j-1)^{th}$ joint frame:

- **Link length a_i:** distance along x_j from O_j to the intersection of the x_j and z_{j-1} axes.
- **Link offset d_i:** distance along z_{j-1} from O_{j-1} to the intersection of the j_i and z_{j-1} axes. d_i is variable if joint j is prismatic.
- **Link twist α_i:** the angle between z_{j-1} and z_j measured about x_j.
- **Joint angle θ_i:** the angle between x_{j-1} and x_i measured about z_{j-1}. θ_i is variable if joint j is revolute.

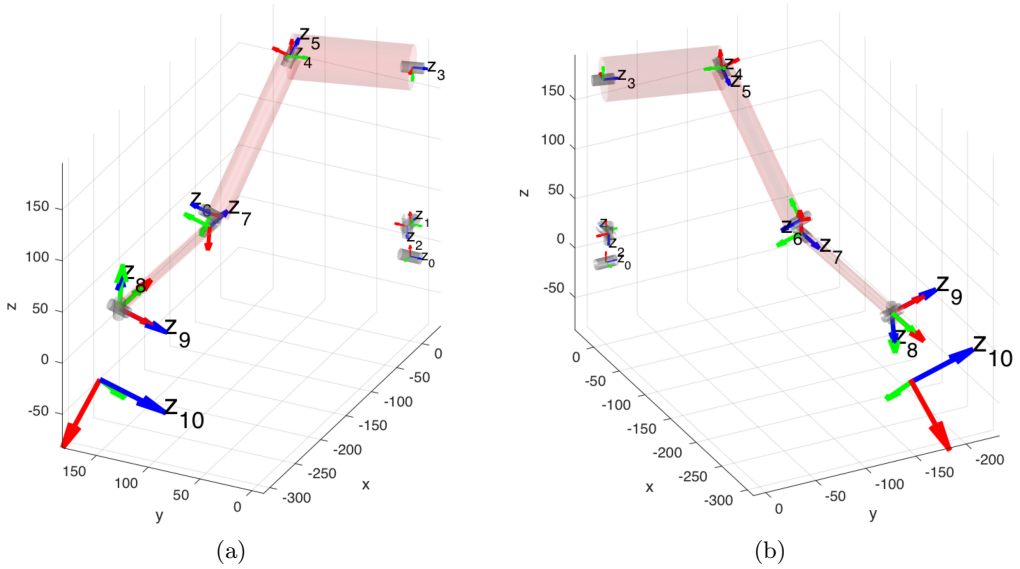


Figure B.3: iCub's end-effector kinematic chain. (a) Right arm and (b) left arm.

B.2 Software Architecture

iCub's physical platform is fully controllable through its YARP-based software architecture. YARP is a free and open set of libraries, protocols, and tools which allows building a robot control system as a collection of programs communicating in a peer-to-peer way [Metta et al., 2006]. The wholly decoupled modules and devices communicate with each other through TCP, UDP, XML, multicast, etc. This endows any YARP-based architecture, such as iCub's one, with a high-level of adaptability, making it perfect for long-term development projects. YARP is written in C++ and is supported by Windows, Linux and macOS.

YARP is similar to robot operating system (ROS) [Quigley et al., 2009], despite the former states not being an operating system [Metta et al., 2006]. In ROS, each program is a *node*

Link i	a_i **[mm]**	d_i **[mm]**	α_i **[deg]**	θ_i **[deg]**
1	32	0	90	(-22 → 84)
2	0	-5.5	90	-90 + (-39 → 39)
3	-23.3647	-143.3	90	-105 + (-59 → 59)
4	0	-107.74	90	-90 + (5 → -95)
5	0	0	-90	-90 + (0 → 160.8)
6	-15	-152.28	-90	-105 + (-37 → 100)
7	15	0	90	(5.5 → 106)
8	0	-141.3	90	-90 + (-50 → 50)
9	0	0	90	90 + (10 → -65)
10	62.5	25.98	0	180 + (-25 → 25)

Table B.1: Denavit-Hartenberg paramters for iCub's right end-effector. The first three links are from the torso. The last seven links are from the right arm.

Link i	a_i **[mm]**	d_i **[mm]**	α_i **[deg]**	θ_i **[deg]**
1	32	0	90	(-22 → 84)
2	0	-5.5	90	-90 + (-39 → 39)
3	23.3647	-143.3	-90	105 + (-59 → 59)
4	0	107.74	-90	90 + (5 → -95)
5	0	0	90	-90 + (0 → 160.8)
6	15	152.28	-90	75 + (-37 → 100)
7	-15	0	90	(5.5 → 106)
8	0	141.3	90	-90 + (-50 → 50)
9	0	0	90	90 + (10 → -65)
10	62.5	-25.98	0	(-25 → 25)

Table B.2: Denavit-Hartenberg paramters for iCub's left end-effector. The first three links are from the torso. The last seven links are from the left arm.

and communication is established through *topics*. Similarly, the programs in YARP are called *modules* and communicate through *ports*. Even though iCub's architecture and YARP are well documented, there is not any formal comparison between the aforementioned systems.

B.2.1 Arms Control

The Denavit-Hartenberg convention [Hartenberg and Denavit, 1964] is systematic in the choice of the location and orientation of all reference frames. This allows to represent the link i with

the same strategy, i.e. a product of four basic transformations:

$$A_i = R_{z,\theta_i} T_{z,d_i} T_{x,a_i} R_{x,\alpha_i}, \tag{B.1}$$

where $A_i = {}^{j-1}H_j$ with $j = i$, i.e. the homogeneous transformation defining the state of the joint j with respect to the joint $j-1$, and $R_{m,k}$ and $T_{m,k}$ respectively stand for an homogeneous rotation and translation matrix in the m-axis with parameter k.

Given this definition, the roto-translation from the 0^{th} reference frame (i.e. iCub's root) to the N^{th} reference frame (e.g. right end-effector, involving N DoFs) is computed as:

$${}^{0}H_N = \prod_{i=1}^{N} A_i. \tag{B.2}$$

Equation (B.2) lets us computing the FK for any of the iCub's arms. In other words, given a feasible joint configuration set, ${}^{0}H_N$ indicates the location and orientation of the end-effector's reference frame with respect to iCub's root frame. These transformations are already available in the iCub's architecture, specifically in the `iKin` library. This library does not only allow to perform FK, but also the IK, i.e. computing the required joint configuration set for reaching a certain position with the end-effector. For that, the library has an application called `cartesian_solver` [Pattacini et al., 2010], which minimises:

$$\mathbf{q} = \arg\min_{\mathbf{q}\in\mathbb{R}^n} \left(\left\| \boldsymbol{\alpha}_d - K_\alpha(\mathbf{q}) \right\|^2 + w \cdot (\mathbf{q}_r - \mathbf{q})^\top W_r (\mathbf{q}_r - \mathbf{q}) \right) \quad s.t. \begin{cases} \left\| \mathbf{x}_d - K_x(\mathbf{q}) \right\|^2 = 0 \\ \mathbf{q}_L < \mathbf{q} < \mathbf{q}_U \end{cases}, \tag{B.3}$$

where $\mathbf{q}$ defines the joint configuration of the N involved DoFs to reach $\mathbf{x}_d$ and $\boldsymbol{\alpha}_d$, which are the end-effector's desired pose and orientation, respectively; K_x and K_α are the forward kinematic maps for the position and orientation part, respectively; $\mathbf{q}_r$ is used to keep the solution close to a given rest position in the joint space (weighted by an overall positive factor $w < 1$ and individual weights for each joint embeeded in the diagonal matrix W_r. The solution $\mathbf{q}$ is guaranteed to be within the physical bounds expressed by $\mathbf{q}_L$ and $\mathbf{q}_U$ (defined in Table B.1 and Table B.2 for the right and left end-effector, respectively).

When exploiting one `cartesian_solver` for the control of each end-effector, the DoFs belonging to the torso must be disabled in the minimisation problem. Otherwise, the low-level controllers receive different control commands from each `cartesian_solver`, which can damage the robot.

Bibliography

Ajoudani, A., Zanchettin, A. M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K., and Khatib, O. (2017). Progress and prospects of the human–robot collaboration. *Autonomous Robots*, pages 1–19.

Akgun, B., Cakmak, M., Jiang, K., and Thomaz, A. L. (2012). Keyframe-based learning from demonstration. *International Journal of Social Robotics*, 4(4):343–355.

Alciatore, D. and Ng, C. (1994). Determining manipulator workspace boundaries using the Monte Carlo method and least squares segmentation. *ASME Robotics: Kinematics, Dynamics and Controls*, 72:141–146.

Ardón, P., Ramamoorthy, S., and Lohan, K. S. (2018). Object affordances by inferring on the surroundings. In *Workshop on Advanced Robotics and its Social Impacts (ARSO)*. IEEE. Forthcoming.

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483.

Bajcsy, A., Losey, D. P., O'Malley, M. K., and Dragan, A. D. (2018). Learning from physical human corrections, one feature at a time. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 141–149. ACM.

Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). Robot programming by demonstration. In *Springer handbook of robotics*, pages 1371–1394. Springer.

Bristow, D., Tharayil, M., Alleyne, A. G., and Others (2006a). A survey of iterative learning control. *Control Systems, Institute of Electrical and Electronics Engineers*, 26(3):96–114.

Bristow, D. A., Tharayil, M., and Alleyne, A. G. (2006b). A survey of iterative learning control. *IEEE Control Systems*, 26(3):96–114.

Dautenhahn, K. and Nehaniv, C. L. (2002). The correspondence problem. In *Imitation in Animals and Artifacts, MIT Press*. MIT Press.

Fajen, B. R. and Warren, W. H. (2003). Behavioral dynamics of steering, obstable avoidance, and route selection. *Journal of Experimental Psychology: Human Perception and Performance*, 29(2):343.

Felip, J., Laaksonen, J., Morales, A., and Kyrki, V. (2013). Manipulation primitives: A paradigm for abstraction and execution of grasping and manipulation tasks. *Robotics and Autonomous Systems*, 61(3):283–296.

Gams, A., Nemec, B., Ijspeert, A. J., and Ude, A. (2014). Coupling movement primitives: Interaction with the environment and bimanual tasks. *IEEE Transactions on Robotics*, 30(4):816–830.

Goodrich, M. A. and Schultz, A. C. (2007). *Human Robot Interaction: A Survey*, volume 1.

Grunwald, G., Borst, C., Zöllner, J. M., et al. (2008). Benchmarking dexterous dual-arm/hand robotic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on Performance Evaluation and Benchmarking, Nice, France*.

Guenter, F., Hersch, M., Calinon, S., and Billard, A. (2007). Reinforcement learning for imitating constrained reaching movements. *Advanced Robotics*, 21(13):1521–1544.

Hartenberg, R. S. and Denavit, J. (1964). *Kinematic synthesis of linkages*. McGraw-Hill.

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 *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2587–2592. IEEE.

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.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles*, pages 396–404. Springer.

Kober, J., Bagnell, J. A., and Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274.

Kramberger, A., Gams, A., Nemec, B., and Ude, A. (2016). Generalization of orientational motion in unit quaternion space. In *IEEE-RAS International Conference on Humanoid Robots*, pages 808–813.

Lin, H.-C., Smith, J., Babarahmati, K. K., Dehio, N., and Mistry, M. (2018). A projected inverse dynamics approach for multi-arm cartesian impedance control. In *IEEE International Conference on Robotics and Automation*.

Lioutikov, R., Kroemer, O., Maeda, G., and Peters, J. (2016). Learning manipulation by sequencing motor primitives with a two-armed robot. In *Intelligent Autonomous Systems 13*, pages 1601–1611. Springer.

Makris, S., Tsarouchi, P., Surdilovic, D., and Krüger, J. (2014). Intuitive dual arm robot programming for assembly operations. *CIRP Annals-Manufacturing Technology*, 63(1):13–16.

Metta, G., Fitzpatrick, P., and Natale, L. (2006). YARP: yet another robot platform. *International Journal of Advanced Robotic Systems*, 3(1):8.

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.

Montesano, L., Lopes, M., Bernardino, A., and Santos-Victor, J. (2008). Learning object affordances: from sensory-motor coordination to imitation. *IEEE Transactions on Robotics*, 24(1):15–26.

Nguyen-Tuong, D. and Peters, J. (2011). Model learning for robot control: A survey.

Nicolescu, M. N. and Mataric, M. J. (2003). Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 241–248. ACM.

Norrlof, M. (2002). An adaptive iterative learning control algorithm with experiments on an industrial robot. *IEEE Transactions on robotics and automation*, 18(2):245–251.

Pairet, È. and Broz, F. (2018). Apprenticeship learning: A survey. *Manuscript in preparation*.

Pairet, È., Hernández, J. D., Lahijanian, M., and Carreras, M. (2018). Uncertainty-based Online Mapping and Motion Planning for Marine Robotics Guidance. In *Intelligent Robots and Systems (IROS), 2018 IEEE/RSJ International Conference on*. IEEE.

Park, D.-H., Hoffmann, H., Pastor, P., and Schaal, S. (2008). Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 91–98. IEEE.

Pastor, P., Hoffmann, H., Asfour, T., and Schaal, S. (2009). Learning and generalization of motor skills by learning from demonstration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 763–768. IEEE.

Pattacini, U., Nori, F., Natale, L., Metta, G., and Sandini, G. (2010). An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1668–1674. IEEE.

Pfeifer, R., Lungarella, M., and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *science*, 318(5853):1088–1093.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe.

Rai, A., Meier, F., Ijspeert, A., and Schaal, S. (2014). Learning coupling terms for obstacle avoidance. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 512–518. IEEE.

Rai, A., Sutanto, G., Schaal, S., and Meier, F. (2017). Learning feedback terms for reactive planning and control. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2184–2191. IEEE.

Schaal, S. and Atkeson, C. G. (2010). Learning control in robotics. *IEEE Robotics and Automation Magazine*, 17(2):20–29.

Smith, C., Karayiannidis, Y., Nalpantidis, L., Gratal, X., Qi, P., Dimarogonas, D. V., and Kragic, D. (2012). Dual arm manipulation: A survey. *Robotics and Autonomous systems*, 60(10):1340–1353.

Stenmark, M., Haage, M., Topp, E. A., and Malec, J. (2018). Supporting semantic capture during kinesthetic teaching of collaborative industrial robots. *International Journal of Semantic Computing*, 12(01):167–186.

Stulp, F., Raiola, G., Hoarau, A., Ivaldi, S., and Sigaud, O. (2013). Learning compact parameterized skills with a single regression. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, pages 417–422. IEEE.

Taylor, M. E. and Stone, P. (2009). Transfer Learning for Reinforcement Learning Domains : A Survey. *Journal of Machine Learning Research*, 10:1633–1685.

Topp, E. A. (2017). Knowledge for synchronized dual-arm robot programming. In *AAAI Fall Symposium Series 2017*. AAAI Press.

Ude, A. (1999). Filtering in a unit quaternion space for model-based object tracking. *Robotics and Autonomous Systems*, 28(2):163–172.

Ude, A., Nemec, B., Petrić, T., and Morimoto, J. (2014). Orientation in cartesian space dynamic movement primitives. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2997–3004. IEEE.

Zöllner, R., Asfour, T., and Dillmann, R. (2004). Programming by demonstration: dual-arm manipulation tasks for humanoid robots. In *IROS*, pages 479–484.