

UNIVERSITY OF CRETE
DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF SCIENCES AND ENGINEERING

Learning from Demonstration to Accomplish Robotic Manipulation Tasks

by

Maria Koskinopoulou



PhD Dissertation

Presented

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

Heraklion, December 2019

UNIVERSITY OF CRETE
DEPARTMENT OF COMPUTER SCIENCE
Learning from Demonstration to Accomplish Robotic Manipulation Tasks

PhD Dissertation Presented
by **Maria Koskinopoulou**
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

APPROVED BY:



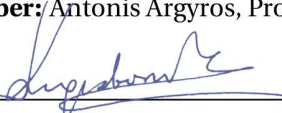
Author: Maria Koskinopoulou



Supervisor: Panos Trahanias, Professor, University of Crete, Greece



Committee Member: Antonis Argyros, Professor, University of Crete, Greece



Committee Member: Kostas Kyriakopoulos, Professor, National Technical University of Athens, Greece



Committee Member: Pietro Falco, Research Director, ABB Corporate Research, Vasteras, Sweden



Committee Member: Zoe Doulgeri, Professor, Aristotle University of Thessaloniki, Greece



Committee Member: Kostas Alexis, Assistant Professor, University of Nevada, Reno



Committee Member: Ales Ude, Professor, Jozef Stefan Institute, Ljubljana, Slovenia



Department Chairman: Angelos Bilas, Professor, University of Crete

Heraklion, December 2019

Acknowledgments

Along my PhD research road, I have met wonderful people who played a very important role in achieving this goal. First and foremost, I would like to express my unique appreciation and thanks to my supervisor Professor Panos Trahanias who has been a tremendous mentor for me. I would like to sincerely thank him for encouraging my research and allowing me to grow as a scientist. His constant and attentive guidance on both my research and career has been invaluable and his contributions on research ideas made my PhD experience mostly productive and exciting.

I also especially thank my advisory committee members, Professors Antonis Argyros and Kostas Kyriakopoulos for their support even at hardship. I'm grateful that you made my defense an enjoyable moment, and for your excellent comments and suggestions during these years. In addition, a warm thanks to the four members that together with the advisory committee constituted my PhD examination committee, Dr. Pietro Falco, Prof. Zoe Doulgeri, Prof. Kostas Alexis and Prof. Ales Ude. It was a great honor for me having you in my PhD committee.

Both the Department of Computer Science, University of Crete and the Institute of Computer Science, Foundation for Research and Technology-Hellas (FORTH), where this thesis was conducted, have provided a wealth of academic stimuli as well as a pleasant and creative environment for which I am very thankful. The Computational Vision and Robotics Laboratory (CVRL) at FORTH is also acknowledged for providing all the required lab-equipment for the experimental part of my work. Financial support for my PhD has been provided by: (a) the Institute of Computer Science, FORTH; (b) the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT); and (c) the program "Research stay for PhD students at ABB Corporate Research, Vasteras, Sweden".

Special thanks to my friend and fellow student Stylianos Piperakis for his support and valuable advice during these years, and also, our CVRL lab-team members and collaborators. In particular, I would like to mention the various contributions of Michalis Maniadakis, Fredy Raptopoulos, Markos Sigalas, Manos Papadakis, Manos Hourdakakis, Maria Pateraki, Nikos Tavoularis, Artemis Agiomavriti, Thodoris Evdaimon, Avgousta Hatzidaki.

During my PhD course I had the unique chance to pursue a seven-month-internship at ABB Corporate Research, in Vasteras, Sweden. I am very grateful to Dr. Pietro Falco, who leaded my research stay at ABB: his pioneer ideas and formulations about robot learning were particularly valuable and constructive. Overall, my stay at ABB was an absolutely

nice experience. I had the chance to meet great people and important collaborators and found a very warm atmosphere. Thank you for this Pietro Falco, Sotiris Stavridis, Henrik Hellström and Davide Santini.

A big thanks goes also to my important lifelong friends: Alkistis Palaiologou, Natasa Gotzi, Marilena Kalitsounaki, Maria Tsiourva, Rea Pantazi, Sotiria Efthimiadi, Argyris Pappas, Sophie Paneri, and Giorgos Makridakis, who supported me all the way towards my goal, continuously being there and encouraging me not to give up during failures or hard personal moments.

Of course, I couldn't forget my family: my parents and especially my beloved brother, my adorable godparents and grandmother for supporting me emotionally throughout not only while pursuing this thesis but most of all, in every step of my life. Without their support, I wouldn't have made it.

Abstract

The current PhD thesis addresses the formulation and implementation of a methodological framework for robot Learning from Demonstration (LfD). The latter refers to methodologies that develop behavioral policies from example state-to-action mappings. To this end, we study the reciprocal interaction of perception and action, in order to teach robots a repertoire of novel action behaviors. Based on that, we design, develop and implement a robust imitation framework, termed IMFO (IMitation Framework by Observation), that facilitates imitation learning and relevant applications in human-robot interaction (HRI) tasks. IMFO can cope with the reproduction of learned (i.e. previously observed) actions, as well as novel ones. Mapping of human actions to the respective robotic ones is achieved via an indeterminate depiction, termed latent space representation. The latter accomplishes a compact, yet precise abstraction of action trajectories, effectively representing high dimensional raw actions in a low dimensional space.

Moreover, throughout this thesis, we examine the role of time in LfD by enhancing the aforementioned framework with the notion of learning both the spatial and temporal characteristics of human motions. Accordingly, learned actions can be subsequently reproduced in the context of more complex time-informed HRI scenarios. Unlike previous LfD methods that cope only with the spatial traits of an action, the formulated scheme effectively encompasses spatial and temporal aspects. Extensive experimentation with a variety of real robotic platforms demonstrates the robustness and applicability of the introduced integrated LfD scheme.

Learned actions are reproduced under the high level control of a time-informed task planner. During the implementation of the studied scenarios, temporal and physical constraints may impose speed adaptations in the performed actions. The employed latent space representation readily supports such variations, giving rise to novel actions in the temporal domain. Experimental results demonstrate the effectiveness of the proposed enhanced imitation scheme in the implementation of HRI scenarios. Additionally, a set of well defined evaluation metrics are introduced to assess the validity of the proposed approach considering the temporal and spatial consistency of the reproduced behaviors.

A noteworthy extension of the above regards force-based object grasping for executing sensitive manipulation tasks. This is also treated in the current thesis via a novel supervised learning scheme, termed SLF (Supervised Learning for Force-based manipulation). SLF is formulated as a three-stage process: (a) supervised trial-execution in simulation to acquire sufficient training data; (b) training to facilitate grasp learning with suitable

robot-arm pose and lifting force; (c) grasp execution in simulation. Subsequently, following sim-to-real transfer, operation in real environments is achieved in addition to simulated ones, generalizing also for objects not included in the trial sessions. The proposed learning scheme is demonstrated in object lifting tasks where the applied force varies for different objects with similar contact friction coefficients, and likewise the grasping pose. Experimental results on the manipulator YuMi show that the robot is able to effectively reproduce demanding lifting and manipulation tasks after learning is accomplished.

In summary, our thesis has studied LfD and has contributed with a novel approach that introduced latent space representations to encode the action characteristics. A framework implementation (IMFO) of our approach allowed extensive experimentation and also conduction of HRI scenarios. The inclusion of temporal aspects in our approach enhanced it to cope with complex, real-life interactions. Finally, the extension of IMFO with force-based grasping facilitated manipulation tasks with sensitive objects.

Keywords: Learning from Demonstration, Machine Learning, Latent Representation, Neural Networks, Human-Robot Interaction, Temporal Planning, Force-based Manipulation.

Περίληψη

Η παρούσα διδακτορική διατριβή αφορά τη μελέτη, ανάπτυξη και εφαρμογή, μεθόδων Μηχανικής Μάθησης μέσω Παρατήρησης (Learning from Demonstration) με στόχο την ρομποτική αναπαραγωγή δράσεων χειρισμού. Η μεθοδολογία αυτή στηρίζεται στην δημιουργία μιάς αντιστοίχισης (mapping) μεταξύ της κινηματικής του ανθρώπινου χεριού και ενός ρομποτικού βραχίονα, ή πιο συγκεκριμένα μεταξύ του πολυδιάστατου χώρου των κινήσεων του ανθρώπου (human actor) με τον επίσης πολυδιάστατο χώρο δράσης του ρομπότ. Η συσχέτιση των ανθρώπινων ενεργειών με αντίστοιχες ρομποτικές επιτυγχάνεται μέσω μιας άδηλης αναπαράστασης, που ονομάζεται λανθάνουσα απεικόνιση χώρου (latent space). Πιο συγκεκριμένα, μελετάμε την αμοιβαία αλληλεπίδραση της αντίληψης και της δράσης, προκειμένου να διδάξουμε τα ρομπότ μια ποικιλία από νέες κινήσεις χειρός. Ως εκ τούτου, στα πλαίσια της παρούσας διατριβής σχεδιάσαμε, αναπτύξαμε και εφαρμόσαμε ένα μεθοδολογικό πλαίσιο μάθησης μέσω παρατήρησης, το οποίο ονομάζεται IMFO (Imitation Framework by Observation), που διευκολύνει την αναπαραγωγή μαθημένων και νέων κινήσεων χειρισμού από ένα ρομπότ (manipulation tasks) και, παράλληλα, έχει ευρεία εφαρμογή σε σενάρια αλληλεπίδρασης ανθρώπου-ρομπότ (HRI) σε καθημερινά περιβάλλοντα.

Επιπλέον, σε αυτή τη διατριβή, εξετάζουμε το ρόλο της χρονικής διάρκειας εκτέλεσης μιας κίνησης μέσα από τη διαδικασία μάθησης από παρατήρηση, ενισχύοντας το διαμορφωμένο πλαίσιο IMFO με την δυνατότητα αναπαράστασης και αναπαραγωγής τόσο των χωρικών όσο και των χρονικών χαρακτηριστικών των ανθρώπινων κινήσεων. Κατά συνέπεια, οι κινήσεις που μαθαίνονται μέσα από το προτεινόμενο πλαίσιο μπορούν να εφαρμοστούν σε πιο σύνθετα σενάρια HRI, όπου η χρονική αλληλουχία των δράσεων είναι σημαντική. Σε αντίθεση με άλλες μεθόδους μάθησης μέσω παρατήρησης (LfD) που περιγράφουν την εκτελούμενη δράση μόνο με βάση τα χωρικά χαρακτηριστικά της, η προτεινόμενη μεθοδολογία περιλαμβάνει τη μελέτη των χωροχρονικών πτυχών μιάς κίνησης. Η ευρεία εφαρμογή της μεθοδολογίας σε πραγματικά πειράματα και διάφορες ρομποτικές πλατφόρμες αποδεικνύει την αποτελεσματικότητα και αποδοτικότητα του προτεινόμενου συστήματος LfD. Παράλληλα, κατά την εφαρμογή της συγκεκριμένης μεθοδολογίας, οι χρονικοί και φυσικοί περιορισμοί μπορεί να επιφέρουν προσαρμογές στην ταχύτητα των αναπαραγόμενων δράσεων, ενισχύοντας το προτεινόμενο πλαίσιο μάθησης και εφαρμογής του. Επιπρόσθετα, εισάγεται ένα σύνολο καλά καθορισμένων μετρικών αξιολόγησης (evaluation metrics)

για να αποτιμηθεί η εγκυρότητα της προτεινόμενης προσέγγισης λαμβάνοντας υπόψη τη χρονική και χωρική συνέπεια των αναπαραγόμενων συμπεριφορών.

Μια αξιοσημείωτη επέκταση του προαναφερθέντος πλαισίου αναφέρεται στην εκμάθηση της δύναμης που επιβάλλεται από τον χρήστη για την επιτυχημένη εκτέλεση λεπτών χειρισμών. Αυτή η διαδικασία παρουσιάζεται επίσης στην παρούσα διατριβή μέσω ενός νέου πλαισίου εποπτευόμενης μάθησης, το οποίο ονομάζεται SLF (Supervised Learning scheme for Force-based manipulation). Το SLF διατυπώνεται ως μία διαδικασία τριών σταδίων: (α) επιβλεπόμενη διαδικασία εκτέλεσης κινήσεων χειρισμού σε προσομοίωση για την απόκτηση επαρκών δεδομένων, (β) διαδικασία εκπαίδευσης (training) για τη διευκόλυνση της μάθησης κινήσεων χειρισμού με την κατάλληλη προσαρμογή του καρπού και της δύναμης πιασίματος και μεταφοράς και (γ) εκτέλεση της κίνησης από ρομποτικό βραχίονα σε προσομοίωση. Στη συνέχεια, με τη χρήση της μεθόδου sim-to-real transfer, επιτυγχάνεται αναπαραγωγή των μαθημένων δράσεων σε πραγματικά περιβάλλοντα γενικεύοντας την εφαρμογή του πλαισίου μάθησης σε επιπλέον συνθήκες χειρισμού εύθραυστων αντικειμένων. Τα αποτελέσματα με τη χρήση του ρομποτικού βραχίονα YuMi, σε πειράματα με διαφορετικά αντικείμενα με παρόμοιους συντελεστές τριβής, και εναλλακτικές πόζες πιασίματος, αποδεικνύουν ότι το ρομπότ είναι σε θέση να αναπαράγει αποτελεσματικά απαιτητικές κινήσεις μεταφοράς και χειρισμού μετά την ολοκλήρωση της διαδικασίας μάθησης.

Συνοπτικά, η παρούσα διατριβή μελετά την διαδικασία μάθησης μέσω παρατήρησης συνεισφέροντας με μια νέα προσέγγιση που εισάγει την μελέτη δράσεων χειρισμού αντικειμένων μέσα από έναν χώρο μειωμένων διαστάσεων, για την εύκολη και συμπαγή κωδικοποίηση των επιμέρους χαρακτηριστικών των δράσεων. Η ανάπτυξη της προτεινόμενης προσέγγισης (IMFO) επιτρέπει την εφαρμογή της σε πειραματικές διαδικασίες μεταφοράς αντικειμένων σε πραγματικά περιβάλλοντα και σενάρια συνεργασίας ανθρώπου-μηχανής (HRI). Ταυτόχρονα μελετώνται τα χρονικά χαρακτηριστικά των κινήσεων ώστε να ενισχυθεί η εφαρμογή της μεθόδου σε σύνθετες, πραγματικές συνθήκες που απαιτούν χρονική ακρίβεια αναπαραγωγής. Τέλος, η διαμόρφωση μιας γενικευμένης διαδικασίας εποπτευόμενης μάθησης για τον χειρισμό εύθραυστων αντικειμένων αναβαθμίζει περαιτέρω το αρχικό πλαίσιο μάθησης.

Contents

Acknowledgments	v
Abstract	vii
Abstract in Greek	ix
Table of Contents	xi
List of Figures	xiii
List of Tables	xvii
1 Introduction	1
1.1 Motivation	1
1.2 Thesis Scope	2
1.3 Contributions of this Dissertation	5
1.4 Thesis Publications	6
1.5 Outline of Dissertation	8
2 State of the Art	9
2.1 Historical Context and Neurophysiological Basis	9
2.2 Observation - Perception of knowledge	11
2.3 Representation of knowledge	13
2.3.1 Policy Formulation	13
2.4 Encoding of Knowledge	15
2.4.1 Dynamical Systems Models	16
2.4.2 Stochastic Models	18
2.4.3 Latent Representations	21
2.5 How to imitate: Learning a policy-mapping	23
2.5.1 Metrics of Imitation	27
2.6 Robot Reproduction	28
2.6.1 Applications - Experimental Scenarios	28
2.6.2 Reinforcement Learning: Real-time adaptation to the current environment	34
2.7 Learning Force-based Manipulation	34
3 Methodological Framework for Robot Learning by Observation	37
3.1 IMFO Formulation	37
3.1.1 Training Phase	38
3.1.2 Robot Reproduction Phase - Inverse Projection	41
3.2 IMFO Evaluation	42

3.2.1	Scenario WRITE	44
3.2.2	Scenario PICK-PLACE	49
3.3	Applications in HRI	50
3.3.1	Novel Action Execution	50
3.3.2	Cooperative Task Execution	53
3.4	Chapter highlights	57
4	Time-informed Action Learning and Reproduction	59
4.1	Spatio-Temporal Representation of Actions	60
4.1.1	Learning Phase	60
4.1.2	Speed Inference Based on Temporal Information	63
4.1.3	Time-informed Robotic Action Execution	64
4.2	Experimental Results	66
4.2.1	Performance Metrics of Spatio-temporal Accuracy	67
4.2.2	Use Case Application - Service Scenario	70
4.3	Chapter Highlights	72
5	Force-based Object Manipulation	73
5.1	SLF Formulation	74
5.1.1	Supervised Trial Execution in Simulation	75
5.1.2	Training to Facilitate Force-based Manipulation	77
5.1.3	Grasp-lifting Execution	80
5.2	Experimental Results	80
5.2.1	Simulation Experiments	81
5.2.2	Transfer Learning for Real Robot Experiments	83
5.3	Chapter Highlights	87
6	Discussion	89
6.1	Conclusions	89
6.2	Directions for Future Work	91
	Bibliography	93

List of Figures

2.1	Conceptual sketch of an imitation learning system.	10
2.2	The two levels of knowledge representation as proposed by Billard and colleagues in [62].	14
2.3	Policy derivation as proposed in [1], using the generalization approach for determining (a) an approximation to the state given the action mapping function, (b) a dynamic model of the system, and (c) a plan of sequenced actions.	16
2.4	Dynamical System Framework [3] commonly used in LfD.	19
2.5	Stochastic processes in LfD.	21
2.6	Imitation from motion capture as described in [96].	23
2.7	An example of two-dimensional dynamics learned from three demonstrations comparing five different methods: GMR, LWPR, GPR, BM, and SEDS, captured by [45].	26
2.8	(a) LfD framework proposed in [47] [71]. (b) Hoap humanoid robot putting an object into a box, kinesthetically driven by a human [71].	29
2.9	Left: Schematical overview of the proposed framework. Right: Tactile correction for learning grasp adaptation. The teacher indicates adaptability within the compliance constraints of a hand posture (Correction). The learner then replays the sequence of corrected poses (Reproduction). The final learned model is able to adapt the pose in response to different contact signatures (Perturbation). Objects: small can (left panels), ruler (right, top), large can and box (right, bottom).	30
2.11	Imitation from motion capture as described in [96].	32
2.13	Overview of the proposed architecture in [114].	35
3.1	IMFO Training Phase.	38
3.2	Left: Example of joint tracking via color detection for the actor; Right: Kinesthetic teaching for the JACO arm.	39
3.3	IMFO Reproduction Phase.	42
3.4	Derived latent representations for actor and JACO arm, respectively.	43
3.5	Right column: Demonstration of actions corresponding to the letters of the word “JACO”; Left column: Robotic reproduction of the same actions.	45

3.6	Illustration of the novel latent trajectories with each variance: Left: HLS representations for the four novel actions. Right: Corresponding RLS representations.	46
3.7	Error bars denoting consistency and repeatability in RLS and RAS.	48
3.8	Scenario PICK-PLACE. Top row: Demonstration of a novel pick-and-place action; Bottom row: Robotic reproduction of the demonstrated action. . . .	49
3.9	(a) Kinesthetic training of the robotic arm. (b) A novel human motion behavior (left), reproduced by the robotic arm (right).	51
3.10	Extracted Latent Hyperspace formulated by five sample trajectories marked as <i>D1 – D4</i> and <i>NOVEL ACTION</i> , accomplished by actor and JACO, respectively: (a) Actor’s Latent Space, and (b) JACO’s Latent Space.	52
3.11	Extracted NAO’s Latent Hyperspace, formulated by five sample trajectories <i>T1 – T5</i> , accomplished by actor and NAO, respectively: (a) Actor’s Latent Space, and (b) NAO’s Latent Space.	54
3.12	HRC scenario whereby a robot assumes a learned action behavior to close an open closet door. Top: JACO arm; Bottom: NAO humanoid robot arm. . .	55
3.13	Trajectories of end-effectors’ movement during the two aforementioned HRC tasks.	56
3.14	Time-aware multi-agent symbiosis.	57
4.1	Schematic overview of the learning process.	61
4.2	Grasping configuration poses; i.e. Grasping handle encoded as “0” (Left), “1” (Center) and “2” (Right).	61
4.3	Graphical illustration of the intersection of two fuzzy intervals.	64
4.4	Block diagram representation of time-informed robotic action execution. . .	65
4.5	Formulated latent space representations for human actor (left) and JACO arm (right), respectively. Actions performed at normal speed are represented by magenta circles, whereas slow and fast actions (in low and high speed respectively) are depicted with cyan and yellow circles, respectively.	68
4.6	Snapshots from the serving scenario.	71
5.1	Left: ABB YuMi Arm. Right: Ridgeback-YuMi robot model in Gazebo simulator. Red-Green-Blue lines denote the XYZ axes, respectively, for the world, the left arm’s end-effector and the wrist.	74
5.2	Conceptual scheme of the proposed SLF.	75
5.3	Left: Configuration grasping poses for green objects, marked as “light” grasping <i>Pose#1</i> . Middle: the “normal” grasping <i>Pose#2</i> . Right: the “heavy” one <i>Pose#3</i>	76
5.4	Block diagram of the proposed DNN topology.	78

5.5	Schematic block diagram of the proposed RNN topology.	79
5.6	Integrated system design.	80
5.7	Top row: Snapshots from failure case of lifting object because of slipping. Bottom row: Successful reproduction of lifting motions after training of the proposed scheme.	82
5.8	Learning curves for the cases of DNN and RNN training.	83
5.9	RMSE for the six arm joints and the wrist's force from the 1000 simulations; green bars indicate the prediction errors for the green class of objects, blue bars for the blue one and red bars for the red one; black lines indicate the standard deviation from the corresponding mean values; bold bars indicate the errors with the DNN implementation and light bars with the RNN in all cases.	84
5.10	Mean trajectories of object lifting manipulation during training via kines- thetic teaching. Green Curve: average trajectory for reach-grasp-lift the green class object. Blue Curve: average trajectory for reach-grasp-lift the blue class object. Red Curve: average trajectory for reach-grasp-lift the red class object. Standard deviations are marked with the lighter thick line for each trajectory. Black circle denotes the starting point, whereas black and yellow crosses indicate the grasp and place points, respectively, as they were computed via the zero velocity crossing algorithm.	85
5.11	Left: a black box of 0.01kg mass for the light class; Middle: a remote control of 0.1kg for the normal one; Right: a green box of 0.5kg that represents the heavy class of objects.	86
5.12	Sim-to-Real learning curve.	86
5.13	Two representative grasp-lifting shots from the objects used for validation, namely tape and green box, are shown in this figure.	87
5.14	Snapshots from the implemented egg-lifting experimental setup.	88

List of Tables

3.1	Indicative Errors for the novel action behavior.	50
3.2	Performance Metrics	56
4.1	Mean error values (μ) and standard deviation (σ) of the computed metrics for each group of speed, namely Low (L), Normal (N) and High (H)	70
5.1	Experimental Results for novel object manipulation	87

Chapter 1

Introduction

1.1 Motivation

Learning new behaviors enables robots to expand their repertoire of actions and effectively learn new tasks. The development of such policies by hand is often very challenging, so machine learning techniques have been proposed to cope with the relevant inherent complexity. A distinct approach to policy learning in a natural way, regards the so-called Learning from Demonstration (LfD), also referred to in the literature as Imitation Learning or Programming by Demonstration (PbD) [1].

Learning from Demonstration (LfD) has become an important topic in robotics research with notable applications in broader areas, such as object manipulation, human-robot interaction, robotic companions, and human-robot collaborative task execution [2–6]. LfD can by itself reveal remarkable advantages as summarized below:

- LfD is a powerful mechanism for reducing the complexity of the search spaces for learning. When observing either valid or non-valid examples, one can potentially reduce the search for a possible solution, by either starting the search from the observed valid solution, or conversely, by eliminating from the search space what is known as unacceptable solution [7–9].
- Imitation learning offers an implicit means of training a machine, such that explicit and tedious programming of a task can be minimized or even eliminated. Imitation learning is thus a natural means of endowing robotic machines with new capabilities [10].
- Studying and modeling the coupling of perception and action, which is at the core of imitation learning, greatly facilitates the understanding of the mechanisms by which the self-organization of the perception-action loop could arise during development [1, 8].

LfD methods were originally introduced in order to achieve faster and more accurate learning of robotic-manipulation tasks, in contrast to tedious explicit programming of

robotic behaviors [11, 12]. In this thesis, we investigate the reciprocal interaction of perception and action, in order to teach robots a variety of novel action behaviors. A major issue in motor learning is concerned with the difficulty of learning motor skills that are high dimensional. To deal with this problem, this thesis placed emphasis on representing the high-dimensional motor space in a subspace of lower dimensionality, termed *latent space*. Accordingly, our work focused on the design and implementation of a robust imitation framework based on LfD that facilitates the execution of arm-motion behaviors and the application of robots in human-robot interaction (HRI) tasks in everyday environments.

It is worth noting that depending exclusively on one perception channel greatly limits the information that a robot may acquire and use for learning. Humans are endowed with a very rich multimodal perception system [13, 14], which is heavily used when performing physical interaction tasks [15, 16]. In contrast research on LfD has focused on learning tasks using unimodal sensing, i.e. vision. Taking inspiration from the above, this thesis also proposes to merge vision and temporal cognition in order to improve the learning process.

While most works in the field have focused on learning the kinematics of motions, research regarding force-based skills has been very limited. Force-based grasping and manipulation may be crucial when contact with the environment takes place, mostly in manipulation of objects [17], and physical interaction with a partner [18]. Relevant works have shown that the human central nervous system is composed of internal models that control the interactions between the body and its surroundings [19, 20]. Some of these models are dedicated to predicting the outcome or anticipating force resulting from an individual's conscious action. Inspired by the latter, our work also addressed how robots may learn force-based manipulation tasks from human demonstrations.

1.2 Thesis Scope

The current PhD thesis aims at formulating, developing and implementing a novel methodological framework that facilitates robots to acquire human-like behavioral acts by observing human demonstrators. Accordingly, the introduced LfD framework, termed as IMFO (IMitation Framework by Observation), facilitates learning and reproduction of either observed or novel actions. Mapping of human actions to the respective robotic ones is achieved via an intermediate depiction, termed latent space representation. The latter accomplishes a compact, yet precise abstraction of action trajectories, effectively representing high dimensional raw actions in a low dimensional space.

In addition, despite the significant number of works exploring LfD, there are important parameters of action implementation that have not been sufficiently studied yet. In particular the role of temporal information in the computational representation and reproduction of actions remains poorly understood. The execution time of activities is directly

linked to the speed of reproduced actions. The ability to adjust speed is fundamental for humans, allowing to deal with cases where the physical properties of manipulated objects impose constraints (e.g. slow down to move a glass full of water), or emergency-like situations (e.g. speed up to accomplish a goal within certain time constraints). It is therefore important to study how spatio-temporal variations affect the representation and execution of actions considered in LfD scenarios.

Accordingly, this thesis also addressed speed adaptation in the context of IMFO, considering how speed shapes the low dimensional latent representation of the demonstrated actions. Specifically, the actions considered in the current study are arm motions, although the introduced formulation can be readily generalized to other human actions. In short, the proposed method regards augmenting the algorithm that implements the transformation from the full configuration space to the compact latent space, with temporal information that affects execution speed. The compressed representation of similar actions with different temporal characteristics shows that speed plays a major role in the derived latent representation, effectively separating similar arm motions that are executed at different speeds. Accordingly, the latter actions assume unambiguous latent space representations when only speed of execution varies, allowing thus the accurate reproduction of acts with different velocities.

The proposed spatio-temporal formulation of IMFO readily lends itself to integration with time-informed planning approaches [21] to effectively address temporal constraint satisfaction in real-world scenarios. Following our previous work on latency estimation [22], the composite system is capable to cope with cases where completion of certain behaviors is expected to be delayed. The elimination of latency is accomplished through the estimation of a requested, reduced time for action completion. Taking advantage of learning from demonstration actions at different speeds, we select the action implementation that best matches the requested completion time, therefore facilitating the realization of the composite behavior within the predefined time limits.

Finally, the current thesis copes with the topic of impedance control through observation, by studying the more involved case of force-based object manipulation, that is the case where the appropriate required force to lift and manipulate a rigid object varies for different objects and object positions relative to the arm. We present a Supervised Learning scheme for executing sensitive Force-based manipulation tasks. The proposed scheme, termed SLF, assumes prior knowledge of the position and the weight of the target object which implicitly indicates the suitable force that has to be applied to achieve a precision-grip lifting movement. SLF learns the required force via simulated training, whereas a neural network structure is employed to map the observed lifting actions to actual grasping commands. The proposed learning scheme has been demonstrated in object lifting and manipulation tasks where the applied force varies for different objects with similar contact friction coefficients.

Research Objectives. Though LfD has proven a successful tool for robot policy development, there still exist open areas for research, several of which are covered in this thesis. In particular, this thesis establishes a novel methodological framework for encoding novel manipulation action behaviors and systematically pursues the following concrete objectives:

- The majority of LfD methodologies are specific to executed tasks and also the employed robotic embodiments and do not easily scale to behaviors beyond the learned ones. The development of a more general-purpose approach may greatly facilitate robotic task execution in broader contexts. The proposed methodology focuses on the formulation of an abstract and compact intermediate latent space that abstracts the irrelevancies from different embodiments and also supports the execution of novel behaviors.
- The temporal aspects of an unfolding task are usually not considered in LfD systems. Still, rich information may be derived by including relevant temporal characteristics in LfD scenarios. To this end, this thesis examines the role of time in the latent representations of both the robotic and human demonstrator's workspaces and studies how speed shapes the low dimensional latent representations of the demonstrated actions.
- Besides patio-temporal LfD, training to acquire impedance based control and hence relevant behaviors is important in various tasks. This is addressed in our work via the formulation of a Supervised Learning scheme for Force-based (SLF) lifting tasks. SLF greatly facilitates reproduction of tasks that involve manipulation of sensitive objects.
- A standardized set of evaluation metrics that would facilitate rigorous assessment of the accuracy of the results and the level of imitation during robotic reproduction is largely missing. The development of generally accepted metrics as well as the compilation of benchmark-data is of utmost importance towards the systematic assessment of LfD implementations. As a result, this thesis formulates well-defined metrics to effectively assess the validity and accuracy of the proposed methodologies.
- LfD furnishes robots with the ability to learn and reproduce useful tasks. Accordingly, it may comprise an indispensable building block of more general Human-Robot Interaction (HRI) and/or Human-Robot Collaboration (HRC) frameworks. To this end, the proposed LfD approaches are introduced, studied and assessed in HRI and HRC scenarios.

1.3 Contributions of this Dissertation

As already mentioned above, the current thesis addressed several topics in the broader LfD field. More specifically, our initial work focused on the formulation and development of the Imitation Framework by Observation (IMFO) for learning manipulation tasks by observing a human demonstrator. The relevant formulation is based on a low-dimensional representation, termed latent space, with the aim of abstracting the trajectory and kinematic representations of observed actions. As it is common practice in LfD methodologies, IMFO formulation comprises three distinct phases, i.e. (i) information acquisition (vision based data acquisition), (ii) information encoding (dimensionality reduction approach), and (iii) robotic reproduction (regardless of the embodiment, i.e. robotic platform). In turn, we placed emphasis on the investigation of the temporal aspects of a demonstrated action behavior and how the temporal characteristics can be transferred to the robotic reproduction. To this end, the basic IMFO formulation has been extended in order to allow both spatial and temporal characteristics of the demonstrated actions to be considered. Finally, our research dealt with another challenging problem, namely learning force-based robot behaviors through observation. The above comprise the main contributions of this dissertation, as explained in more detail below.

- Given a set of demonstrated arm-movements, dimensionality reduction is used to find a low-dimensional manifold (the latent space) on which the demonstrations lie. The space defined by this manifold can then be used as the state-space representation of the relevant task. Our main contribution consists in the formulation of a framework that implements the above, allowing at the same time preservation of the useful properties of observed actions. Specifically, the introduced dimensionality reduction features particularly suitable assets, i.e. (i) different actions result in well separated trajectories in the latent space, (ii) the neighbouring properties of a 3D trajectory are maintained, via the continuous curves that are formed in the latent space, and (iii) the abstract representation is formulated independently of the embodiment of the actual demonstrator.
- By expanding the initial formulation of IMFO to involve temporal aspects of the demonstrated actions, we enhance robots that learn from demonstration to execute actions at variable speeds. Moreover, we shed light to the key role of temporal information in obtaining compact, latent representations of action behaviors. Additionally, the proposed extended scheme demonstrates the reversibility of the spatio-temporal aspects of actions from the low dimensional latent space to the robotic action space and also effectively combines at the same time temporal planning with LfD.
- Spatio-temporal LfD is further enhanced with the introduction of impedance con-

trol for object grasping. A supervised learning scheme for executing sensitive force-based manipulation tasks is also developed in this thesis. Accordingly, a robotic system is trained to execute various actions by performing multiple trials of lifting boxes that vary on the mass and the appropriate grasping configuration pose. A structured Neural Network module is employed to encode a force-based task where the model variables come mainly from vision (i.e. target object's position and encoded object's mass) and robot's configuration (robot's end-effector position and orientation). Such encoding approach is then the basis to define a force-based LfD model that controls the robot motion, and allows to extend the LfD paradigm to other scenarios, for instance those where the robotic action is based on specific impedance-based behaviors, instead of merely following a given trajectory. Consequently, the proposed force-based approach accomplishes to generalize on the appropriate force required to be applied on a fragile object for effective lifting and manipulation.

- Evaluation metrics are of utmost importance in LfD developments since they facilitate rigorous and objective assessment of relevant implementations. Accordingly, the metrics introduced in our research serve this goal, and fill a known gap in the sector [1]. Interestingly, the mentioned metrics are quite general and avoid limiting assumptions, and hence applicable in a variety of LfD systems and scenarios.
- Action learning and reproduction via observation can greatly support advanced Human-Robot Interaction (HRI) and/or Human-Robot Collaboration (HRC) setups. The latter may benefit from the robotic capacity to naturally execute learned and novel actions in various contexts. This has been systematically pursued in the current thesis, whereby developments in LfD have been incorporated and rigorously tested and assessed in HRI/HRC scenarios.

1.4 Thesis Publications

Most of the material that is presented in the core technical chapters, i.e. Chapter 3, 4 and 5, has already been published in or submitted to peer-reviewed conference proceedings and scientific journals. References to the publications are given below.

- (1) **Maria Koskinopoulou**, Pietro Falco, and Panos Trahanias, A Supervised Learning Scheme to Accomplish Force-based Object Manipulation, *Under Review*, IEEE International Conference on Robotics and Automation (ICRA 2020), Submitted Sep. 2019.

- (2) **Maria Koskinopoulou**, Michail Maniadakis, and Panos Trahanias, Speed Adaptation in Learning from Demonstration through Latent Space Formulation, *Robotica*, pp.1-13, Sep. 2019 [23].
- (3) **Maria Koskinopoulou**, Michail Maniadakis, and Panos Trahanias, Learning Spatio-temporal Characteristics of Human Motions Through Observation. *Advances in Service and Industrial Robotics. RAAD 2018. Mechanisms and Machine Science*, vol 67. Springer, Cham., pp. 82–90, 2018 [24].
- (4) **Maria Koskinopoulou** and Panos Trahanias, A Methodological Framework for Robotic Reproduction of Observed Human Actions: Formulation using Latent Space Representation, in *proc. ACM/IEEE Intern Conf. on Humanoids 2016*, pp. 565-572, Nov. 15-17 2016 [25].
- (5) **Maria Koskinopoulou**, Stylianos Piperakis, and Panos Trahanias, Learning from demonstration facilitates human-robot collaborative task execution, in *proc. 11th ACM/IEEE Intern Conf. on Human-Robot Interaction (HRI 2016)*, pp.59-66, March 7-10 2016 [26].

Contributions to other works

- (1) Michail Maniadakis, Hourdakos Emmanouil, Markos Sigalas, Stylianos Piperakis, **Maria Koskinopoulou**, Panos Trahanias, Time-aware Multi-agent Symbiosis, *Under Review*, Submitted for publication in *Journal Frontiers in Robotics and AI*, Oct. 2019.
- (2) Emmanouil Papadakis, Fredy Raptopoulos, **Maria Koskinopoulou** and Michail Maniadakis, On the use of vacuum technology for applied robotic systems, Accepted for publication in *proc. of ICMRE 2020-IEEE 6th International Conference on Mechatronics and Robotics Engineering*, 12-15 Feb. 2020.
- (3) Stylianos Piperakis, **Maria Koskinopoulou** and Panos Trahanias, Non-Linear State Estimation for Humanoid Robot Walking, Double Publication: *IEEE Robotics and Automation Letters (RA-L)*, vol.3, pp. 3347-3354, 2018/10 and in *proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018 [27], [28].

Announcements

- (1) **Maria Koskinopoulou** and Panos Trahanias, (2016), Robots Learn Actions and Cooperative Tasks by Imitation, Section: Research and Innovation, *ERCIM News* No. 104.

- (2) **Maria Koskinopoulou** and Panos Trahanias, (2015), Robot learning by Demonstration using Gaussian Process Latent Variable Model, Workshop: Women in Robotics, Robotics: Science and Systems (RSS) 2015. (Oral Presentation+Poster).

1.5 Outline of Dissertation

This thesis is structured in the following chapters:

- Chapter 2 presents an overview of the state of the art of LfD methodologies and implementations, starting from the historical context and Neurophysiological basis of LfD. Well-known learning algorithms as well as the different ways of transferring skills to a robot are treated, with focus on the three distinct phases that appear in the literature of LfD, namely (i) observation/perception of knowledge, (ii) representation of knowledge, and (iii) encoding of knowledge. Applications in the field are briefly described in this chapter as well.
- Chapter 3 presents our LfD formulation using Dimensionality Reduction as well as the design and implementation of IMFO. The latter is presented in sufficient detail, with particular focus on the observation, encoding and reproduction of actions. Experimental results and a quantitative evaluation in HRI tasks are also presented.
- Chapter 4 aims at revealing the spatio-temporal representation of actions through LfD. The IMFO framework is extended in order to deal with both spatial and temporal characteristics of the demonstrated actions, and is combined with a high level Daisy Planner to address the time-informed planning in multi-agent setups.
- Chapter 5 deals with the Supervised Learning scheme for Force-based (SLF) grasping to achieve sensitive object manipulation. The appropriate force that is required to be applied on a fragile object for effective lifting and manipulation through a feed-forward neural network topology is examined in this chapter.
- Chapter 6 wraps-up the current PhD thesis and discusses new possible routes of research arising from the work presented in the previous chapters. Issues concerning robust temporal information encoding, learning of impedance-based behaviors, role determination in HRI using temporal and force information, among others, are discussed here.

Chapter 2

State of the Art

2.1 Historical Context and Neurophysiological Basis

Robot Learning from Demonstration has its roots in the early 1980s, and has developed steadily ever since. Back then, and to a large extent even nowadays, robots had to be tediously hand programmed for every task they performed. LfD seeks to minimize, or even eliminate, this difficult step by letting users train their robot to fit their needs [29].

Comprehensive surveys of the area [1], [30], highlight that the vast majority of works in LfD follow a more engineering/machine learning approach. At the core, however, LfD is inspired by the way humans learn from experience, e.g. infants from their parents [31]. Therefore, a large number of LfD methods are motivated from concepts of psychology and biology. Going a step further, in some of these works a computational neuroscience approach is followed whereby neural modeling is adopted to address LfD. Others follow a more cognitive science approach and build conceptual models of imitation learning in animals. Surveys of this area can be found in [32], [10].

The basic idea is derived from insights of the relevant functional mechanisms underlying imitation from behavioral and neuronal data. The central questions regarding robot imitation are concerned with "what to imitate and how to solve the correspondence problem across dissimilar embodiments and task constraints. Very often these differences simply do not allow for a matching on the level of motion trajectory or path. In the goal-directed theory of imitation proposed by Bekkering and colleagues [33] imitative behavior can be considered successful whenever the goal of an action in terms of the desired outcome of the movement is reproduced. The focus on the consequences of the movement requires that the imitator understands the demonstrator's behavior as an intentional motor act directed at a specific goal (e.g. placing an object at a certain position). The "matching hypothesis" forwarded by Rizzolatti and colleagues [34] based on their discovery of the mirror system states that there are neurons, termed mirror neurons, that respond both during the observation and the execution of a similar goal-directed action (*motor simulation*).

Most relevant research projects have either focused on the perceptual side of imitation

by investigating movement systems with low complexity (e.g. mobile robots, pick-and-place industrial robots), or on the motor end by assuming the existence of all necessary perceptual information.

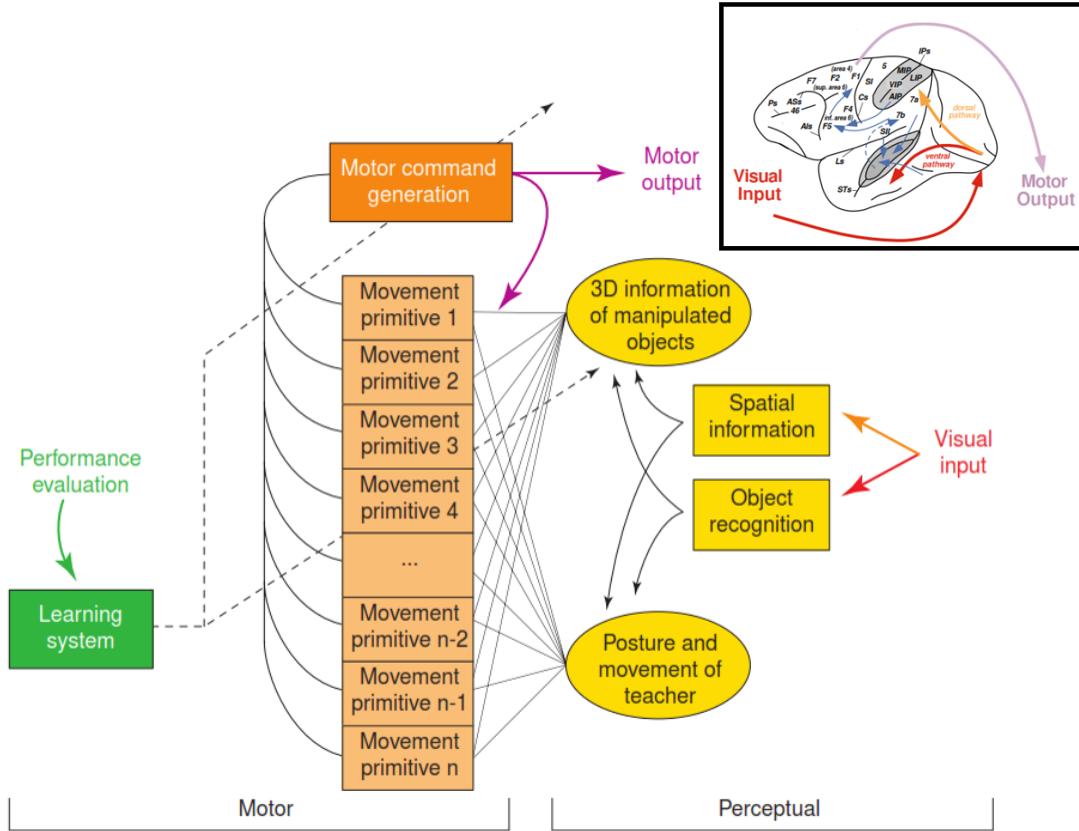


Figure 2.1: Conceptual sketch of an imitation learning system.

Accordingly, the major components of an imitation learning system, inspired by the physiological one, are shown pictorially in Fig. 2.1 [35]. Visual information that is captured by the perceptual system (the human's eyes and the camera system, respectively), is split to spatial information that refers to the "How" question, and object recognition that regards the "What" question. This is in direct analogy to the two paths that are followed by visual information in the brain.

After spatial information about the demonstrator's movement and object information has been extracted, a major issue concerns how such information should be converted into action. For this purpose, the conceptual sketch proposed by Schaal et al. [36] promotes the formulation of movement primitives, also called "movement schemas", "basis behaviors" or "units of action". Movement primitives are action-sequences that accomplish a complete goal-directed behavior. A movement primitive can be as simple as an

elementary action in the symbolic approaches to imitation. However, such low-level representations may not scale well to tasks with many degrees-of-freedom. Thus, it is desirable for a movement primitive to encode complete behaviors, like "grasping a cup", "walking" or "catching an object". Using such primitives dramatically reduces the number of parameters that need to be learned for a particular act. The drawback is that the possible movement repertoire becomes more restricted [37], [38], [39].

In summary, the perceived action of the teacher is mapped onto a set of existing primitives in an assimilation phase. Subsequently, the most appropriate primitives are adjusted by learning to improve the performance in an accommodation phase. The conceptual sketch of Fig. 2.1 highlights the process by which primitives that adequately match the observed input are ultimately translated to motor commands. If a primitive cannot be extracted successfully, then the demonstrated behavior is not satisfactorily matched, and a new primitive must be generated. This concept of movement primitives is closely related to the interpretation of mirror neurons previously refereed. Mirror neurons are deemed to code complete motor acts, that can be considered as movement primitives.

2.2 Observation - Perception of knowledge

Imitation performed through observation relies on data recorded by sensors located externally to the executing platform or internally to the robot system. Generally speaking, imitation implies a process of data selection that can be achieved in different ways. As a characteristic example, the robot learner can directly observe the teacher executions, as usually employed in the case of humanoid systems, where cameras are placed on the head, much like the human eyes.

Typically, the external sensors used to record human teacher executions are vision-based. Motion capture systems utilizing visual markers are applied for the purpose of teaching human motion and manipulation tasks [40], [41]. Vision-based methods capture the motion acts and external means of tracking human motion return precise measurement of the angular displacement of the joints. This kind of data selection has been used in various works for LfD of full body motion [42], [43]. Such methods are advantageous in that they allow the human to move freely, but require good solutions to the correspondence problem. Usually, this is accomplished by an explicit mapping between the human and robot joints, but can be quite difficult if the robot's kinematics differ substantially from the human's ones.

Another, commonly used type of observation is kinesthetic teaching. Here, the robot is physically guided through the task by the teacher. Accordingly, no explicit physical correspondence is needed, as the user demonstrates the skill on the robot's own body. It also provides a natural teaching interface to teach a skill correctly and precisely [44], [2], [45], [39], [46], [47], [48]. One possible drawback of kinesthetic teaching relates with the fact

that the human demonstrator acts with different kinematic characteristics of his/her own body. In addition, tasks that would require synchronization between multiple limbs are difficult to teach kinesthetically. The latter may be remedied by proceeding incrementally, teaching first the task for one hand (e.g. the right hand) and then for the other; unavoidably, this gives rise to a cumbersome process.

Moreover, in immersive teleoperation scenarios, as they are referred in literature, a human teacher is limited to use the robot's own sensors and effectors to perform the task. In an analogous manner to kinesthetic teaching, which limits the user to the robot's own body, immersive teleoperation essentially limits the user to the robot's perception. The teleoperation itself may be done using joysticks or other remote control devices, including haptic devices. The later has the advantage that it can allow the demonstrator to teach tasks that require precise control of forces, while joysticks would only provide kinematic information (position, speed, etc). Recently, an alternative scheme has been proposed by M. Liarokapis et al. in [49], to map humanlike trajectories to robot arm-hand systems with arbitrary kinematics, using teleoperation. The formulating constrained optimization scheme has minimal design complexity and specifications (only the robot forward kinematics (FK) are used).

Teleoperation may be advantageous in that it not only bypass the correspondence problem, but also allows for the training of robots from a distance. As the teacher no longer needs to be near the robot, it is well suited for teaching navigation and locomotion patterns. For instance, in [50] and [51], a humanoid robot is taught balancing techniques from human demonstrations. A haptic interface attached to the torso of the demonstrator was designed to transmit the perturbations induced on the robot and allow the teacher to adapt appropriately. The motion of the demonstrator was immediately re-transcribed in similar robot motion and used for training a model of motion conditioned on perceived forces.

However, as evidenced in the literature, [52], [53], [40], [54], [55], [56] teleoperation is more frequently used solely to transmit the kinematics of motion. In [57], acrobatic trajectories of a helicopter are learned by recording the pan and tilt motion of a helicopter when teleoperated by an expert pilot. In [58], a robot dog is taught to play soccer with a human guiding it via a joystick. Notwithstanding the advantages of teleoperation in LfD, a main disadvantage is that the teacher often needs training to learn to use the remote control device. Additionally, for high-degree of freedom robots, the teleoperation interface can be very complex.

As outlined above, vision-based systems (e.g. cameras, optical tracking systems) have been the most used hardware to observe the teacher demonstrations and perceive the robot's acting environment. Nevertheless, the robotic learner may be endowed with other types of sensors depending on the task to carry out. For instance, proprioceptive sensors provide information about the internal state of the robot (e.g. motor encoders), which

is very useful during kinesthetic, or teleoperated teaching [59], [60]. Auditive perception has also been used in LfD settings, where a human enhances the examples of the task by uttering words or sentences that provide more information about the current state of the demonstration [61].

2.3 Representation of knowledge

Once the demonstration phase is over, the data collected by the robot must be encoded through a model that represents the taught task in a compact way. The assumed representation depends on what the robot needs to learn from the human demonstration and on the complexity of the task.

When a robot is meant to carry out a specific set of motion acts (e.g. trajectories or paths to be followed) depending on a given set of perceptions, where position, velocity and acceleration are variables of interest, the task is usually depicted as a "low-level" representation (Fig. 2.2a). According to this representation, the robot actions are directly determined by appropriate variables, which govern motor commands to be sent to a robotic controller. Hence, an approximation of the perception-motion mapping function must be found. This function must be able to generalize, such that valid solutions are also acquired for similar states that might not have been encountered during demonstrations.

In cases that the robot is assumed to execute complex tasks, the latter are commonly split into a set of subgoals to be achieved by the robot. Accordingly, the task representation is considered as "high-level" (Fig. 2.2b), where the sub-goals are represented as action primitives pairs. Here, the learning process involves the discovery of rules linking the different state-action combinations. Rules represent actions leading from one world state to another, and are typically formulated as a set of preconditions that must hold in the world state the action applies to, and a set of postconditions or effects of the represented action. A sequence of actions is then planned using the learned rules. Unlike in LfD scenarios, planning techniques frequently rely not only on state-action demonstrations, but also on additional information of the teacher's intentions and the acting environment. Moreover, discrete encoding algorithms and graph-based models are used to represent this set of tasks. Fig. 2.2 summarizes the two levels of task representation and their characteristics.

2.3.1 Policy Formulation

As mentioned in the introduction, the main issue in LfD techniques is the formulation of a learning process that accomplishes a mapping between the real world state and the robot's one. To this end, LfD algorithms utilize a learned dataset of demonstrations to derive a policy that represents the demonstrated behavior.

In this context, LfD can be studied as an instance of Supervised Learning. In the latter,

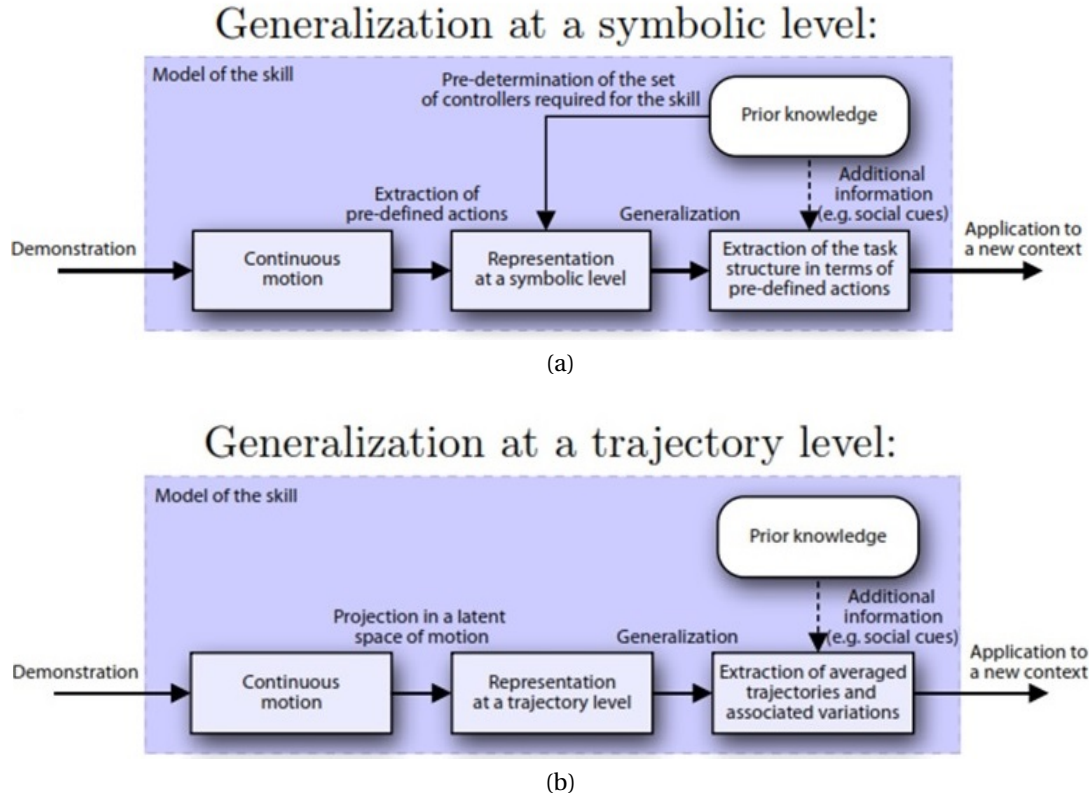


Figure 2.2: The two levels of knowledge representation as proposed by Billard and colleagues in [62].

an agent learns an approximation to the objective function that has been produced by the selected data. In LfD, the training dataset is collected by a sequence of demonstrated examples of the task, executed by a teacher. In practice, as also detailed in [1], an LfD problem should be formulated as follows. Given a set of states, S , and actions A , a policy is defined as a mapping between the observed states and the actions, and can be described by a probabilistic transition function $T(A | S)$. To this end, the policy $p : A | S$ learns actions based on the current observation of the world state.

Policy formulation methods can be characterised as postulated by Argaal et al. [1]. Accordingly, three levels of an LfD framework are depicted. The first level, the phase of observation, is composed by the selected dataset, the second represents the policy formulation and the third the policy learning phase.

With respect to the policy formulation, three basic approaches have emerged [1]:

- *Mapping functions* [63]: Demonstration data is used to directly approximate the underlying function mapping from the world state observations to actions. The goal

of this type of algorithms is to reproduce the underlying teacher policy, which is unknown, and to generalize over the set of available training examples such that valid solutions are also acquired for similar states that may not have been encountered during demonstration. Essentially, this is the most common approach of policy formulation that appears in LfD literature.

- *System model* [64]: Demonstration data is used to determine a model of the world dynamics, and possibly a reward function $R(\mathbf{x})$. A policy is then derived using this information. This approach is typically formulated within the realm of Reinforcement Learning (RL). Demonstration data, and any additional autonomous exploration the robot may perform, generally determine the transition function. To derive a policy from this transition model, a reward function $R(\mathbf{x})$, which associates reward value r with world state s , is either learned from demonstrations or defined by the user.
- *Plans* [31], [65], [66]: Demonstration data, and often additional user intention information, is employed to learn rules that associate a set of pre- and post-conditions with each action, and possibly a sparsified state dynamics model. A sequence of actions is then planned using this information. The planning framework represents the policy as a sequence of actions that lead from the initial state to the final goal state. Actions are often defined in terms of pre-conditions, the state that must be established before the action can be performed, and post-conditions, the state resulting from the action's execution. Unlike other LfD approaches, planning techniques frequently rely not only on state-action demonstrations, but also on additional information in the form of annotations or intentions from the teacher. Demonstration-based algorithms differ in how the rules associating pre- and post-conditions with actions are learned, and whether additional information is provided by the teacher.

A pictorial depiction of the policy derivation approaches, as outlined above, is shown in Fig. 2.3 [1].

2.4 Encoding of Knowledge

Learning a policy may involve simple learning of an approximation to the state-action mapping (objective function), or learning a model of the world dynamics and deriving an appropriate policy (system model). The Machine Learning literature has contributed a good number of algorithms that have already been adopted by LfD methods [7], [64], [55]. Given an appropriate problem mapping, Machine Learning methods are employed to encode the skill so that the robot can successfully reproduce it. The selection process of the model depends on the type of the imitation task and the level of representation of the teacher demonstrations. Learning techniques based on dynamical system models and

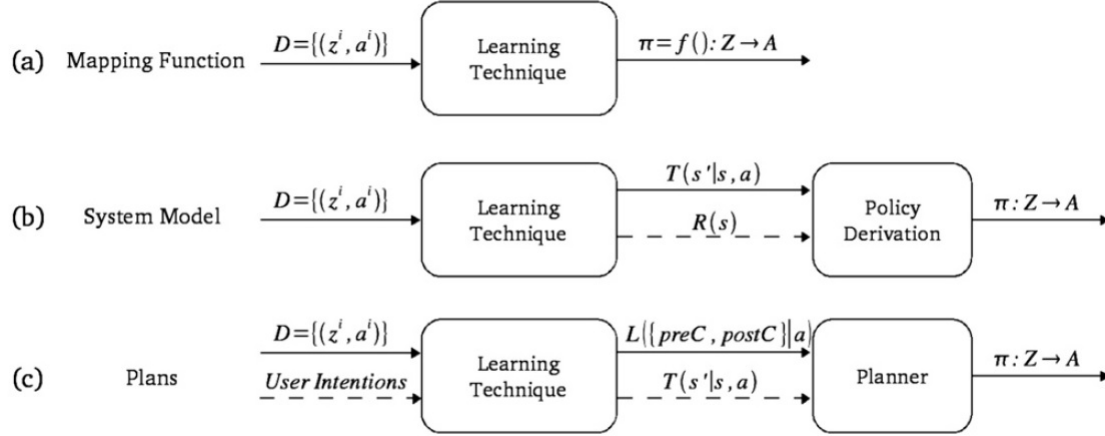


Figure 2.3: Policy derivation as proposed in [1], using the generalization approach for determining (a) an approximation to the state given the action mapping function, (b) a dynamic model of the system, and (c) a plan of sequenced actions.

stochastic approaches stand out over more ad-hoc algorithms, because they provide a more general structure to encode various skills using the same basic framework. The desirable across all of the learning techniques is minimal parameter tuning and fast learning, while requiring few training examples.

2.4.1 Dynamical Systems Models

In the process of learning how to imitate human-like motion acts, it is common to represent movements in kinematic coordinates, e.g. angle joint space of a robot or Cartesian coordinates of an end-effector, etc. In practice only kinematic variables are observable by a vision-based set-up and are useful in LfD methods. Nevertheless, a kinematic motion plan, as a change of position as a function of state or velocity, can be described in motor commands by a controller. Dynamical Systems (DS) provide a powerful tool for robust control of point to point robot motions from a small set of demonstrations. They ensure high precision in reaching a desired target, yet can be easily modulated to generate new motions in slightly different environments. Relevant works highlighted the successful learning of discrete (i.e. point-to-point) robot motions either through time-dependent or time-independent DS [67], [68]. While these works address the fundamental concern when learning DS, i.e. stability, they can only be used for generating motions with zero velocity at the target. More recently, in [69], S.M. Khansari-Zadeh and A. Billard studied DS for discrete robot movements that model motions with both zero and non-zero velocities.

In DS approaches, the control policy to steer a robotic platform is modelled by a first

or higher order DS. When controlled through a DS, robot motion unfolds in time with no need to re-plan [70]. An estimate of the DS can be built from a few demonstrations of the task at hand. The estimated DS captures the invariant features in the user demonstrations, and can generate motions that resemble the learned ones [45], [71], [39], [40], [72], [69], [73]. Each DS model codes a specific motion (behavior), and is called a movement primitive (also known as motor primitive). They can be seen as building blocks that can be used to generate more complex or new motions through sequencing or superimposition of the primitives. This modularity of DS-based movement primitives is essential as it allows controlling a wide repertoire of movements from a (small) set of basic motions.

Regarding learning methods based on DS, Ijspeert et al. [3] proposed nonlinear differential equations to form control policies in trajectory formation. In the context of robot imitation learning, Schaal et. al. [54] were among the first groups to suggest the idea of using a programmable DS formulation that can be adjusted to different tasks. This idea was further extended in [30], where a method is proposed, called Dynamic Movement Primitives (DMPs), to build an estimation of nonlinear DS via Imitation Learning. DMPs can be used to generate discrete and rhythmic movements. The latter has been used for different robotics applications such as walking [38], drumming [74], pouring [75], obstacle avoidance [76], [77], and catching objects in flight [78]. In DMP approaches, the dynamical system represents a complete flow field instead of a single-trajectory. In other words, they encode a whole attractor landscape in which the desired trajectory is produced. Such flow fields can be constructed from demonstrations, which can automatically correct external perturbations and guarantee convergence to a goal state. DMPs have also been extensively used in imitation of reaching movements [65], for encoding rest-to-rest motions in articulated mobile robots, and also employed in manipulation tasks where the behavior of the primitives is influenced by perceptual cues.

The main idea behind DMPs regards the definition of a dynamical system (e.g. a set of first order systems) to describe the demonstrated motion. Formally speaking, for a state variable $\xi \in \mathbb{R}^d$ that defines the state of a robotic system, its evolution in time is governed by an autonomous (time-independent) DS according to:

$$\dot{\xi} = F(\xi, \theta) \quad (2.1)$$

where θ is the set of parameters defining F . In turn, the DS should be deformed in a manner that will facilitate learning of the system's parameters. In [51], for instance, Eq. 2.1 is decomposed into two terms, and formulated as a product of a target field $\mathbb{E}(\xi, \theta_E)$ and a strength factor $v(\xi, \theta_u)$:

$$\dot{\xi} = v(\xi, \theta_u) \times \mathbb{E}(\xi, \theta_E) \quad (2.2)$$

The structure of Eq. 2.2 is in accordance to principles from physics, where the motion of a particle in space can be defined as the product of an appropriate field (e.g. gravity, elec-

trical field, etc.) times a scalar (e.g. mass, electric charge, etc.). The former is a property that describes the space that surrounds a particle, and the latter defines the characteristics of the particle. Similarly, in Eq. 2.2, the target field describes the form of a motion and the strength factor determines its intensity. The DS parameters θ_u and θ_E can be learned from the teacher's demonstrations. An important issue in such approaches lies in the fact that these parameters should be estimated, such that they (a) ensure the accomplishment of the task starting from any point in space (i.e. global convergence), and (b) generate robot motions that follow the human demonstrations accurately.

Such an approach has been the basis for further reformulations and modifications of DMP. For instance, Hoffmann et al. [79] stated that the original DMP could be expressed with a mechanical analogy by defining the basis force components used in DMP to modulate the movement as virtual damped springs, thus moving the learning problem to the estimation of virtual equilibrium points instead of estimating forces. On the other hand, Pastor et al. [67] proposed a DMP framework where sensory information captured in the demonstration phase may modify the desired trajectory in an online manner, so that the measured sensory experience remains close to the expected one. This idea shares similarities with the perceptual coupling for DMP proposed by Kober et al. [80]. In this work the original formulation was modified by including a coupling with external variables, most of them considered as perceptual cues.

As S.M. Khansari-Zadeh et al. pointed out [81], the main advantage of using DS-based formulation can be summarized as: "Modeling movements with DS allows having robotic systems that have inherent adaptivity to changes in a dynamic environment, and that can swiftly adopt a new path to reach the target". This advantage is the direct outcome of having a unified planning and execution unit.

Summarizing in a schematic graph of control flow of a dynamical system model in LfD, Fig. 2.4 [3] depicts a motion in kinematic coordinates (i.e. the Cartesian or robot's joint space), viewing the process as a low-level controller that converts kinematic variables into motor commands (e.g. force or torque). The whole architecture can be decomposed into two loops. The inner loop consists of a controller generating the required commands to follow the desired motion and a system block to model the dynamics of the robot. The outer loop specifies the next desired position and velocity of the motion with respect to the current state of the robot. An inverse kinematics block may also be considered in the outer loop to transfer the desired trajectory from the Cartesian to the joint space. The derived controller, driven by a DS, is robust to perturbations as it considers all possible solutions to reach a single target.

2.4.2 Stochastic Models

Stochastic expressions are frequently adopted in LfD due to the following:

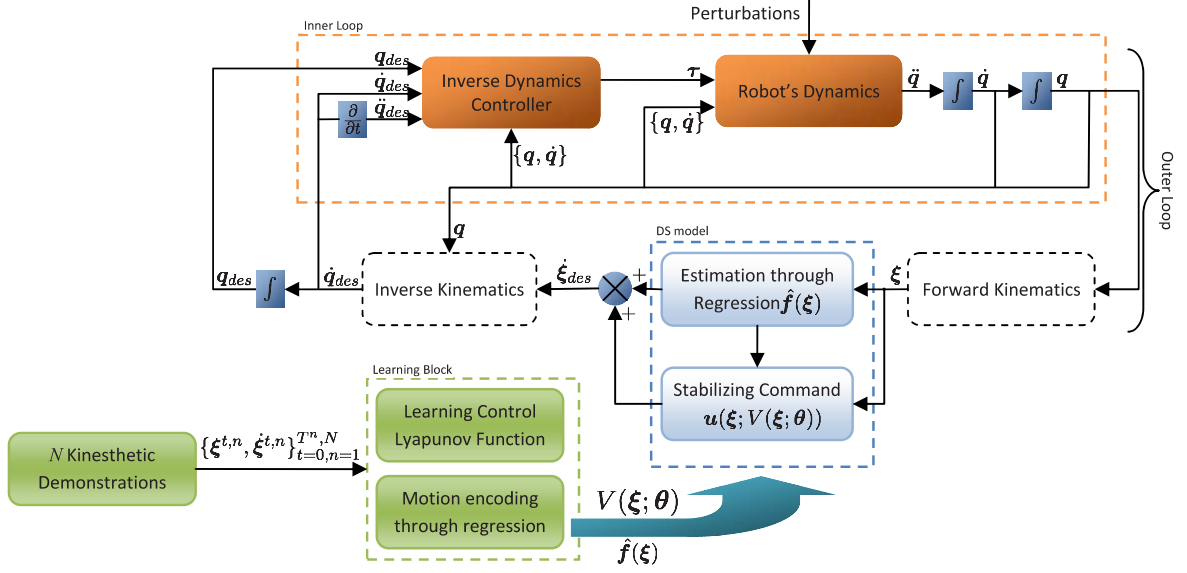


Figure 2.4: Dynamical System Framework [3] commonly used in LfD.

- On-line learning can be accomplished using simple cumulative frequency calculation from interaction experience.
- The confidence of a decision can be expressed as a probability value; the latter may also control robot interaction.
- A stochastic process may cancel out the influence of noise, such as user-errors in a demonstration.

Bayesian Networks have been adopted in various works in order to build such a stochastic model [8], [82], [83], [84]. A Bayesian Network is a reasoning model in which the relation between cause and effect of multiple phenomena is expressed as a probability value. Gaussian Mixture Models (GMM) also constitute the basis of several LfD frameworks performing successfully in a variety of scenarios [85], [86], [78], [87], [88]. Broadly speaking, GMM can be considered as a statistical encoding tool where a mixture of experts (i.e. normal distributions acting as the states of the model) represent the data, allowing a localized characterization of the different parts of the demonstrated task. Calinon et al. [41] used this mixture modeling to teach simple manipulation tasks to a humanoid robot. However, one of the main drawbacks of GMM is the strong assumption of having aligned data streams, that is, fixed time length demonstrations. Therefore, a pre-processing phase over the training datapoints is needed to obtain such data. Among the solutions, one can find Dynamic Time Warping (DTW) [4], [5] and Hidden Markov Models (HMM).

HMM is a powerful method to encode time-series data and may be also considered as an extension of the original GMM, where the temporal evolution of the data is encoded

through the evolution of a hidden state. Such temporal information is encapsulated by transition probabilities for every pair of states. Thus, HMMs can be used to encode temporal and spatial variations of complex signals, and to model, recognize and reproduce various types of human demonstrations. Recently, some extensions of the standard HMM formulation have also been applied to learn tasks from imitation. Kruger et al. [89] used a parametric version of HMM to learn reaching movements, where the model states linearly depend on a given parameter of the task, i.e. the location of the object to be grasped. In [90], the authors propose to encode time and space constraints of a trajectory following task using an explicit-duration HMM, which was shown to provide good results when the robot faced strong perturbations during the execution of the task.

In the context of LfD, a stochastic process is usually formulated as follows. A dataset $\xi = \xi_j^N$ is defined by N observations $\xi_j \in \mathbb{R}^D$ of sensory data over time (e.g. joint angle trajectories, hand paths), where each datapoint ξ consists of a temporal value $\xi_t \in \mathbb{R}$ and a spatial vector $\xi_s \in \mathbb{R}^{D-1}$. The dataset ξ is modeled by a **Gaussian Mixture Model (GMM)** of K components, defined by the probability density function of:

$$P(\xi_j) = \sum_{k=1}^K p_k \mathbb{N}(\xi_j | \mu_k, \sigma_k), \quad (2.3)$$

where p_k are prior probabilities and $\mathbb{N}(\xi_j | \mu_k, \sigma_k)$ are Gaussian distributions defined by mean vectors μ_k and covariance matrices σ_k , whose temporal and spatial components can be represented separately as:

$$\begin{aligned} \mu_k &= \{\mu_{t,k}, \mu_{s,k}\} \\ \sigma_k &= \begin{pmatrix} \sigma_{tt,k} & \sigma_{ts,k} \\ \sigma_{st,k} & \sigma_{ss,k} \end{pmatrix} \end{aligned} \quad (2.4)$$

For each component k , the expected distribution of ξ_s given the temporal value ξ_t is defined by as:

$$\begin{aligned} P(\xi_s | \xi_t, k) &= \mathbb{N}(\xi_s | \hat{\xi}_{s,k}, \hat{\sigma}_{ss,k}) \\ \hat{\xi}_{s,k} &= \mu_{s,k} + \sigma_{st,k} \sigma_{tt,k}^{-1} (\xi_t - \mu_{t,k}) \\ \hat{\sigma}_{ss,k} &= \sigma_{ss,k} - \sigma_{st,k} \sigma_{tt,k}^{-1} \sigma_{ts,k} \end{aligned} \quad (2.5)$$

By considering the complete GMM, the expected distribution is formulated as:

$$P(\xi_s | \xi_t) = \sum_{k=1}^K \beta_k \mathbb{N}(\xi_s | \hat{\xi}_{s,k}) \quad (2.6)$$

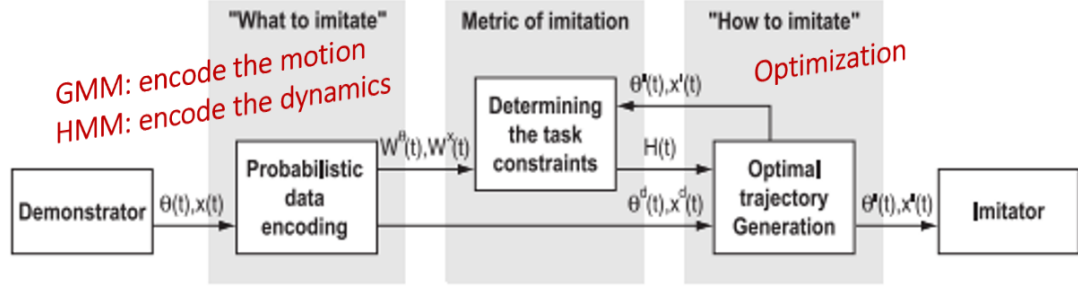


Figure 2.5: Stochastic processes in LfD.

where $\beta_k = P(k \mid \xi_t)$ is the probability of the component k that is responsible for ξ_t . By using the linear transformation properties of Gaussian distributions, an estimation of the conditional expectation of $(\xi_s \mid \xi_t)$ is computed by $P(\xi_s \mid \xi_t) \sim \mathcal{N}(\hat{\xi}_s, \hat{\sigma}_{ss})$, where the parameters of the Gaussian distribution are obtained as previously. By evaluating $(\hat{\xi}_s, \hat{\sigma}_{ss})$ at different time steps, we obtain a generalized form of the motions $\hat{\xi} = \{\xi_t, \hat{\xi}_s\}$ and associated covariance matrices describing the applied constraints.

As a result, GMM can offer a stochastic representation of the data, allowing a localized characterization of the different parts of the demonstrated task. Alternatively, HMMs have also been proposed in the literature for encoding time-series data. In this context, HMMs can be viewed as an extension of the original GMMs, where the temporal evolution of the data is encoded through the evolution of a hidden state. Such temporal information is encapsulated by transition probabilities for every pair of states. Thus, HMMs can be used to encode temporal and spatial variations of complex signals, and to model, recognize and reproduce various types of human demonstrations. In LfD, HMMs have been used for teaching collaborative lifting tasks to a humanoid robot [91], [92], for learning and reproduction of a bi-manual task and tennis table strokes [93], [4], [81], and as a basis for a hierarchical incremental learning of full body motion [94], [78].

Fig. 2.5 [90] presents a general block diagram of stochastic LfD processes.

2.4.3 Latent Representations

Feature selection methods usually operate by keeping only variables with high information content and discarding others in the original data-space. This approach is useful when it is essential to retain the original data provided by some of the inputs of the problem. In other words, the original features may convey information that can be further interpreted and readily used, than if they were projected on a different space.

Dimensionality reduction consists a standard method to deal with either a large number of variables, or with highly correlated parameters. Several techniques have been pro-

posed to achieve such a transformation. One standard linear transformation for dimensionality reduction is *Principal Component Analysis* (PCA) [95]. This transformation is derived from eigenvectors corresponding to the largest eigenvalues of the covariance matrix of the data. The method seeks to optimally represent the data in terms of minimal mean-square-error (MSE) between the representation and the original data.

PCA has been successfully applied in kinesthetic LfD for building a latent space onto which spatio-temporal trajectories are projected, to find an optimal representation for a given task [83], [8], [82]. This allowed to eliminate redundancies of the original training dataset while keeping the relevant information of the demonstrations in a subset of new variables constructed from a linear combination of the original inputs. To this end, given an observation data set $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$, $x_n \in \mathbb{R}^D$, such methods search for an L -dimensional representation, where usually $L \ll D$. The derived representation, $\mathbf{Y} = \{y_1, y_2, \dots, y_n\}$, $y_n \in \mathbb{R}^L$, termed as latent variables, is obtained by an orthogonal linear transformation of the original data Eq. 2.7, which is defined by the matrix $\mathbf{A} = \{v_1, v_2, \dots, v_D\}$, with v_i being the eigenvectors of the covariance matrix of \mathbf{X} with associated eigenvalues (linear PCA). D is the minimal number of eigenvectors used to obtain a satisfying representation of the data.

$$\mathbf{x}_n = \mathbf{W}\mathbf{y}_n + \eta_n \quad (2.7)$$

where the matrix $\mathbf{W} \in \mathbb{R}^{D \times L}$ specifies the linear relationship between the data space and the latent space (described below), the noise values η_n follow a Gaussian distribution $\mathcal{N}(\eta_n | \mathbf{0}, \beta^{-1}\mathbf{I})$ with unit covariance, and β denotes the inverse variance. To this end, the conditional probability of a data point \mathbf{x}_n will also follow a Gaussian distribution as shown in Eq. 2.8:

$$p(\mathbf{x}_n | \mathbf{y}_n, \mathbf{W}, \beta) = \mathcal{N}(\mathbf{x}_n | \mathbf{W}\mathbf{y}_n, \beta^{-1}\mathbf{I}) \quad (2.8)$$

In the need of expanding the problem in the non-linear case, a non-linear kernel function is used as a covariance matrix instead of the linear one used in PCA. Indeed, in [96], Shon et al. learned a joint latent variable space with Gaussian processes, having applied a generalization of PCA, called Gaussian Latent Variable Model [97], using the Radial Basis Function (RBF) as a covariance matrix. One sample latent space, captured from [96], its corresponding actor's and robot's states respectively, is illustrated in Fig. 2.6.

Another similar algorithm that makes use of second-order statistical information, namely covariances, is the linear discriminant analysis (LDA) [53]. This technique applied to classification problems finds a transformation from the eigenvectors of a matrix that captures the compactness of each class and the separation of the class means. Independent component analysis (ICA) is another tool to find interesting projections of the data by maximizing the divergence to a Gaussian density function in order to decorrelate and denoise the data, as well as to reduce the dimensionality of the dataset to make this one tractable. This criterion corresponds to finding a projection of data that looks maximally clustered [93].

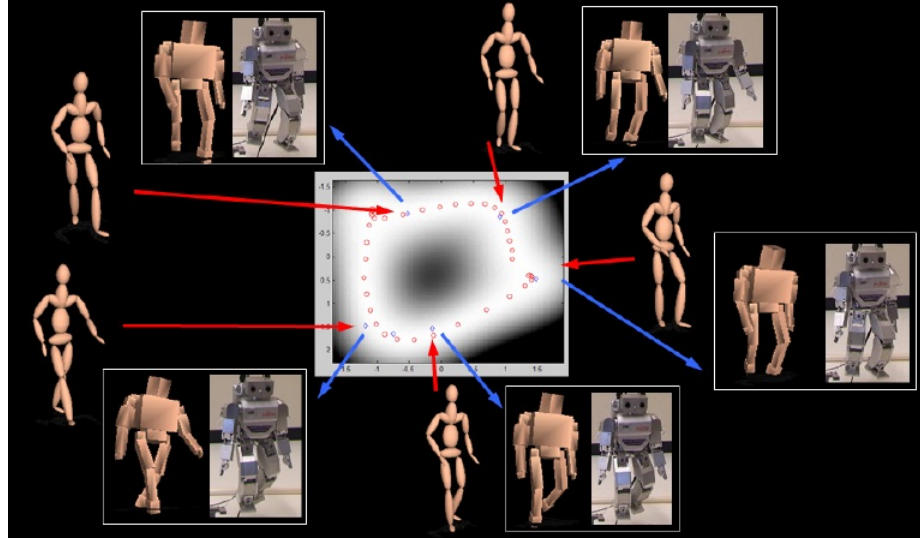


Figure 2.6: Imitation from motion capture as described in [96].

2.5 How to imitate: Learning a policy-mapping

Once the task model has been defined, it is essential that an appropriate learning strategy is specified. In the case of dynamical systems, the task is often time-dependent, thus at the reproduction phase the learned DMP is used to reproduce the task using temporal variables. The smoothness of the reproduction depends on the type of non-linear equations used for encoding the demonstrations. Similarly, when encoding through probabilistic models, retrieving smooth trajectories is a demanding issue too. To this end, averaging approaches have been employed where generalized movements are retrieved by averaging over a large number of trajectories previously generated from the trained model. Such approaches are considered time consuming and computationally expensive, and still cannot guarantee the smoothness of the end result.

Interpolation-based approaches were also proposed to obtain a reproduction from HMM-based encoding, where the mean of the Gaussian distributions is considered to obtain series of key points for interpolating them [37], [41]. The main drawback of such approaches is the fact that the covariance information is ignored. Calinon et al. [4] introduced the use of Gaussian Mixture Regression to retrieve a time-based trajectory from a set of demonstrations encoded by a GMM. GMR also provides smooth generalized trajectories with associated covariance matrices describing the variations and correlations across different variables, considering the covariance data encapsulated by the Gaussian states.

In contrast to the above approaches, other regression-based methods have been also used to encode and reproduce robotic skills, mostly based on trajectory tracks. A com-

monly used regression technique is locally weighted regression (LWR) [5], [48], [40], a memory-based algorithm that combines the simplicity of linear least squares along with a weighting mechanism to learn non-linear functions. This approach was the basis for two further extensions, namely receptive field weighted regression (RFWR) [98], and locally weighted projection regression (LWPR) [78]. The former dealt with the problem of moving from a batch process to an incremental learning strategy, but suffers from the curse of dimensionality. This drawback was remedied by LWPR, which was shown to operate efficiently in high dimensional spaces.

On the other hand, Gaussian processes (GP) have also been applied to LfD tasks. As stated above, Grimes et al. [83] proposed to use GP for nonparametric forward model learning in whole-body motions of a humanoid robot. Nonetheless, its main disadvantage is considered to be its high computational cost. One possible solution to this problem is based on sparse GP, where only a subset of the latent variables are treated exactly, and the remaining variables are given some approximate, but computationally cheaper treatment. Grollman and Jenkins compared this approach with LWPR in the context of LfD, where both techniques provided good function approximation capabilities [99].

However, regarding hard memory and timing constraints, sparse GP showed to be more suitable for real-time interaction. In contrast, Nguyen-Tuong et al. [100], solved the high computational cost problem by partitioning the training data into local regions and learning an independent GP model for each region, similarly to how LWPR works. Schneider and Ertel [101] also proposed a local approximation, where the training inputs were assigned to the local model that best fits and then an individual GP on each of these models was trained.

Furthermore, in [68], [39], Kronander and Khansari-Zadeh et al. proposed an alternative method called Stable Estimator of Dynamical Systems (SEDS) to ensure globally asymptotical stability. Specifically, SEDS minimize the model estimation error given the demonstrated data while ensuring that the learned autonomous DS is globally stable at the target. They studied the outcome of two objective functions SEDS-MSE and SEDS-Likelihood for this optimization problem. The models derived from the optimization of both objective functions benefit from the inherent characteristics of autonomous DS, i.e. online adaptation to both temporal and spatial perturbation. Accordingly, each objective function has its own advantages and disadvantages. As shown in their experiments, log-likelihood SEDS is more accurate and smoother than MSE. Furthermore, the MSE cost function is slightly more time consuming since it requires computing GMR at each iteration for all training data points. However, the MSE objective function requires fewer parameters than the likelihood one which may make the algorithm faster in higher dimensions or when higher number of components is used.

Additionally, Khansari-Zadeh et al. [102] presented the Binary Merging (BM) algorithm, that tackled the problem of estimating (identifying) an unknown non-linear dynamical

system from a few demonstrations while ensuring its global stability. However, as the author quoted in [45], though BM provided sufficient conditions to make DS locally stable, it had still relied on determining numerically the stability region and had had a limited region of applicability.

In these approaches, robot discrete motions are formulated as a control law driven by autonomous Dynamical Systems (DS). For a state variable $\xi \in \mathbb{R}^d$ that can be used to unambiguously define a discrete motion of a robotic system (e.g. ξ could be the robot's joint angles, position and/or orientation of an arm's end-effector in the operational space, etc), the system is defined as:

$$\dot{\xi} = f(\xi), f \in \mathbb{R}^d \quad (2.9)$$

where $f(\xi)$ is a continuous function that codes a specific behavior, such as reaching for a cup or swinging a golf club, etc. Starting in an initial configuration ξ^0 , the robot motion ξ^t , $t \in [0, \infty)$ is given by integrating from Eq. 2.9.

When modeling robot discrete motions with DS, the problem is defined as a problem of estimating the optimal system's parameters that can ensure global stability. It is noted that the function $\hat{\xi}$ is globally asymptotically stable at the target ξ^* if $f(\xi^*) = 0$ and $\forall \xi^0 \in \mathbb{R}^d$, the generated motion converges asymptotically to ξ^* :

$$\lim_{t \rightarrow \infty} \xi^t = \xi^*, \forall \xi^0 \in \mathbb{R}^d \quad (2.10)$$

Which means that $\hat{\xi}$ is locally asymptotically stable if it converges to ξ^* only when ξ^0 is contained within a subspace.

In a recent work, S.M. Khansari-Zadeh and A. Billard [81] approached LfD by defining a Control Lyapunov Function (CLF) control scheme to ensure global asymptotic stability of nonlinear DS. Given a set of demonstrations of a task, the proposed framework follows three steps:

1. Learn a valid Lyapunov function from the demonstrations by solving a constrained optimization problem,
2. Use a state-of-the-art regression technique to model an (unstable) estimate of the motion from the demonstrations,
3. Use a valid Lyapunov function (1) to ensure stability of a regression model(2) during the task execution via solving a constrained convex optimization problem.

The proposed approach allows learning a larger set of robot motions compared to existing methods that are based on quadratic Lyapunov functions. Additionally, by using the CLF formalism, the problem of ensuring stability of DS motions becomes independent

from the choice of regression method. Hence, it allows the user to adopt the most appropriate technique based on the requirements of the task at hand without compromising stability.

Researchers in the field of LfD generally agree that a widely adopted assumption to represent complex skills and non-linear motions is to decompose them into smaller units of action, and weighted combination of linear systems. Examples of models that can be reformulated in this way are the GMR based approaches, and methods whose core is the DMP [38], [74]. In most of the papers associating with LfD, hybrid techniques are mostly applied for formulating an LfD model and learning it, succeeding to combine the advantages of each [102], [35].

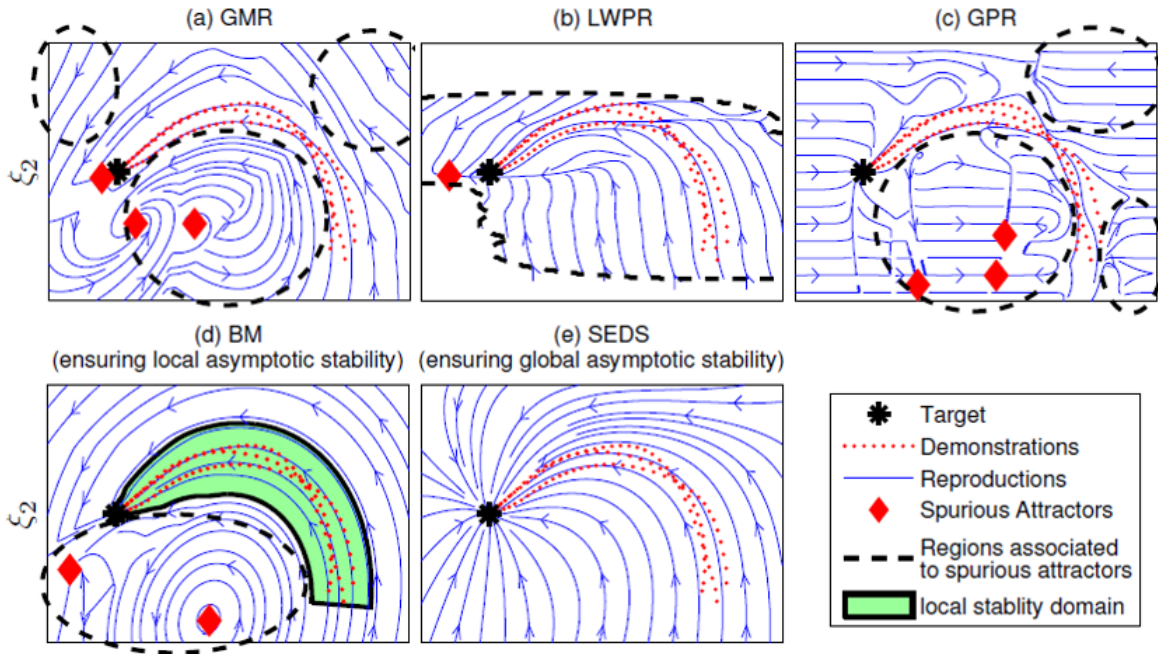


Figure 2.7: An example of two-dimensional dynamics learned from three demonstrations comparing five different methods: GMR, LWPR, GPR, BM, and SEDS, captured by [45].

The performance of the above methods in a simple estimation task was examined quantitatively by Khansari-Zadeh et al. [45]. Fig. 2.7 illustrates a case of an unstable estimation of a non-linear DS using the above methods for learning a two dimensional motion. Fig. 2.7 depicts the stability analysis of the dynamics learned with:

1. GMR, where the trajectories converge to a spurious attractor just next to the target. In the rest space, they either converge to other spurious attractors far from the target or completely diverge from it.

2. LWPR, where all trajectories inside the black boundaries converge to a spurious attractor, whereas outside them, the velocity is always zero (a region of spurious attractors) and the motion stops.
3. In the case of GPR trajectories converge to the target in a narrow area close to demonstrations, and they are attracted to local minima outside that region.
4. BM ensures local asymptotic stability at the target, hence the model can only be applied in a region close to demonstrations.
5. SEDS succeed to ensure global asymptotic stability. and all trajectories converge to the target. This ensures that the task can be successfully accomplished starting from any point in the operational space without any need to re-index or re-scale.

In all above cases, regions of attractions are usually very close to demonstrations and thus should be carefully avoided. However, the critical concern is that there is not a generic theoretical solution to determine beforehand whether a trajectory will lead to a spurious attractor, to infinity, or to the desired attractor. Thus, it is necessary to conduct numerical stability analysis to locate the region of attraction of the desired target which may not exist, or be very narrow.

2.5.1 Metrics of Imitation

LfD is a relatively young but rapidly growing field. As highlighted by this report, a wide variety of approaches address the challenges presented by this learning method. However, to date there exists little direct comparison between algorithms [103]. One reason for this absence of direct comparisons is that most approaches are tested using only a single domain and robotic platform. As a result, such techniques are often customized to that particular domain and do not present a general solution to a wide class of problems. Additionally, the field of LfD research currently lacks a standard set of evaluation metrics, further complicating performance comparisons across algorithms and domains. Improved methods for evaluation and comparison constitute a fundamentally important area for future work. Existing evaluation methods include several proposed for specific areas of LfD, and potentially some from the Human-Robot Interaction (HRI) community [104], [105], [106]. Formulation of evaluation criteria would help to drive research and the development of widely applicable general-purpose learning techniques.

As noted in [4], metrics of imitation can be categorized into subgroups with respect to the imitation task. Defining a generalized imitation metric as a function of the variable that is reproduced by the robot, may give rise to:

$$J = \|\xi_a^\theta - \xi_{rob}^\theta\|_{\mathbf{W}^\theta}^2 + \|\xi_a^x - \xi_{rob}^x\|_{\mathbf{W}^x}^2 + \|\xi_a^y - \xi_{rob}^y\|_{\mathbf{W}^y}^2 \quad (2.11)$$

where $\xi_a^{\theta,x,y}$ is the actor's studied variable each time, e.g. end effector's trajectory, joint angles configuration etc. Additionally, $\xi_{rob}^{\theta,x,y}$ is the corresponding robot's one. The vectors $\mathbf{W}^{\theta,x,y}$ are the computed weights. Depending on the task, these variables bear different importance, and the different levels of relevance are extracted by observation of the task produced by a human expert. In a goal-directed framework, these three variables have also different levels of relevance. If an object is manipulated, the end-effector's variable reveals the highest importance. If there is no object in the scene, then usually the joint angles are of higher importance. Moreover, in case of grasping, moving or dancing tasks, the grasping or walking configuration is highlighted in relation to the others. Accordingly, as can be seen in the literature, the quantification of the imitation success is a critical element of LfD process.

2.6 Robot Reproduction

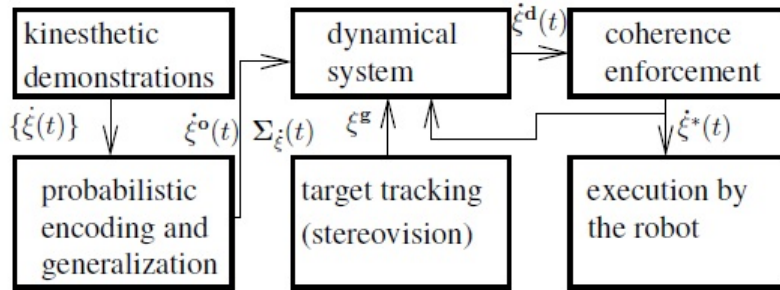
2.6.1 Applications - Experimental Scenarios

Works dealing with LfD have been carried out in different set-ups, where both sensory information and demonstration samples differ for each application. Most of the efforts have focused on teaching a given scenario or a specific trajectory to be followed by the robot. Such trajectories may correspond to a set of desired variable profiles of the robot's space, such as position, velocity, joint angles, etc. This low-level learning has been successfully applied mostly to manipulation tasks, grasping skills, gesture reproduction and whole body motion pattern imitation. Furthermore, the development of compliant robots brings new possibilities in imitation learning, by extending the skill transfer problem towards tasks involving force control, and towards reactive systems able to cope with various sources of perturbation coming from the interaction with the user and the environment.

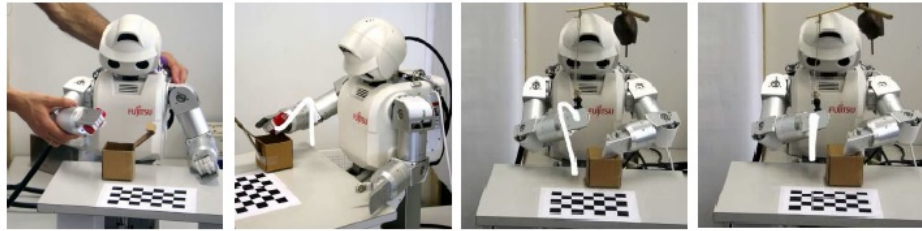
In this section, we present selected important applications and notable experimental setups that have arisen from LfD works. A large amount of research works in LfD have focused on endowing robots with manipulation skills to allow them to interact with objects populating their environments. Assembly setups, such as the well-known peg-in-hole task, have served as the appropriate bench to study LfD issues. Nevertheless, over the years research on LfD has significantly strengthen and applied to more demanding scenarios and high level tasks. As the most low level tasks and the simple manipulation scenarios have been covered in the published book of A. Billard [62], in the rest of this section we will present some more recent and notable developments in the field.

In [47], Hersch et al. proposed an algorithmic procedure that enables a robot to perform constrained reaching tasks in a dynamic environment. Their framework is grounded by an hybrid LfD model consisting of constant movement feature extraction in conjunction with a dynamical system (Fig. 2.8a). This allows the system to reproduce a task, while

adapting to new external conditions. The task of putting a ball into a box, regardless of the environment conditions, was kinesthetically taught, as illustrated in Fig. 2.8b.



(a)



(b)

Figure 2.8: (a) LfD framework proposed in [47] [71]. (b) Hoap humanoid robot putting an object into a box, kinesthetically driven by a human [71].

Sauser et al. [55] introduced an approach for grasp adaptation which learns a statistical model to adapt hand posture based solely on the perceived contact between the object and fingers. Using a multi-step learning procedure, the model dataset is built by first demonstrating an initial hand posture, which is then physically corrected by a human teacher pressing on the fingertips, exploiting compliance in the robot hand. To this end the learner achieved to replay the resulting sequence of the hand postures, to generate a dataset of posture-contact pairs that are not influenced by the touch of the teacher. The key feature of this work was that the learned model may be further refined by repeating the correction-replay steps. Alternatively, the model may be reused in the development of new models, characterized by the contact signatures of a different object. The task that was evaluated by i-cub robot, is illustrated in Fig. 2.9.

Furthermore, Billard's group employed robots with sporting skills of playing golf or tennis, previously driven by a human's expertise. Obtained results reported in [3], [71] are illustrated in Fig. 2.10a and Fig. 2.10b, respectively. Playing mini golf problem is tackled as a hitting object task. To this end, having learned an adaptable hitting motion that can be used to hit with different speed and direction, the robot needs to learn what speed

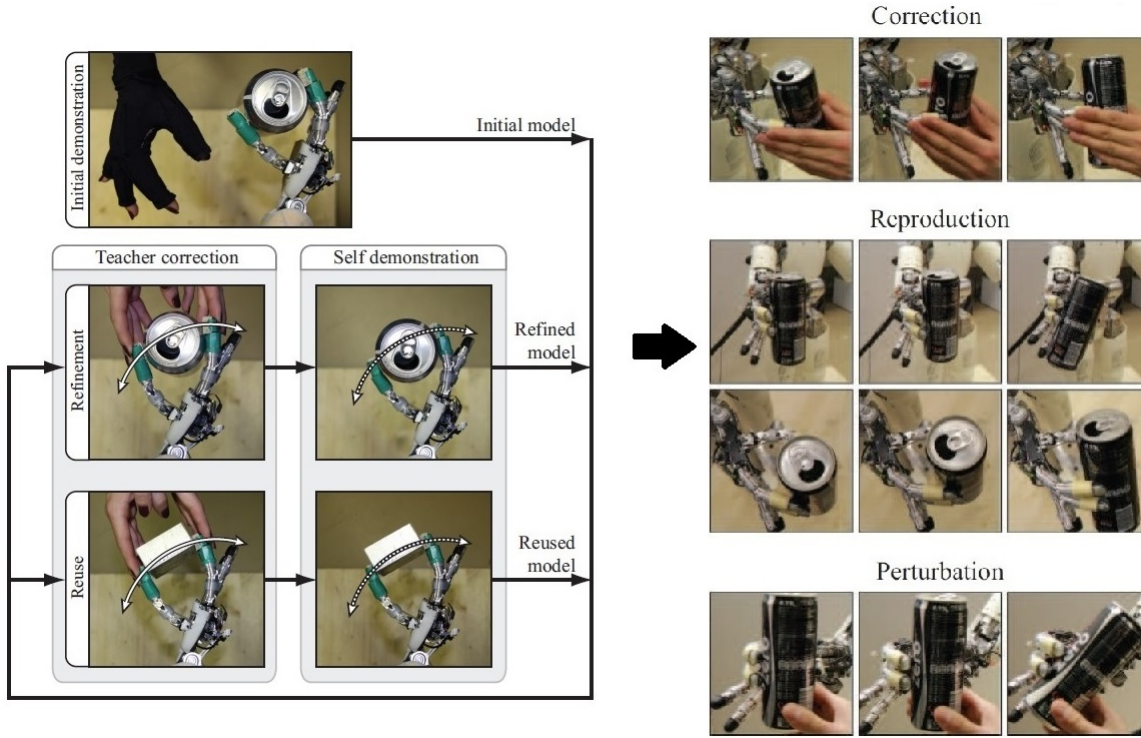
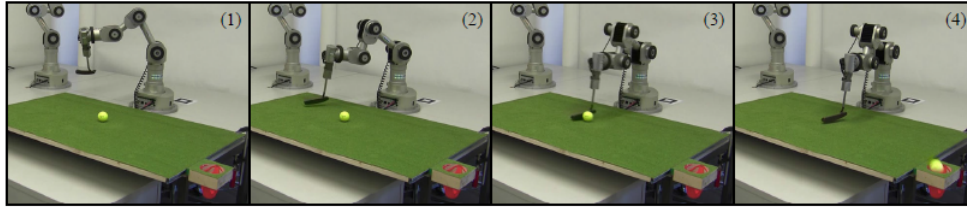


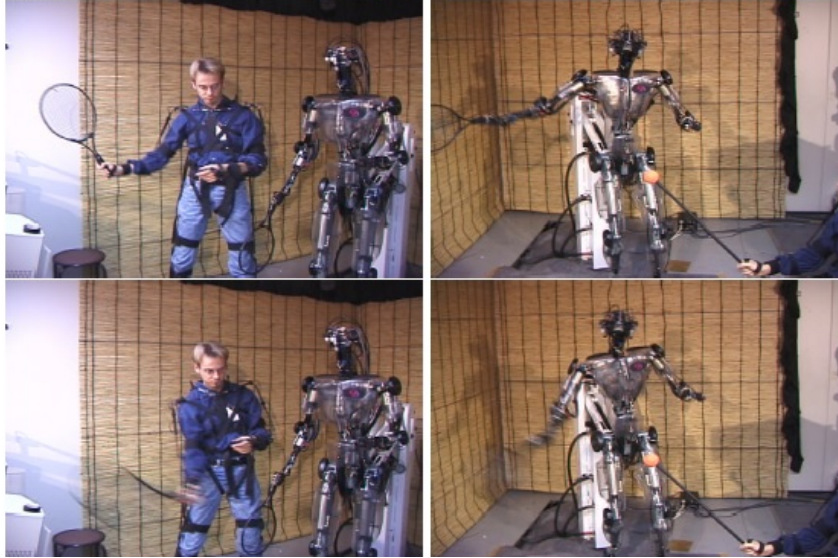
Figure 2.9: **Left:** Schematic overview of the proposed framework. **Right:** Tactile correction for learning grasp adaptation. The teacher indicates adaptability within the compliance constraints of a hand posture (Correction). The learner then replays the sequence of corrected poses (Reproduction). The final learned model is able to adapt the pose in response to different contact signatures (Perturbation). **Objects:** small can (left panels), ruler (right, top), large can and box (right, bottom).

and direction should be used in each situation, in other words, the optimal pair of the sought variables should be computed for each input vector. The problem is solved by using a supervised learning approach, providing a training set of appropriate parameters for different inputs. It should also be noted that the training data is field-specific, as each field requires different hitting parameters. The above works aimed at improving the DS global stability. To this end, they achieved to deal with demanding high level issues that can in real-time reciprocate to any perturbations and changes on the acting space.

In a recent work from the same group [78], the framework illustrated in Fig. 2.11 was proposed for real time catching of flying objects with rigid but uneven mass distributions and non-rigid mass distributions. They introduced a system composed by two iterating threads. The first thread continuously predicts the object trajectory and iteratively updates the best-catching configuration and catching time with each new measurement of



(a) Learning to play mini golf [71].

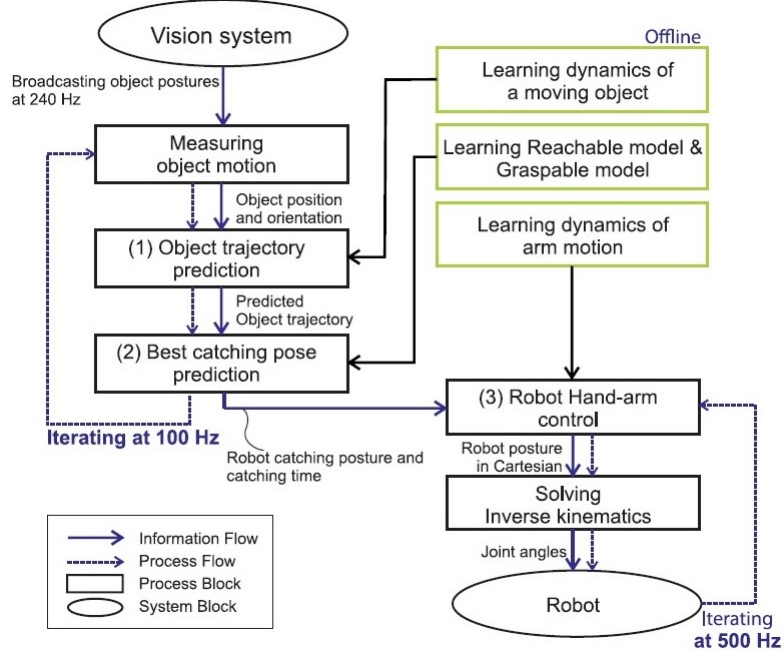


(b) Learning to play tennis [3].

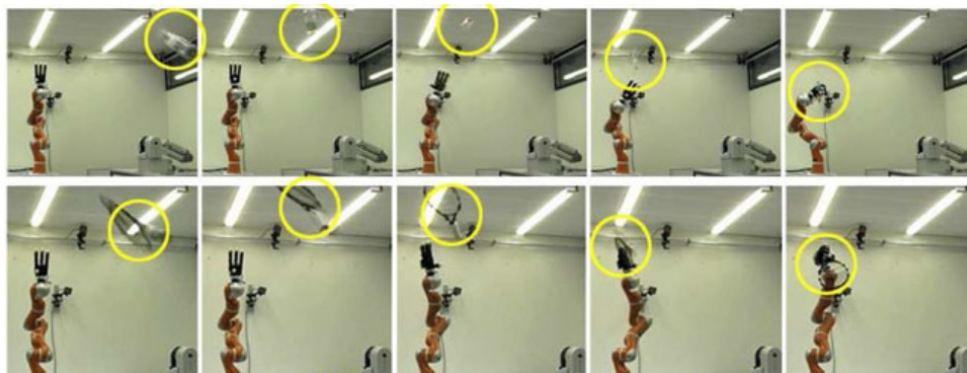
Figure 2.10: Learning sporting skills.

the flying object. The updated catching configuration is set as the target for the robot-arm controller. The second thread, i.e. the arm controller, continuously adapts the end-effector posture to the changes in the predicted best catching configuration and catching time. The arm controller computes the trajectory of the hand in Cartesian space. The latter is subsequently converted into joint angles by solving the Inverse Kinematics of the robotic system.

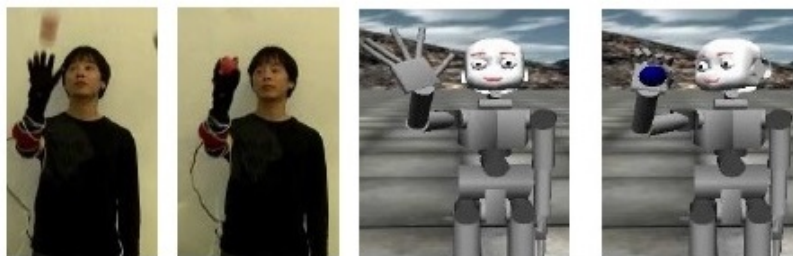
In order to determine the final catching configuration, a data-driven probabilistic approach was employed for estimating a distribution of admissible grasping posture on the object and computing the robot's reachable space. In the same paper, they further show that these techniques determine the optimal catching configuration in real time. To generate the robot arm and finger motion to intercept the object, DS models are used in conjunction with a timing controller, as the task is highly time-dependant. As noted by the authors, catching-failures were observed in real experiments, due to the fact that the robot's dynamics were not considered in the task-model.



(a)



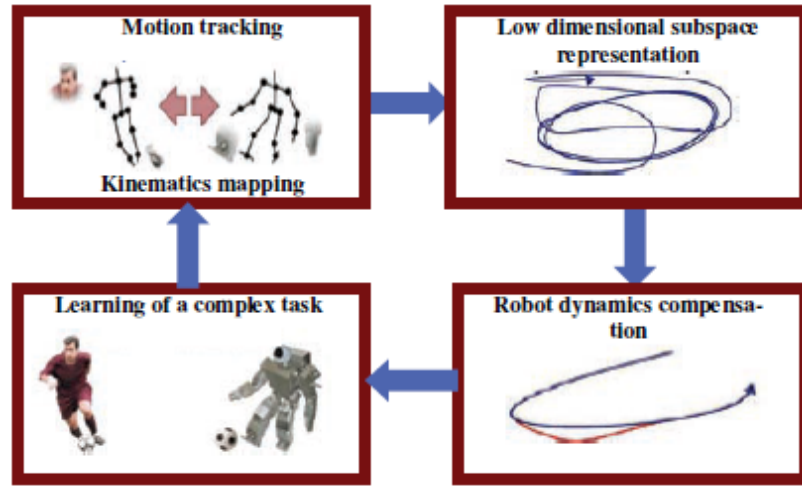
(b)



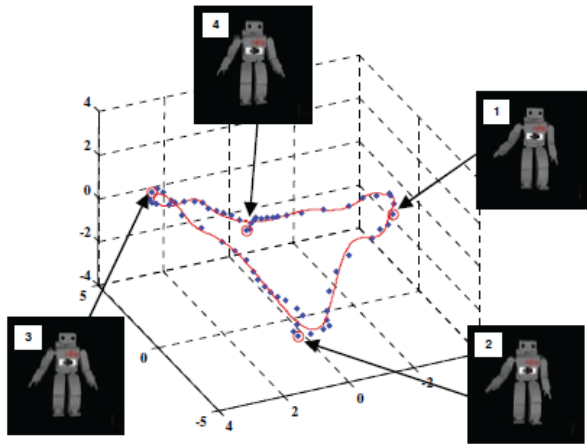
(c)

Figure 2.11: Imitation from motion capture as described in [96].

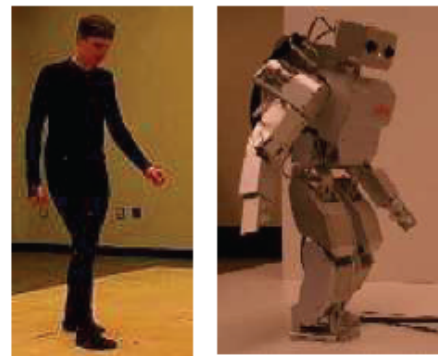
Additionally, probabilistic models and dimensionality reduction techniques in the context of LfD have been studied by the research group of P. N. Rao. In [107], [7], [108], they examined the LfD framework based on stochastic model formulation and linear dimensionality reduction methods that can reduce the dimensions of the acting space. To this end, in [107] a HOAP-2 robot was taught to walk by humans, as illustrated in Fig. 2.12, based on an intermediate, latent-space representation (see Fig. 2.12b). In this kind of representation, each latent point represents a robot's action state, whereas a sequence of latent points represents a complete movement.



(a)



(b) The resulting latent space.



(c) Robot learns to walk.

Figure 2.12: From latent space to robot's walking.

2.6.2 Reinforcement Learning: Real-time adaptation to the current environment

Once a new skill is learned by a robotic system after imitating a teacher's behavior, the next issue regards how it should perform when it is confronted with a novel (previously unseen) context. Early approaches addressed the above by having the agent ask the teacher explicitly. A better solution would be to allow the robot extract the necessary task-specific information from the demonstrations. This so constructed task-dependent skill is much more flexible than the context dependent one. Reinforcement Learning (RL) approaches, with origins in the Machine Learning field, have been employed in relevant works [109], [3]. RL, also referred in the literature as learning from experience, describes how a learning agent can achieve optimal behaviour based on interactions with its environment and reward feedback. RL enables a robot to autonomously discover an optimal behavior through trial-and-error interactions with its environment. Instead of explicitly detailing the solution to a problem, in RL the designer of a control task provides feedback in terms of a scalar objective function that measures the one-step performance of the robot. Especially in the field of Robotics, previous work focused on using prior knowledge from expert demonstrators to render the learning problem tractable and speed up learning [44], [55], [110], [111]. Even though RL as an LfD approach has had successes, physical and computational differences between demonstrator and learning agent, and limitations of the demonstrator, typically result in suboptimal demonstrations, compromising the quality of behaviour that is acquired by the agent [112], [113], [64].

The ability to learn from experience is a desirable trait in any learning system. The most popular approaches implemented within LfD systems to date offer reward-type evaluations of policy performance. Reward functions are often sparse, and only offer an indication of how desirable a given state is; reward gives no indication of which action would have been more appropriate. Richer forms of performance evaluation, however, could benefit approaches able to incorporate policy performance feedback. There is no reason to expect that formalizing these richer evaluations is easier than the admittedly challenging task of formally defining reward functions. One promising solution may be formulated by having the demonstration teacher provide feedback on learner performance, in addition to providing the example executions.

2.7 Learning Force-based Manipulation

In the current section we briefly review and highlight representative works from contemporary literature regarding robot learning based on force limitations. Recently, Falco et al. proposed a hybrid approach based on reactive control and Reinforcement Learning (RL) to implement in-hand manipulation skills with a low-cost, underactuated prosthetic hand

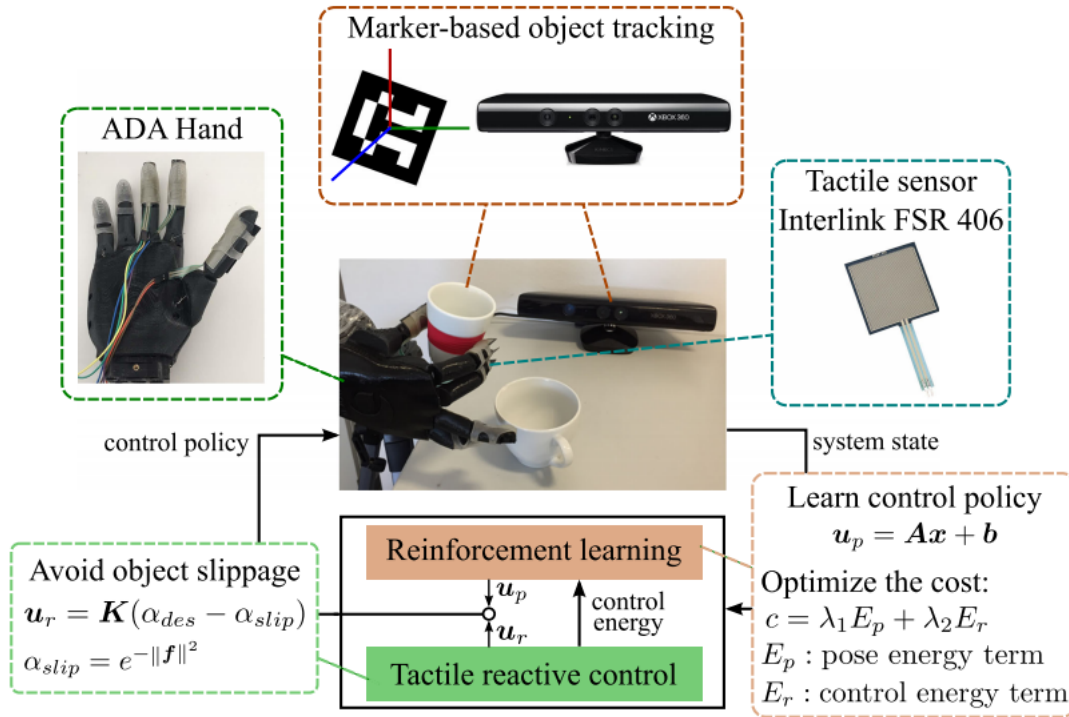


Figure 2.13: Overview of the proposed architecture in [114].

in the presence of irreversible events [114], [115]. The proposed architecture is depicted in Fig. 2.13. The two layers work in synergy through a bidirectional exchange of information. The Reinforcement Learning (RL) layer sends to the Reactive Control (RC) layer the trajectory associated to the current policy. The RC layer locally corrects the trajectory in order to avoid the object slipping during the learning procedure. On the other hand, the control layer sends to the learning layer information concerning the control energy during the trajectory, which measures how strong the tactile reactions were. Whereas, Kaelbling in [116] exploited hidden Markov models to plan and act in partially observable stochastic domains. Evrard et al. [117] adopted Gaussian mixture models to represent the variance over time in the demonstrated trajectories to teach a humanoid physical collaborative tasks. Gupta et al. [118] proposed an algorithm for policy learning and generalization, that allows complex dexterous manipulators to learn from multiple human demonstrations. Additionally, Li et al. in [119] learn grasp adaptation through experience and tactile sensing.

Execution of learned actions is in many cases affected by temporal and dynamical constraints or perturbations that are externally or naturally imposed [120], [121]. Various works have attempted to introduce contemporary learning methods into the force and impedance control to improve the tracking performance and the robustness of the repro-

duced robotic behavior [122], [123], [124], [125], [126]. In [127], NN control was applied in impedance controller to compensate the uncertainties in real time. Li and Liu [128] designed an adaptive impedance hybrid controller, which could implement the required contact force and track the target position in orthogonal subspaces with no other information of the environment.

Current methods that learn force-based behaviors directly from demonstrations or multiple trials [129, 130] typically use a least squares formulation to solve for time-varying impedances, either at each point in the demonstration or in the context of a Gaussian mixture model [131], [132]. In [133] the authors propose to intentionally introduce perturbations into the demonstrations to observe the human demonstrator's recovery stiffness, whereas in [134] more sophisticated input modalities were introduced to facilitate for the demonstrators to directly specify the desired stiffness [134]. In a different approach, Lee et al in [17] used a dataset of kinesthetically teaching trajectories to extract a single trajectory, along with time-varying feedback gains that determine the required poses and forces for a successful manipulation. Reciprocally, Rambow et al. in [135] used teleoperated demonstrations to autonomously manipulate deformable objects. Moreover, in [136], A. Nair et al. propose a combining self-supervised learning and imitation approach for vision-based manipulation of deformable objects such as ropes and clothes.

During the last decade, Machine Learning approaches have gradually appeared both in industrial applications [137], [138] as well as in house-robot scenarios [139], [140], [141]. Accordingly, our work follows a direct approach, whereby the system is trained by observing successful lifting and manipulation tasks, and learns to apply appropriate grasping posture and the suitable required force, and also generalize the latter for various object positions, shapes and masses (Chapter 5).

Chapter 3

Methodological Framework for Robot Learning by Observation

In this Chapter we introduce IMFO (IMitation Framework by Observation) as a novel LfD methodological framework to enable robots reproduce human actions, based on the coupling of perception and action, which is at the core of imitation learning. IMFO can cope with the reproduction of learned (i.e. previously observed) actions, as well as novel ones. By modeling the reciprocal interaction of perception (actor's world) and action (robot's world), the proposed framework effectively accomplishes to map the observed actor's space to the robot's one by formulating an intermediate, latent space representation.

Accordingly, IMFO succeeds in endowing robotic systems with human-like action capabilities. At first, an initial, observation phase is formulated whereby, by means of kinesthetic teaching, a set of demonstrated arm-motions is learned. In the learning process, respective human and robot actions are represented in the corresponding latent spaces and a mapping (association) across the latent spaces is established. In turn, a novel human action gives rise to a representation in the human latent space, which via the learned mapping is transformed to the robot's one. The latter is inversely mapped to the robot's action space effectively reproducing the observed behavior.

3.1 IMFO Formulation

IMFO comprises a complete methodological framework aiming at learning and reproducing human behavioral acts. It relies heavily on a latent space representation of both observed and produced motions. The introduced representation greatly facilitates generalisation of learned actions and abstraction of the robot kinematic configuration. Accordingly, two distinct phases can be identified within IMFO, a training and a reproduction phase. In turn, training consists of three steps:

- Observation, whereby, via an appropriate sensory set-up, data acquisition is performed regarding an arm-motion of the demonstrator and the corresponding one

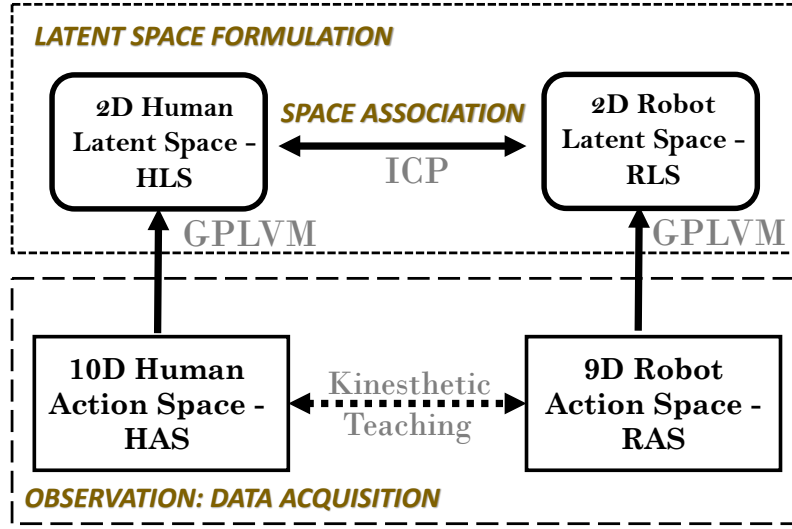


Figure 3.1: IMFO Training Phase.

of the robotic system.

- Latent space formulation, by means of GPLVM: observed actions are compactly represented in an appropriate latent space.
- Space association, is accomplished via the ICP algorithm, which builds a mapping across the two latent spaces.

Conceptually, IMFO training formulation is illustrated in Fig. 3.1. Following training any novel action presented to the robot can effectively be represented in the robot's latent space. Via an inverse projection this is subsequently mapped to the robot's physical configuration space, and as a result the action is reproduced by a robotic system. The above are presented in more detail in the following.

3.1.1 Training Phase

Observation-Data Acquisition

Imitation of actions via observation relies on data recorded by sensors that are usually external to the executing platform. Typically, visual sensors are used to record human-teacher actions. Motion capture systems utilizing visual markers on the observed joints are generally employed for data acquisition of human motion and manipulation acts [26].

In the case of IMFO, human actions are recorded by an external RGB-D camera system. By tracking the desired number of demonstrator's joints, a high dimensional actor's space $\mathbf{O} = \{o_1 \dots o_n\}$ of D dimensions is established.

Each demonstrated (observed) arm-motion is transferred to the robot’s action space by means of “kinesthetic teaching”. More specifically, we capitalise on the robot’s compliance to physically steer it through the task by the teacher. Accordingly, no explicit physical correspondence is needed, as the user demonstrates the skill on the robot’s own body. It also provides a natural teaching interface to teach a skill correctly and precisely [142].

In the current work we consider primitive arm-motion, and the human arm is tracked by placing prominent color markers on each joint, namely shoulder (J1), elbow (J2) and wrist (J3). For each joint the 3D coordinates (x, y, z) are derived by the external Kinect RGB-D camera [143]. Additionally, we use a binary index (0 or 1) to represent the state of the hand, i.e. 0 for closed fingers and 1 otherwise. Overall, the configuration of the human arm is represented as a point in a 10D space, termed “Human Action Space” (HAS).



Figure 3.2: **Left:** Example of joint tracking via color detection for the actor; **Right:** Kinesthetic teaching for the JACO arm.

The robotic system employed in this research is the JACO robotic arm, by Kinova Robotics, namely a six-joint arm manipulator. Its physical space is represented by the angles of the six joints and, moreover, the three angles of its hand-fingers. This results in a physical representation for the robotic system in a 9D space, termed “Robot Action Space” (RAS). A typical snapshot from the physical attachment of the robotic arm to the human arm is illustrated in Fig. 3.2, whereby the latter steers the former kinesthetically.

Latent Space Formulation

A fundamental step of the IMFO methodological framework comprises the representation of an action behavior in a suitable latent space. The relevant transformation is based on the Gaussian Process Latent Variable Model, GPLVM in short, which is used in order to derive the latent space representations [144]. GPLVM can be considered as a non-linear generalization of PCA. Effectively, it provides a more accurate representation of an abstract

multidimensional space projected to a lower dimensional one. The mathematical derivation of GPLVM is extensively described in the original publication [144]. Herewith, we confine the relevant presentation to the essential algorithmic steps that are implemented in order to formulate the two latent representations of the actor's and the robot's acting spaces.

GPLVM performs a non-linear dimensionality reduction in the context of Gaussian processes (GP); an algorithmic tabulation of its main steps is outlined in Alg. 1. The underlying probabilistic model, specified by GPLVM, is still a GP regression model, that needs to be learned, i.e. adjusting the Kernel's parameters (β). Theoretically, both the observed data $\mathbf{O} \in \mathbb{R}^D$ and the latent space data $\mathbf{X} \in \mathbb{R}^q$ (where, $q \ll D$) should be known a priori, thus, an initial estimation of \mathbf{X} is performed using PCA.

Once \mathbf{X} has been initialized, GP regressions and corrected estimations of \mathbf{X} are performed iteratively until convergence has been achieved or until a maximum number of iterations has been reached. To perform the GP regression, the likelihood of the GP and \mathbf{X} given \mathbf{O} has to be maximized with respect to the parameters of the kernel. In practice this optimization is performed using scaled conjugate gradient descent (steps 5-10 in Alg. 1), where β is Kernel's parameter and μ, n are parameters of the gradient descent process.

Algorithm 1 Kernel GPLVM

```

1: Given a set of observations  $\mathbf{O} \in \mathbb{R}^D$ ,  $\mathbf{N}$ : the number of iterations
2: Initialise latent variables  $\mathbf{X} \in \mathbb{R}^2$  through PCA
3: for  $\mathbf{N}$  iterations do
4:   Randomly Select a subset of  $\mathbf{X}$ ,  $\mathbf{x}_s \subset \mathbf{X}$  & find the neighbors around  $\mathbf{x}_s$  :  $\mathbf{X}_s = \mathbf{X} \in \mathbb{R}$ 
5:   Compute the gradients:  $\frac{\partial L}{\partial \mathbf{X}_s}$  and  $\frac{\partial L}{\partial \beta_s}$ 
6:   Update  $\mathbf{X}$  and  $\beta$  by:
7:    $\Delta \mathbf{X}_t = \mu_x \Delta \mathbf{X}_{t-1} + n_x \frac{\partial L}{\partial \mathbf{X}_s}$ 
8:    $\mathbf{X}_t \leftarrow \mathbf{X}_{t-1} + \Delta \mathbf{X}_t$ 
9:    $\Delta \beta_t = \mu_\beta \Delta \beta_{t-1} + n_\beta \frac{\partial L}{\partial \beta_s}$ 
10:   $\beta_t \leftarrow \beta_{t-1} + \Delta \beta_t$ 
11: end for

```

By the end of this phase, the two latent space representations are established. More specifically, the “Human Latent Space” (HLS) and the “Robot Latent Space” (RLS) are derived as sets of points $x_{ai} \in \mathbb{R}^q$ and $x_{ri} \in \mathbb{R}^q$, respectively. Via maximum-likelihood estimation of the intrinsic input data dimension, as proposed in [145], the dimension of HLS and RLS has been determined as $q = 2$; the latter was also experimentally verified.

Space Association

Having established the two latent point clouds, an appropriate matching transformation across corresponding points in HLS and RLS should be defined. Given two sets of corre-

sponding latent variables $\mathbf{X}_a = \{x_{a1}, \dots, x_{an}\} \in \mathbb{R}^q$ and $\mathbf{X}_r = \{x_{r1}, \dots, x_{rn}\} \in \mathbb{R}^q$, the transformation is derived as a pair of translation t and rotation r that minimizes the sum of squared error:

$$E(t, r) = \frac{1}{n} \sum_{i=1}^n \|x_{ai} - rx_{ri} - t\|^2 \quad (3.1)$$

where x_{ai}, x_{ri} are corresponding points. The correspondence between the pairings is ensured by the observation phase of IMFO, where the trajectories are captured kinesthetically.

In this work the Iterative Closest Point (ICP) algorithm [146] is employed to solve eq. (3.1), as it is a powerful algorithm for calculating such geometric transformations across $2D$ spaces with high accuracy. ICP uses an iterative process to align the two point clouds and estimate the combination of rotation and translation using the mean squared error cost function. The resulting transformation pair $L_T = (t, r)$ can subsequently be used to map a novel point from HLS to the corresponding point in RLS.

3.1.2 Robot Reproduction Phase - Inverse Projection

The proposed IMFO framework, as explained above, can be employed to associate a human action behavior with a robotic one. Effectively, a correspondence is established in the two latent action representations. In order to reproduce a demonstrated novel action by a robotic arm, a final step is needed to transform the representation in the robot's latent space (RLS) backward to a robot's physical configuration, namely RAS. Formally speaking, this inverse process should be implemented as the inverse of GPLVM. This inverse transformation (marked as INV-Tr) is formulated, as proposed in [147], [148], as a high order interpolation of the training learned data, using Radial Basis Function (RBF).

In IMFO this is addressed by computing off-line the latent representations (RLS) of a sufficiently large population of physical configurations (RAS) of the robotic arm. In practice, we iterate over all arm-DoFs, and for each arm configuration in RAS, the representation in latent space (RLS) is obtained via the GPLVM. The above iterations are performed by (a) respecting the physical limits of each arm-joint, and (b) employing an appropriate iteration step. A small step value results in a denser representation of the RAS-RLS pairings. For points in RLS that are not included in the above pre-computed pairs, the corresponding points in RAS are derived by interpolation. Experimentally it has been established that neither the step-value nor the actual method of interpolation were critical. This is due to the fact that the employed inverse transformation should not render an accurate (exact) replica of a demonstrated act, but rather a reproduced robotic behavior sufficiently similar to the latter.

Accordingly, the following steps are employed to effectively reproduce a novel arm motion by the robotic system (Fig. 3.3):

- Novel act observation: the act is perceived and mapped as a trajectory in HAS.
- Human latent space transformation: GPLVM is employed to represent the above trajectory in HLS.
- Mapping to robot latent space: the $L_T = (t, r)$ mapping across the two latent spaces is used to obtain the representation in RLS.
- Action reproduction: the formulated inverse transformation INV-Tr maps the latter representation to RAS, therefore accomplishing recreation of the observed action.

Figure 3.3 presents a block diagram of the above outlined steps.

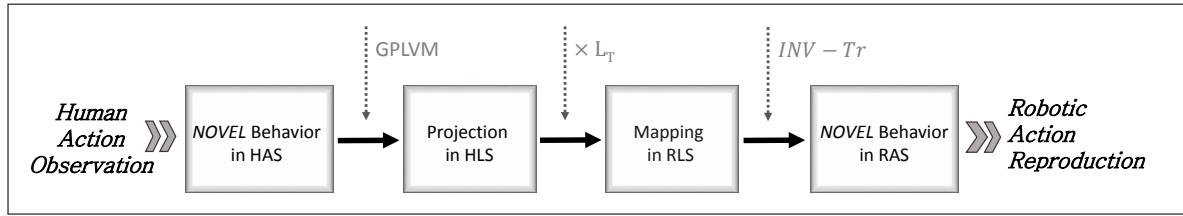


Figure 3.3: IMFO Reproduction Phase.

3.2 IMFO Evaluation

Extensive experimentation has been performed to assess the performance of IMFO methodological framework in realistic scenarios. As mentioned above, the relevant experimental set-up regarding the training phase, involved kinesthetic teaching in order to demonstrate certain action behaviors to the robotic arm. In all our experiments a six-joint arm manipulator was used, namely the JACO robotic arm, by Kinova Robotics. Its joints can be controlled independently either with position-control or with torque-control. Accordingly, the compliant mode of the JACO arm was used in order to physically steer the arm to execute various motion trajectories. The training dataset is composed by sequences of paired poses executed by the two agents, i.e. actor human arm and JACO robot arm, respectively. More specifically, the training actions recorded were basic right-arm-movements of reaching, pushing and grasping with differences in the posture configuration.

The employed procedure, besides being simple and robust, facilitated the execution of multiple demonstrations. Accordingly, the described procedure, i.e. action demonstration, action replication by the robotic arm, registration of HAS-RAS parameters, was repeatedly executed for an adequate number of manipulating actions. More specifically, the training phase included 30 actions, of approximately 200 frames each (at 30 *fps*), hence a number of 6000 configuration poses. In other words, the result of the training phase was the population of HAS and RAS with 6000 points each. In our experiments we concluded

that a number of 30 actions was completely satisfactory, by starting from a rather small number of 10 actions and gradually increasing the training data set. Although 10 actions might seem a small number, experimentally it was verified that they span densely HAS and RAS and, moreover, are adequate for the considered human-like scenarios. By progressively increasing the action count to 30, algorithm's scalability has also been verified.

Following the algorithmic steps of IMFO as described in Section III, the corresponding latent space representations have been extracted. In other words, via GPLVM the above HAS and RAS point clouds have been transformed to HLS and RLS representations, respectively. The latter, for a sample subset of seven training trajectories (out of the total 30 training actions), marked as *Train1* . . . *Train7*, are provided in Fig. 3.4. The left part of Fig. 3.4 illustrates the HLS representations, whereas the right part illustrates the corresponding RLS ones. As can be observed, the obtained representations are appropriate for the given task, since different actions result in distinct trajectories in the latent spaces. Moreover, the neighbouring property of GPLVM is effectively demonstrated, resulting in adjacent latent points for adjacent points in the physical space.

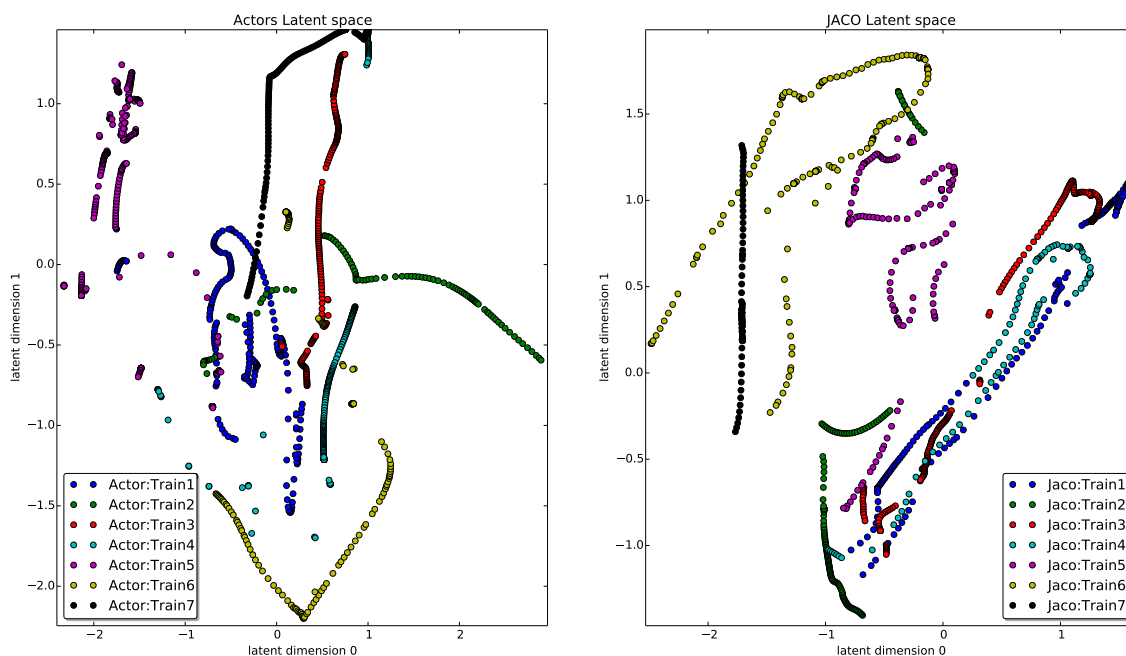


Figure 3.4: Derived latent representations for actor and JACO arm, respectively.

We note at this point that the derived RLS depends on the specific kinematics of the employed robot. If the JACO robot were to be replaced by another one, a different latent space representation would result. Such differences are in practice not observed in the derived HLS. Substitution of the actor by another one (with minor arm-kinematic differ-

ences) does not actually affect the actor’s latent space.

3.2.1 Scenario WRITE

Having established the training dataset of learned actions as described above, we adopted two scenarios to systematically evaluate IMFO. In the current section we present in detail the first evaluation scenario, namely WRITE whereas Section IV-B accounts for the second, namely PICK-PLACE. More specifically, in scenario WRITE we conducted a set of experiments whereby the robotic arm was assumed to write its own name, that is the word “JACO”. In other words, the observed novel actions were the four actions that correspond to the writing of the four letters “J”, “A”, “C”, “O” on a flat level surface. Snapshots from the above described procedure are illustrated in the right column of Fig. 3.5, where the human actor is performing the motions for the four letters.

According to the steps outlined in Section III, the trajectories of the above four novel acts are projected to HLS via the GPLVM. L_T is then employed to map the latter representations to RLS. Finally INV-Tr is invoked to inversely transform the trajectories to RAS, effectively resulting to the motions being reproduced by the robotic arm. The left column of Fig. 3.5 shows sample instances from the reproduction of the four named letters.

The above experimental procedure was repeated 40 times in order to extract statistically meaningful error measures. Figure 3.6 illustrates the trajectories that correspond to the four letters in the latent spaces. Figure 3.6(Left) refers to HLS, whereas Fig. 3.6(Right) shows the representation in RLS. For visualisation purposes, both figures present the mean trajectories over the 40 iterations. Additionally, the covariances over the individual trajectories are depicted on the figures as gray scales. As can be observed, the uncertainty in the trajectories is rather limited for the case of the human actor. This is due to the fact that the four motions were performed by the same actor and hence they were very similar in subsequent iterations. The trajectory uncertainties are somewhat higher in RLS, which is due to the L_T transformation that performs the mapping from HLS to RLS. Nevertheless, visual inspection of the induced covariances in both plots of Fig. 3.6 verifies that the uncertainty introduced by L_T is insignificant and, as will be shown in subsequent results, within acceptable limits.

Aiming at a rigorous quantitative assessment of IMFO, we introduce herewith three error measures that quantify (a) RLS consistency and repeatability (E_{RLS}); (b) RAS consistency and repeatability (E_{RAS}); (c) robotic end-effector trajectory (E_{EF}).

RLS consistency and repeatability - E_{RLS}

Human actions that are very similar (almost identical) should result to RLS representations with small deviations. Such actions are the ones that correspond to the 40 iterations for each of the four letters in “JACO”. In other words, the 40 HAS trajectories for each letter

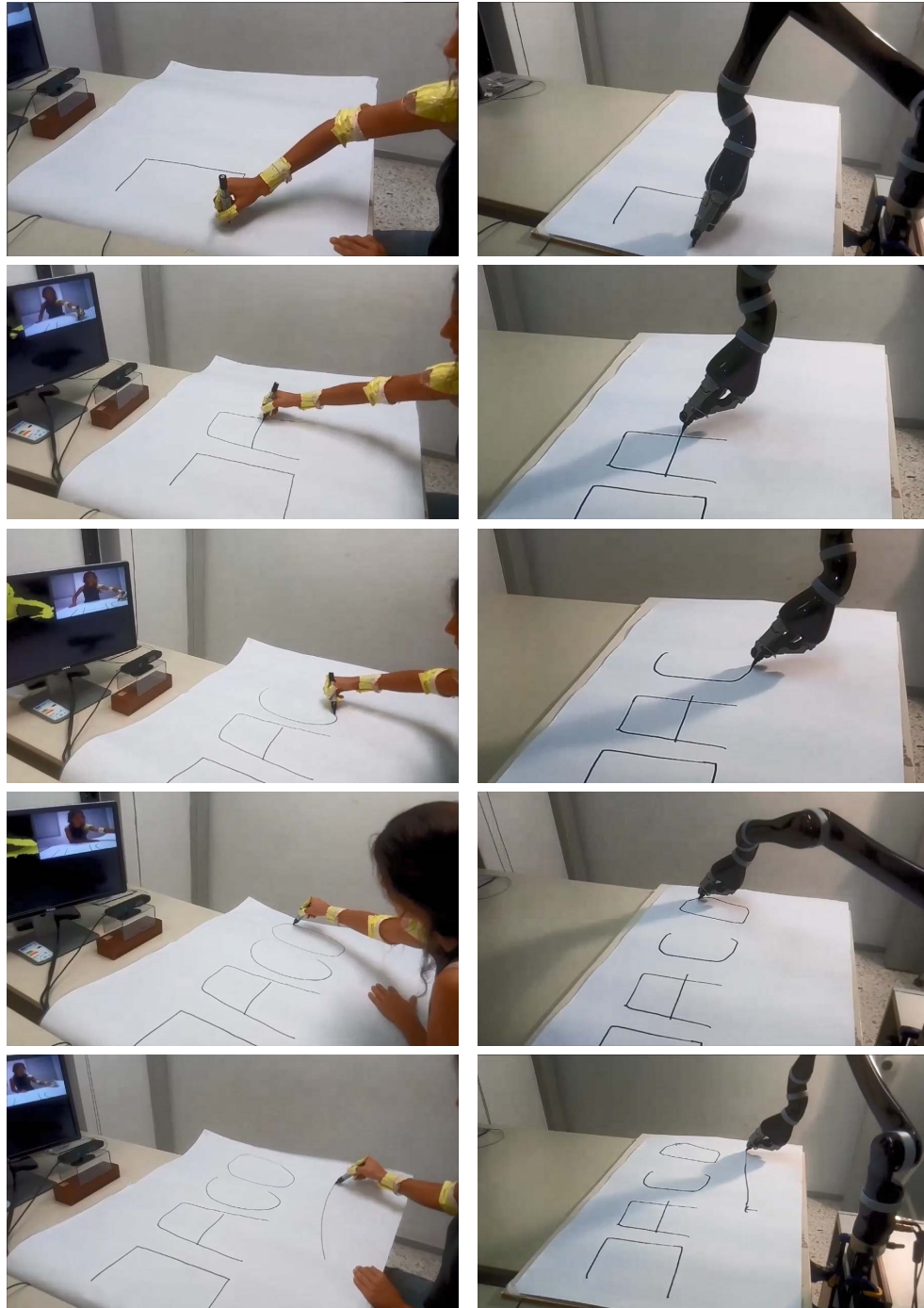


Figure 3.5: **Right column:** Demonstration of actions corresponding to the letters of the word “JACO”; **Left column:** Robotic reproduction of the same actions.

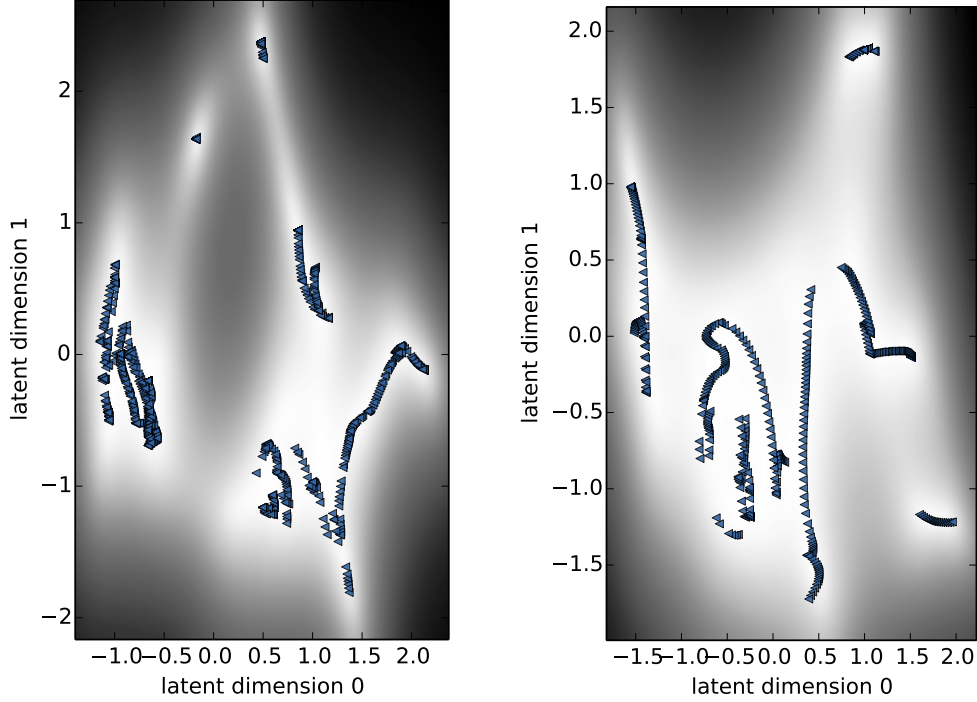


Figure 3.6: Illustration of the novel latent trajectories with each variance: **Left:** HLS representations for the four novel actions. **Right:** Corresponding RLS representations.

should ideally give rise to 40 similar RLS trajectories. The latter is quantitatively assessed with the E_{RLS} error, defined as follows.

Let a trajectory $\mathcal{X}_r^i = \{x_{r1}^i \dots x_{rM}^i\}$ in RLS. Let also N such trajectories corresponding to N similar actions in HAS ($N = 40$ in our case) and let $\bar{\mathcal{X}}_r$ be the mean trajectory over all N trajectories. The Mahalanobis distance of \mathcal{X}_r^i from $\bar{\mathcal{X}}_r$ is obtained as:

$$\mathcal{D}(\mathcal{X}_r^i, \bar{\mathcal{X}}_r) = \frac{1}{M} \sum_{k=1}^M \|x_{rk}^i - \bar{x}_{rk}\|_{\mathbf{S}^{-1}}^2 \quad (3.2)$$

where M is the number of points in each trajectory. The mean over all $\mathcal{D}(\mathcal{X}_r^i, \bar{\mathcal{X}}_r)$ errors provides a quantitative measure of action consistency and repeatability in RLS; it is computed as:

$$E_{\mathcal{D}} = \frac{1}{N} \sum_{i=1}^N \mathcal{D}(\mathcal{X}_r^i, \bar{\mathcal{X}}_r) \quad (3.3)$$

where $N = 40$ is the number of iterations. As a final step, $E_{\mathcal{D}}$ is normalized according to the actual magnitude of RLS. In practice, the loci of points in RLS, after the training phase, approximates the surface of an ellipsoid. We take as the RLS magnitude the ellipsoid characteristic variable $d = \sqrt{a^2 - b^2}$, where a, b are the semi-major and semi-minor axes, respectively. Accordingly, E_{RLS} is computed as:

$$E_{RLS} = \frac{E_{\mathcal{D}}}{d} \% \quad (3.4)$$

The computed values of E_{RLS} for the four letters of the word “JACO” are given in Fig.3.7 (black error bars). As can be observed, the errors in all cases are very small, ranging from 1% to 4%.

RAS consistency and repeatability - E_{RAS}

Trajectories in RLS give rise, via the INV-Tr, to trajectories in RAS. The latter defines the actual robotic action that is reproduced as a result of a demonstrated novel action by a human actor. Following similar formulation as in the case of E_{RLS} , we compute E_{RAS} in order to quantitatively characterize the reproduction of novel actions by the robotic system.

Given a trajectory $\mathcal{X}_R^i = \{x_{R1}^i \dots x_{RM}^i\}$ in RAS, and $\bar{\mathcal{X}}_R$ be the mean trajectory over all N trajectories, we define in a similar fashion the distance of \mathcal{X}_R^i from $\bar{\mathcal{X}}_R$ as:

$$\mathcal{D}_{\mathcal{R}}(\mathcal{X}_R^i, \bar{\mathcal{X}}_R) = \frac{1}{M} \sum_{k=1}^M \|x_{Rk}^i - \bar{x}_{Rk}\|^2 \quad (3.5)$$

Accordingly, the mean error in RAS is obtained as:

$$E_{\mathcal{D}_R} = \frac{1}{N} \sum_{i=1}^N \mathcal{D}_{\mathcal{R}}(\mathcal{X}_R^i, \bar{\mathcal{X}}_R) \quad (3.6)$$

Finally, $E_{\mathcal{D}_R}$ is normalized according to the magnitude of JACO’s operational space. The latter is approximately a sphere of radius $r = 0.9m$ (e.g. the absolute stretching of the arm). Hence, the normalized error E_{RAS} is obtained by:

$$E_{RAS} = \frac{E_{\mathcal{D}_R}}{r} \% \quad (3.7)$$

Figure 3.7 summarizes the obtained E_{RAS} errors (grey error bars), contrasting them with the previously computed E_{RLS} errors. As expected, physical space errors are relatively

higher, and this is due to the fact that, errors in the physical robot's space are derived after the inverse mapping transformation INV-Tr, which unavoidably increases the error. Further to that, the low error values in both RAS and RLS are commendable. As such they serve as the experimental verification of the adopted means to obtain the latent space representation and, from that, the actual reproduction of the observed action.

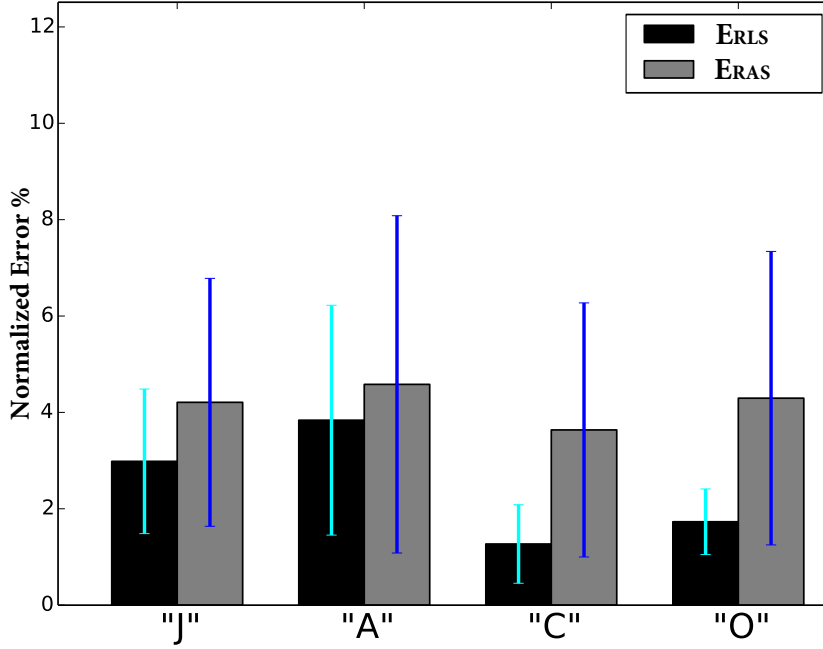


Figure 3.7: Error bars denoting consistency and repeatability in RLS and RAS.

Robotic end-effector trajectory - E_{EF}

An additional error measure is calculated as a metric of imitation of the end-effector. More specifically, E_{EF} is assumed to quantify the precise reproduction of a demonstrated act by the robotic end-effector, and it is defined as the 3D-error in the latter's trajectory. Thus, for a mean actor's trajectory $\bar{\mathcal{X}}_A = \{\bar{x}_{A1} \dots \bar{x}_{AM}\}$ and the corresponding reproduced by the robot $\bar{\mathcal{X}}_R = \{\bar{x}_{R1} \dots \bar{x}_{RM}\}$, E_{EF} is given by the formula:

$$E_{EF} = \frac{1}{M} \sum_{k=1}^M \|\bar{x}_{Ak} - \bar{x}_{Rk}\|^2 \quad (3.8)$$

Interestingly, in all four cases E_{EF} resulted in very small values, and specifically less than 20 millimetres. More importantly, the computed errors are negligible for the studied imi-

tation tasks, and do not affect the behaviors exhibited by the robotic system.

3.2.2 Scenario PICK-PLACE

The above described evaluation scenario resulted in very promising quantitative metrics. By design, in scenario WRITE the robotic end-effector was confined to move on a 2D plane. Although this does not limit by any means the performed IMFO evaluation, we conducted an additional scenario whereby the above restriction was removed. The latter, namely scenario PICK-PLACE, assumed a 3D trajectory of the robotic end-effector in a pick-and-place task.

Accordingly, the demonstrated action consisted of (a) picking a *blue object* from a box (orange box in Fig. 3.8) using a top spherical grasping configuration, (b) following a 3D trajectory in space, and (c) placing it on a shelf with a rotated (lateral) posture. Snapshots of the human demonstration, along with the robotic reproduction are shown in Fig. 3.8.

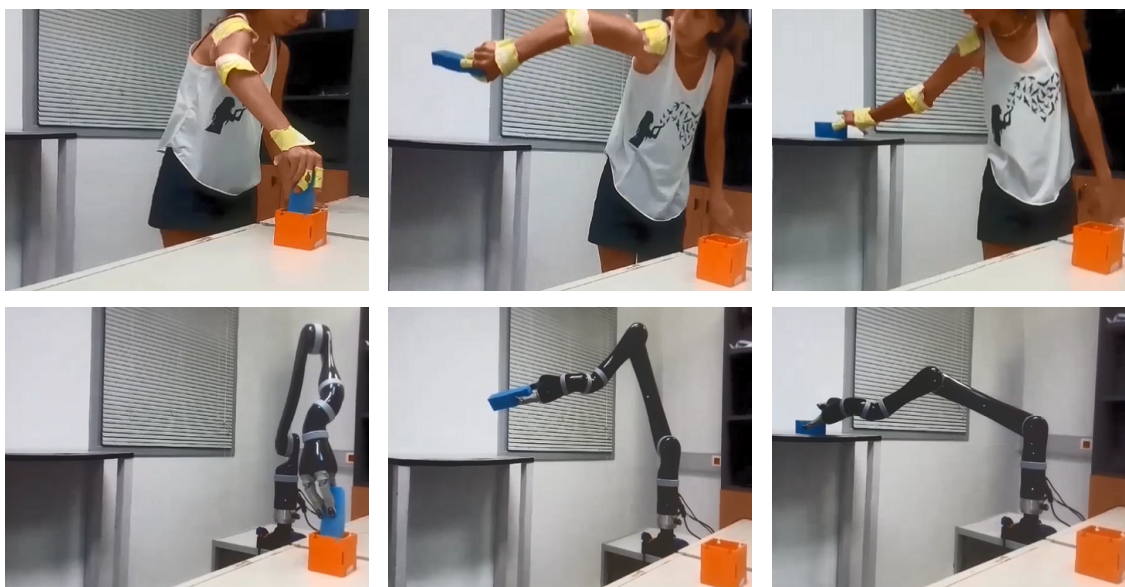


Figure 3.8: Scenario PICK-PLACE. **Top row:** Demonstration of a novel pick-and-place action; **Bottom row:** Robotic reproduction of the demonstrated action.

Similarly to the above described scenario WRITE, the current pick-and-place task was repeated 40 times, in order to use the same performance metrics as above. The obtained E_{RLS} , E_{RAS} and E_{EF} for the PICK-PLACE scenario are tabulated in Table 3.1.

As can be observed, even in this more demanding scenario, very accurate performance metrics were obtained. The consistency and repeatability measures assumed slightly higher values compared to the previous evaluation scenario, whereas the error in the end-effector

trajectory increased by very little.

Effectively, the obtained results provide important evidence regarding the accuracy and robustness of the proposed framework. This is also demonstrated in the provided snapshots from our experiments, and in particular in the supplementary video that documents IMFO and the conducted evaluation scenarios (<https://www.youtube.com/watch?v=d-ggpYQbm3Y>).

Table 3.1: Indicative Errors for the novel action behavior.

E_{RLS}	5.57%
E_{RAS}	5.98%
E_{EF}	47mm

3.3 Applications in HRI

As it is noted, we have set the goal, throughout this thesis, to formulate an LfD methodological framework, based on a latent space representation of the actors' configurations using dimensionality reduction, for learning robotic action behaviors. An advantage of the proposed formulation over other state of the art methods for imitation learning, as already mentioned, is that the mapping between the observed and the action spaces is established independently of the teacher's and the robot's kinematics. At the same time, LfD methods are usually faced with the correspondence problem across the two (teacher-learner) spaces [149], [150]. The latter is greatly facilitated in our approach with the employment of the latent space representation. As a result, besides successful reproduction of demonstrated actions, the proposed approach is also capable to learn novel action behaviors. To this end, the latter behaviors can credibly be applied in real Human-Robot Collaboration (HRC) scenarios for cooperative task execution [151]. The latter constitutes a main contribution of our work, demonstrating effective HRC.

3.3.1 Novel Action Execution

In this part, novel action imitation was tested on the JACO robotic arm. JACO's compliant mode of operation is particularly useful in order to appropriately steer the arm and hence have it execute (learn) a desired action behavior.

Here, the physical spaces of the demonstrator and the robot were recorded by utilizing information for the joints of the respective arms. In the case of the human actor the three arm joints were used, namely shoulder, elbow and wrist, and for each one of them the 3D coordinates (x, y, z) were derived. Prominent color markers were placed on each joint, and a Kinect RGB-D camera was utilized to unambiguously track the joints and extract their 3D

coordinates (see Fig. 3.9a) [152], [153]. By tracking the three arm joints of the teacher as above, a $9D$ representation of the observed physical space is formed. The above procedure, besides being simple and robust, facilitated the execution of multiple demonstrations.

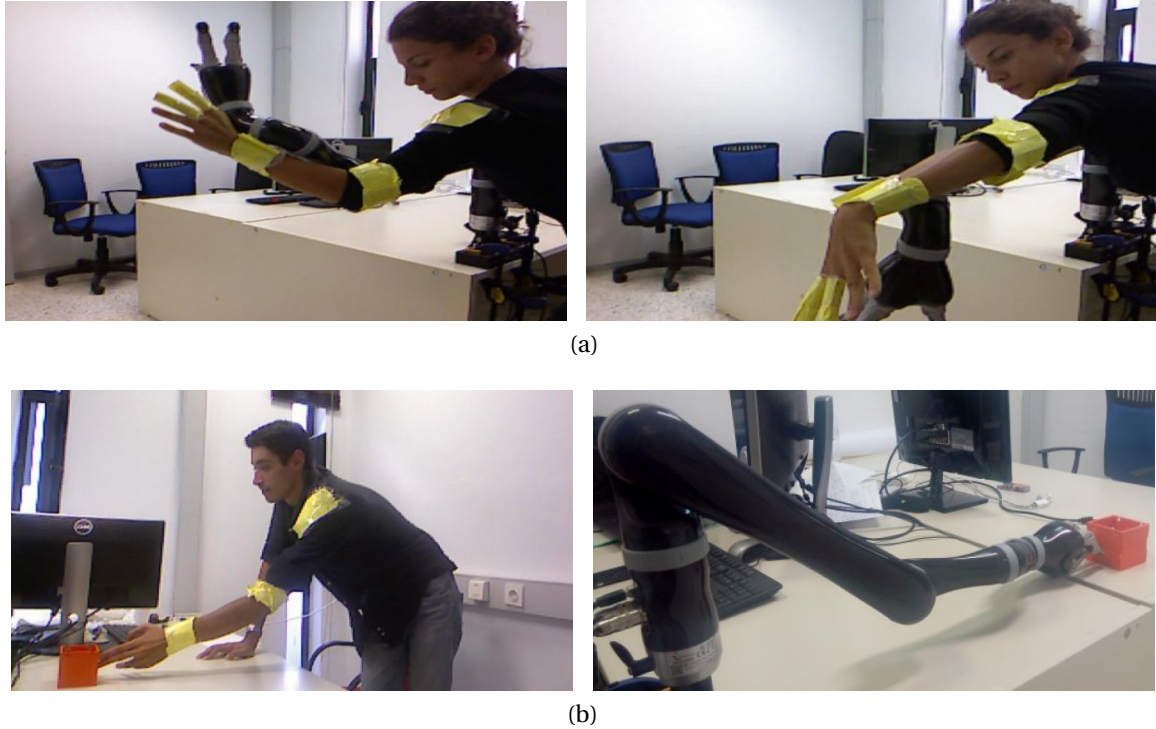


Figure 3.9: (a) Kinesthetic training of the robotic arm. (b) A novel human motion behavior (left), reproduced by the robotic arm (right).

Regarding the robot's physical space, we obtained the angles of the arm's six joints during a performed act via the available encoders. This resulted in a $6D$ physical representation for the robotic system. The above described procedure, i.e. action demonstration, action replication by the robotic arm, registration of the physical space parameters for both arms, was repeatedly executed for an adequate number of reaching actions. More specifically, the training phase was completed with 30 actions, of 30 *fps* each. In our experiments we concluded that a number of 30 actions was utterly satisfactory, by gradually increasing the training data set, starting from a rather small number of 10 actions. Although 10 actions might seem a small number, experimentally it was verified that they span densely the action space and, moreover, are adequate for the considered HRC tasks. By gradually increasing the action count to 30, algorithm's scalability has been verified. It is noted at this point that actions with different durations are regarded different in our implementation, since time is not directly taken into account in the proposed method but rather is considered as an intrinsic feature of the motion-action captured. Examples of

the performed actions by the human teacher are shown in Fig. 3.9a. In the same figure, the physical attachment of the robotic arm to the human arm is also illustrated. Accordingly, the robotic arm performs an arm motion similar to the one performed by the human arm.

Following the above, the latent space representation has been extracted for both cases, human and corresponding robot trajectories. A home-made implementation of GPLVM has been utilized for this task. Experimentally it was established that a latent space dimension of 2 was appropriate for the tested actions. Nonetheless, theoretical verification of the above has also been obtained via maximum-likelihood estimation of the intrinsic input data dimension, as proposed in [145]. For illustrative purposes, four sample motions have been identified, and the corresponding latent space representations are provided in Fig. 3.10, with the four motions being marked as $D1 \dots D4$. In Fig. 3.10b the named representations are given for the case of the demonstrator, whereas Fig. 3.10a shows the respective plots for the robotic arm. As can be observed, the obtained representations are appropriate for the given task, since different actions result in distinct trajectories in the latent spaces. Moreover, the neighbouring property of GPLVM is effectively demonstrated, resulting in adjacent latent points for adjacent points in the physical space. As a last step, ICP was used to establish the correspondence across the two latent spaces and form the latent hyperspace.

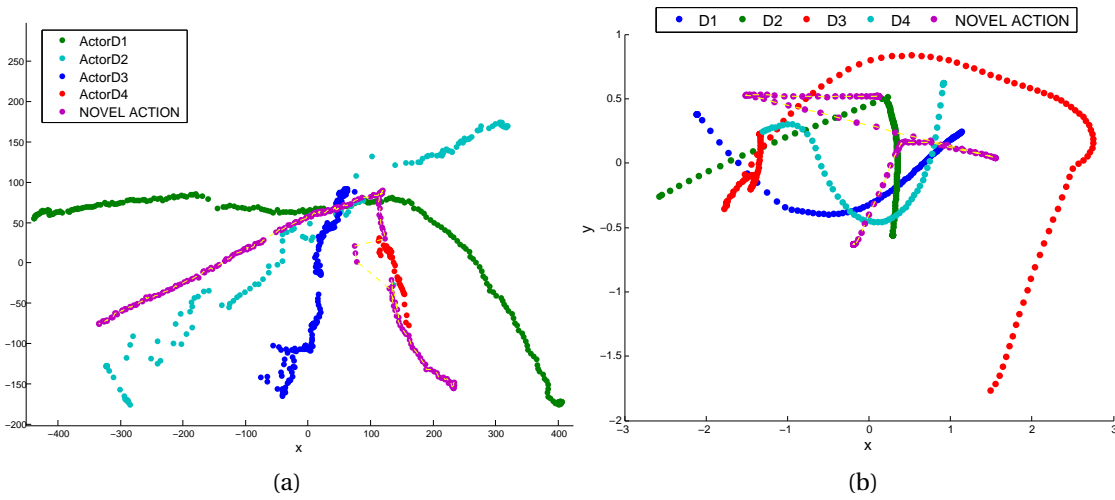


Figure 3.10: Extracted Latent Hyperspace formulated by five sample trajectories marked as $D1 - D4$ and *NOVEL ACTION*, accomplished by actor and JACO, respectively: (a) Actor's Latent Space, and (b) JACO's Latent Space.

To examine the validity of the extracted latent hyperspace, a set of experiments was

conducted whereby, firstly, a newly presented action behavior was asked to be reproduced by JACO and in turn, the latter learned actions (novel or not) are involved in human-robot collaborative tasks. At first, a novel action was either performed by a different human actor, or by the same one like above but this time performing a behavior not included in the set of learned actions. Fig. 3.9b shows a sample experiment, where a different actor performs a “docking action” (pushing the orange box). This 120-frame sequence of $9D$ vectors is projected to the formatted actor’s latent space of Fig. 3.10b, marked as *NOVEL ACTION*. By left multiplying the latter with the transformation matrix obtained by the space association algorithm, the initial actor’s trajectory is projected to the JACO’s latent space, shown in Fig. 3.10a. Finally, the representation of the indicative movement in JACO’s joints angles is computed by applying INV-Tr, resulting in a representation in the actual joint space of JACO. This action can be effectively reproduced by JACO; a snapshot of this behavior is shown in Fig. 3.9b.

3.3.2 Cooperative Task Execution

This category of experiments regards the validation of realistic HRC scenarios on two robotic systems, i.e. six-joint arm manipulator JACO KINOVA arm and the NAO humanoid robot by Aldebaran Robotics which comprises a small humanoid with 25 DoFs; in the case of the latter. For the tasks at hand, posture control was accomplished as in [154], whereas only one arm of the humanoid with six DoFs was used. Evidently, the kinematics of the two systems, i.e. the JACO arm and the NAO arm, are completely different. The latter allowed us to validate the abstraction property of the latent space representation.

Accordingly, the learned actions by a robotic system were utilized in a real HRC context, where the robotic system was expected to perform specific actions. The actual experiments conducted assumed HRC scenarios, whereby a human was opening a closed compartment and removing an object from it. Subsequently, the robot was expected to perform the correct action to properly close the compartment. Example cases of compartments tested involve drawers, closets and cabinets. The respective actions that the robotic systems should perform consisted of “forward pushing” (close a drawer), “side-ward reaching” (close an open closet door) and “reaching from the top” (close a cabinet). Furthermore, the robots were trained with two additional actions namely “hello-waving” and “sideways pointing”. Those additional actions were included in order to furnish the robots with richer repertoire of actions and hence operate more realistically in the HRC scenarios.

The above actions were demonstrated by a human actor and learned by both robotic systems, in a similar way as already explained in Subsection 3.3.1. Figure 3.11 presents the latent spaces that were registered for the human actor and the NAO arm in this training session. In the actual execution of the HRC scenario, visual recognition of the relevant

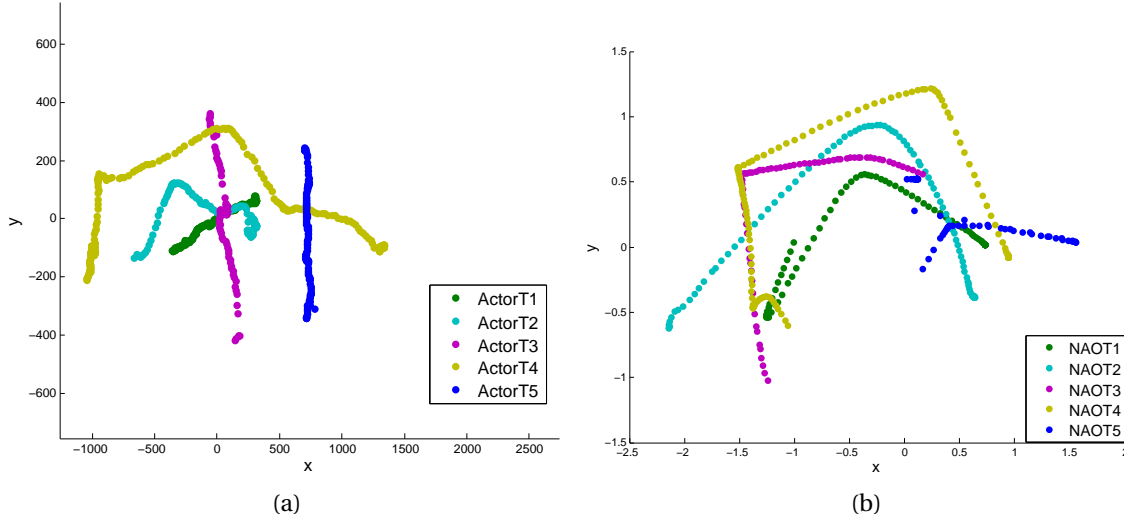


Figure 3.11: Extracted NAO's Latent Hyperspace, formulated by five sample trajectories $T1 - T5$, accomplished by actor and NAO, respectively: (a) Actor's Latent Space, and (b) NAO's Latent Space.

compartment was used in order to index in the latent space and retrieve the correct action representation. In addition, the robotic action was triggered by visual detection of the removed object based on color information [155]. Interestingly, in all HRC experiments conducted, both robotic systems collaborated successfully and capably closed the respective compartment by employing the correct action behavior (https://www.youtube.com/watch?v=h7cMOX_hXGI). Example cases are illustrated in Fig. 3.12 for both robotic systems employed in the relevant experiments.

Quantitative Evaluation in HRI

According to [4], the definition of imitation metrics in LfD is task specific. In other words, the task under consideration may give rise to appropriate metrics for quantitatively characterizing execution of the learned behavior. A general formula has been proposed in [4], as:

$$J = \|\xi_a^\theta - \xi_{rob}^\theta\|_{W^\theta}^2 + \|\xi_a^x - \xi_{rob}^x\|_{W^x}^2 + \|\xi_a^y - \xi_{rob}^y\|_{W^y}^2 \quad (3.9)$$

where, $\xi_a^{\theta,x,y}$ are the actor's variables that are considered in the demonstrated action, e.g. 3D-coordinates of the arm's joints. Similarly, $\xi_{rob}^{\theta,x,y}$ are the corresponding robot's ones, and $W^{\theta,x,y}$ are appropriate weights that effectively scale the contribution of each variable in the final result. In order to quantitatively assess the reproduction of learned actions by a



Figure 3.12: HRC scenario whereby a robot assumes a learned action behavior to close an open closet door. **Top:** JACO arm; **Bottom:** NAO humanoid robot arm.

robotic platform, we have instantiated eq.(3.9) as follows:

$$ERROR_{Traj} = \|\xi_a^{EE} - \xi_{rob}^{EE}\|^2 \quad (3.10)$$

where ξ_a^{EE} and ξ_{rob}^{EE} stand for the end-effector's 3D trajectories (3D coordinates) accomplished by actor and robot, respectively. For the set of experiments conducted in our study, eq. (3.10) captures the error between the demonstrator's and the robot's 3D-trajectories and therefore quantitatively evaluates task reproduction. Fig. 3.13 illustrates the end-effectors' 3D-trajectories followed by the three agents involved, namely Actor, NAO and JACO, during the two HRI behaviors "*Close an open closet*" and "*Close an open drawer*", respectively.

Table 3.2 summarizes the quantitative evaluation results, as derived from eq.(3.10), for each of the five sample arm-motions that were used to formulate the latent hyperspaces of JACO and NAO respectively. The first two rows in each table correspond to the assumed trajectories for the "*Close an open closet*" and "*Close an open drawer*", respectively (see Fig. 3.13). As can be observed from the tabulated figures in Table 3.2, both robotic systems performed the learned actions with high accuracy. The 3D-trajectory errors are very small (less than 10 centimeters in all experiments). More importantly, the computed errors are negligible for the studied imitation tasks, and do not affect the behavior expressed by the

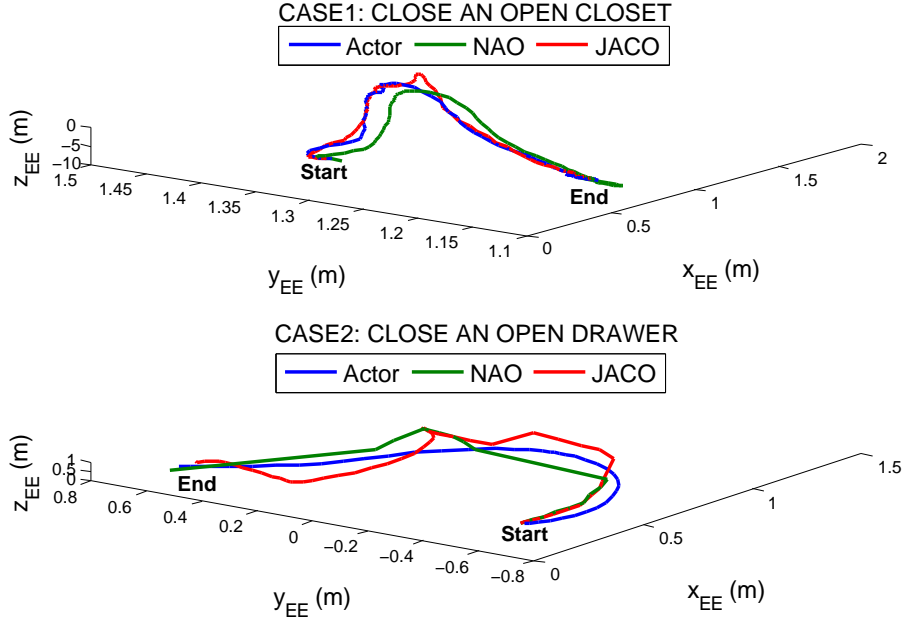


Figure 3.13: Trajectories of end-effectors' movement during the two aforementioned HRC tasks.

Table 3.2: Performance Metrics

	$ERROR_{Traj}(m)$	
Action	JACO	NAO
T1	0.086	0.089
T2	0.078	0.068
T3	0.049	0.063
T4	0.088	0.040
T5	0.064	0.056

robotic system.

Lately, the proposed formulation is also being tested in a time-aware multi-agent symbiotic scenario [156]. Following this scenario, the robotic action is triggered by visual detection of a red bowl based on color information. The HRI system is composed by an external RGB-D camera, a six-joint-arm, JACO Kinova Robot and the NAO humanoid robot. The JACO robotic arm is supposed to pick some objects and place them in a red bowl, that NAO robot carries on. Grasping movements are previously learnt through an off-line learning process. The robotic arm is taught to perform a variety of grasping movements

by observing a set of demonstrations provided by a human teacher kinesthetically.

Snapshots from the real experiment are illustrated in Fig. 3.14, where both robotic systems collaborated successfully in order to serve breakfast to the human agent on time.

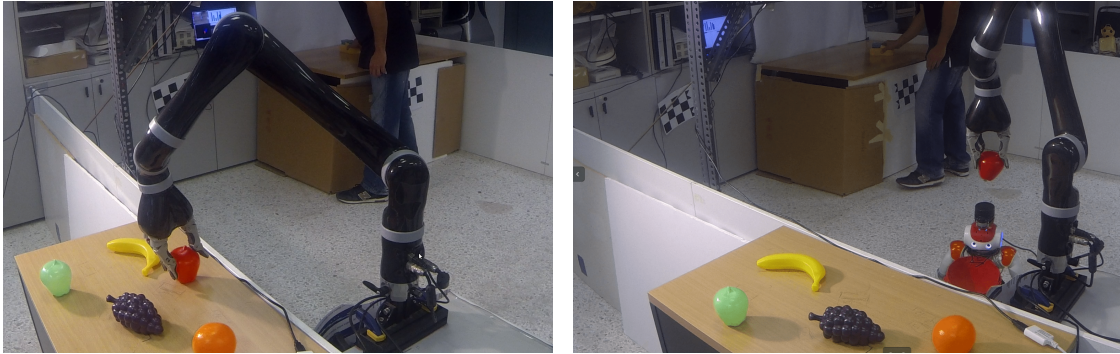


Figure 3.14: Time-aware multi-agent symbiosis.

3.4 Chapter highlights

In this chapter we dealt with the formulation and experimental assessment of IMFO, as a novel LfD methodological framework. Accordingly, LfD has been approached by employing a low dimensional, latent space representation that effectively abstracts small variations in demonstrated actions. It does so by recording significant features of arm-motions, abstracting at the same time irrelevant differences that stem from different embodiments and arm kinematics. The employed GPLVM transformation from the actor's physical space to the actual latent space was proven very effective in appropriately encoding the salient, pertinent information from a motion act by robustly formulating the aforementioned space. More specifically, different actions result in well separated trajectories in the latent space, and also the neighboring property of a 3D trajectory is maintained, by the continuous curves that are formed in the latent space. Additionally, we have studied and validated IMFO methodology and have successfully applied this in HRC task execution scenarios. In the current work an action behavior was deemed successful, provided that the reproduced trajectory was appropriate for the task at hand. For a more systematic assessment of learned actions we have also introduced suitable error metrics to cater for mismatches in the reproduced trajectories and measure the obtained errors.

The so far encouraging results attest for the validity and effectiveness of the proposed approach in various scenarios, namely "write" scenario, "pick-and-place" and several HRC tasks. Nevertheless, open issues for investigation still exist and constitute promising areas of future work. Specifically, we intend to extend our approach with additional relevant parameters, such as velocity, acceleration and force. Such a holistic approach to LfD may clearly support more complex HRC scenarios, such as cases where action timing plays a

role in human-robot cooperation. The latter scenarios, augmented with the presence of additional agents (humans and/or robots), constitute interesting research directions.

Chapter 4

Time-informed Action Learning and Reproduction

Interestingly, the topic of speed adaptation of executed actions has received rather limited attention. In [4] the high level temporal alignment of demonstrated actions is used to guide trajectory generation in the actor space. In addition, in [157] actions learned through a slow demonstration procedure are gradually self-improved to accomplish speed adaptation in action execution. Furthermore, to the best of our knowledge, LfD and associated latent representations that compactly describe spatio-temporal features have not been extensively studied to-date in a temporal context. However, indicative works, such as those of Calinon et al. in [4] and in [158] have set the basis, by formulating the well known Dynamical Movement Primitives (DMPs), learning time and space constraints during a task. Additional works [159], [160], [130], [161], by Ijspeert et al., Rozo et al. and Ewertoneta et al. have also investigated the implementation of DMPs within multiple conditions to reproduce force-based manipulation tasks and learn reactive and proactive behaviors in Human-Robot Collaboration.

Execution of learned actions is in many cases affected by temporal constraints that are externally imposed [162]. Processing of such constraints in planning problems has been typically based on Simple Temporal Networks, which are mapped on the equivalent Distance Graphs, to verify the existence of no negative cycles and thus prove the consistency and dispatchability of the plan [163]. Along this line, recent works have considered back propagation rules to dynamically preserve dispatchability of plans [164], as the implementation of the plan proceeds and time constraints are updated. However, these works focus on specifying the start moment of a given action in relation to the others, without considering the role of speed adaptation on action synchronization and temporal constraint satisfaction. To address this issue, we have recently proposed an interval calculus approach to estimate the expected latency of actions which directs informed adaptations on action implementation [22]. The present work aims at combining the estimation of expected latency and the concise specification of a revised execution time, with the ability to learn by

observation actions demonstrated at different speeds, in order to implement an enhanced composite system that is capable to effectively comply with dynamically changing temporal constraints.

4.1 Spatio-Temporal Representation of Actions

In this Chapter, following the basic formulation introduced in Chapter 3, we extend the IMFO methodological framework [25] to incorporate action learning and representation in the spatio-temporal domain. The latter effectively facilitates action reproduction in a given scenario, and action execution at a speed different than the learned one. Action learning is accomplished via a latent representation of observed actions. Such a representation achieves to compactly depict pertinent action information and abstract from the actual kinematic configuration of a system, e.g. human demonstrator or robotic platform.

4.1.1 Learning Phase

A conceptual representation of the employed methodology is depicted in Figure 4.1. Henceforth, and without loss of generality, we consider only arm motions. The spatial representation of such actions is readily available from the trajectory of the parameters in the arm's configuration space. The latter parameters, that convey spatial information of the executed action, are augmented with timestamps that signify time of occurrence of the particular snapshot in the action trajectory. Effective transformation of the spatial (configuration space) and temporal (timestamp) parameters into a latent (compact) space establishes a unified spatio-temporal representation of the considered action, which enables robots to execute actions at different speeds. With the above representation in place, learning is accomplished by establishing an association across the two latent representations, that correspond to the human action and the robotic one. This is presented in detail in the following.

Data Acquisition

Let an arm motion trajectory performed by a human actor and the corresponding one by the robotic system. The former describes a trajectory in a 11D configuration space, that is 3 joints of 3 coordinates each, 1 grasp parameter and 1 timestamp value. This is termed as HAS (Human Action Space). The coordinates for the 3 joints in HAS are computed as the center points of relevant joints. For that a model based approach has been adopted, analogous to the approach followed in [143]. Similarly, we obtain an 8D representation for robot arm actions (6 arm-joints, 1 grasp parameter, 1 timestamp value), termed as RAS (Robot Action Space). The latter representation arises since we employ in this work the 6-joint-arm JACO, by Kinova Robotics. The grasp parameter introduced above assumes

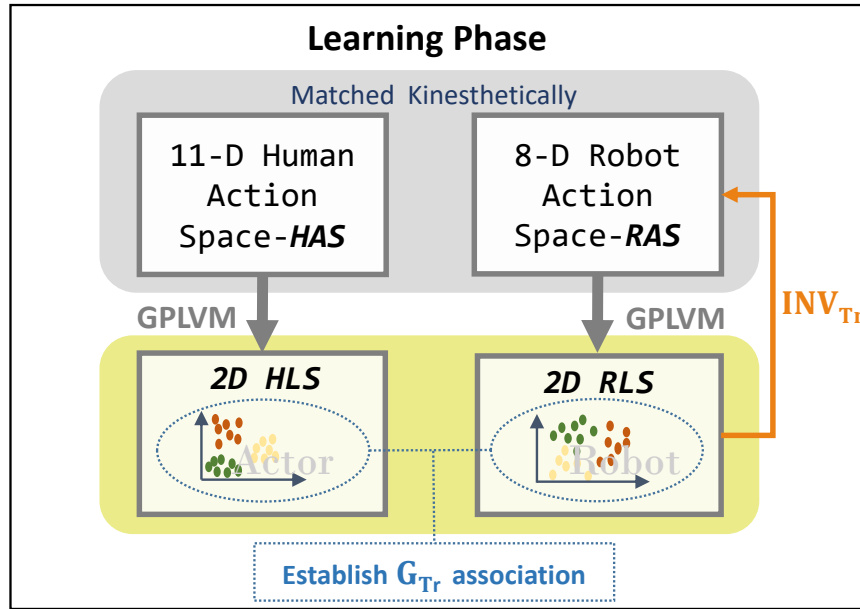


Figure 4.1: Schematic overview of the learning process.

values 0, 1 or 2, to represent the grasping configuration pose of the hand, as illustrated in Fig. 4.2. The correspondence of human and robot arm motions is achieved by adopting kinesthetic teaching. In other words, the human demonstrator performs several demonstrations that form trajectories in a rather high dimensional actor space (HAS). The same acts are performed by the robotic platform, physically steered by the human, formulating an equivalent high robot's space (RAS).



Figure 4.2: Grasping configuration poses; i.e. Grasping handle encoded as “0” (Left), “1” (Center) and “2” (Right).

Latent Space Representation

In order to facilitate meaningful association across HAS and RAS, both action spaces are transformed to analogous latent representations. To this end, we employ the Gaussian Process Latent Variable Model algorithm, GPLVM in short [144], as formulated in [25]; an algorithmic depiction of GPLVM is outlined in Alg. 1 template of Chapter 3.

As it previously referred, GPLVM effectively provides an accurate representation of an abstract multidimensional space projected to a lower dimensional one, by performing a non-linear dimensionality reduction in the context of Gaussian processes. Hence, two latent space representations are established. The “Human Latent Space” (HLS) is derived as a set of points $X_{HLS} \in \mathbb{R}^q$ and the “Robot Latent Space” (RLS) as $X_{RLS} \in \mathbb{R}^q$. Using maximum-likelihood estimation methodology [145], the optimal dimension of HLS and RLS is determined as $q = 2$. The latter has been also experimentally verified.

Space Association

Having established the two latent spaces, an appropriate geometric transformation G_{Tr} , that matches the corresponding points across the derived latent representations HLS and RLS, is defined. Given two sets of corresponding 2D latent variables $\mathbf{X}_{HLS} = \{X_{H1}, \dots, X_{Hn}\} \in \mathbb{R}^2$ and $\mathbf{X}_{RLS} = \{X_{R1}, \dots, X_{Rn}\} \in \mathbb{R}^2$, the transformation is derived as a pair of translation t and rotation r that minimizes the sum of squared error:

$$E(t, r) = \frac{1}{T} \sum_{i=1}^n \|X_{Hi} - rX_{Ri} - t\|^2 \quad (4.1)$$

where X_{Hi}, X_{Ri} are corresponding points. The correspondence between the pairings is ensured during the data acquisition, where the trajectories are captured by kinesthetic teaching. The Iterative Closest Point (ICP) algorithm [146] is employed to minimize eq. (4.1). The resulting transformation pair $G_{Tr}(t, r)$ can subsequently be used to map any point from HLS to the corresponding point in RLS.

Inverse Transformation

In order to enable the robotic arm to reproduce a demonstrated human action, it is necessary to transform the RLS representation to the robotic physical configuration space, namely RAS. Formally speaking, this association should be implemented as the inverse of GPLVM, which is not analytically available. To approximate this inverse transformation (marked as INV_{Tr}), a high order interpolation of the training learned data is formulated, as presented in Section 3.1.2. Here, this is addressed by computing off-line the latent representations (RLS) of a sufficiently large population of physical configurations (RAS) of the

robotic arm. In practice, we iterate over all arm-DoFs, and for each arm configuration in RAS, the representation in latent space (RLS) is obtained via the GPLVM.

The above iterations are performed by (a) respecting the physical limits of each arm-joint, and (b) employing an appropriate iteration step. A small step value results in a denser representation of the RAS-RLS pairings. For points in RLS that are not included in the above pre-computed pairs, the corresponding points in RAS are derived by interpolation. Experimentally it has been established that neither the step-value nor the actual method of interpolation were critical. This is due to the fact that the employed inverse transformation should not render an accurate (exact) replica of a demonstrated act, but rather a reproduced robotic behavior sufficiently similar to the latter.

Consequently, learning concludes with the two latent space representations ($HAS \rightarrow HLS$ and $RAS \rightarrow RLS$) along with the geometric transformation G_{T_r} that associates them ($HLS \leftrightarrow RLS$) and the established inverse mapping INV_{T_r} ($RLS \rightarrow RAS$). As already mentioned above, Fig. 4.1 schematically presents the learning process.

4.1.2 Speed Inference Based on Temporal Information

A variety of issues may affect the temporal aspects of task execution in real world applications. These may regard, for example, the physical properties of interacting objects (e.g. slow down to move a glass of water) or, the need to synchronize with real world temporal constraints or ongoing procedures (e.g. speed up to open the door after a bell-ring). However, time is a parameter that, so far, has been rather rarely considered in robot action planning.

Recently we introduced the Daisy Planner, in an attempt to address time-informed planning in multi-agent setups [21]. Below, we summarize the most relevant issues which are essential for the completeness of the present work. The planner uses a fuzzy number representation of time to enable the processing of temporal information and develop time-informed multi-criteria optimized plans. More specifically, each time interval is mapped on a trapezoidal fuzzy number that is represented by the quadruple (p, m, n, q) . For example, following this formulation, Fig. 4.3 depicts the time interval “approximately 6 to 9 moments” represented as the fuzzy number $T_B = (3, 6, 9, 10)$. In the same figure, negative values represent past time moments.

The present work exploits the fuzzy number representation of time intervals in order to (i) monitor the temporal aspects of composite task implementation, (ii) adapt the execution of individual primitive actions, and (iii) enforce the timely implementation of the overall task. In particular, using the ordinary fuzzy calculus, it is possible to associate the temporal properties of individual actions, predict delays and then take corrective measures that account for speed adaptations, in order to ensure that composite behaviors are implemented on time.

As an illustrative example, we consider the case of a composite task that is based on the sequential implementation of actions A, B and C. The prior knowledge of the system on the completion time of the three actions is as follows $T_A = (3, 4, 7, 9)$, $T_B = (3, 6, 9, 10)$, $T_C = (5, 6, 9, 11)$. Using the well known L-R calculus [165], the total implementation time for the three actions is estimated as: $T_{total} = (11, 16, 25, 30)$. To clarify the concept, let a temporal constraint in the problem formulation, which requires that the three actions should complete at a specific maximum time of $C_{max} = 20$. The system monitors the sequential execution of actions to ensure the timely accomplishment of the composite task. Assuming that 7 moments have been devoted to the implementation of action A, then B and C should be implemented at a maximum time of $C_{max} = 13$ moments. This is represented by the fuzzy time interval $(0, 0, 13, 13)$.

In order to find a safe completion time for action B, the time to be spend on C is subtracted from the maximum available time. According to the L-R fuzzy calculus, this results into $A_B = (0, 0, 13, 13) - (5, 6, 9, 11) = (-11, -9, 7, 8)$, which is the time available for B. We take the intersection of the estimated available time A_B and the actual, known by experience, implementation time T_B to estimate the available temporal flexibility for the execution of B. The intersection of A_B and T_B is calculated as shown in Fig. 4.3. The defuzzification of this interval (implemented by the classic graded mean integration representation [166]) results into the requested action B implementation time, that is $t = 6.2$.

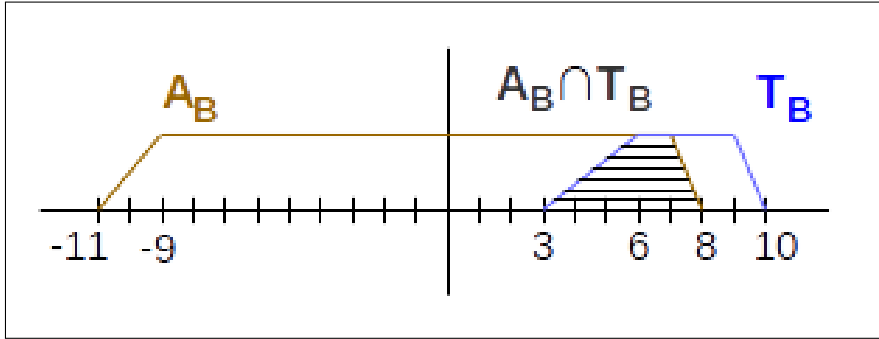


Figure 4.3: Graphical illustration of the intersection of two fuzzy intervals.

4.1.3 Time-informed Robotic Action Execution

The present work considers tasks implemented as a sequence of three actions, namely reach, grasp, and move/place. To ensure that the composite task is implemented on time, the system monitors the execution of each action, estimates possible delays as described above and suggests a completion time for the action to be implemented next. The requested time is used to create an implementation of a known act to within the provided time interval. The obtained, time-modulated human action in HAS is represented through

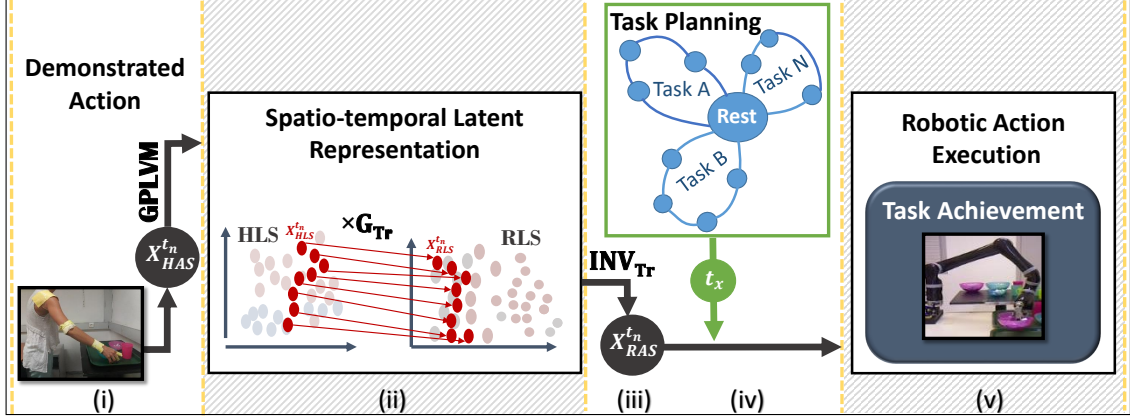


Figure 4.4: Block diagram representation of time-informed robotic action execution.

GPLVM compression to HLS. The latter is in turn transferred to the robot's space by means of the learned HLS to RLS mapping, and subsequently unfolded to RAS. The robot implements the action within the requested time limits by implementing the sequence of configurations encoded in RAS.

In particular, the framework comprises off-line (action learning) as well as on-line (action execution) modules. Accordingly, the following steps, summarized also in Fig. 4.4, are employed to effectively reproduce an arm motion by the robotic system:

- (i) A demonstrated action, presented as a sequence of points in HAS, namely $X_{HAS}^{t_n} = \{x_1, x_2, \dots, x_n\}$, is mapped to the corresponding one in HLS; GPLVM transforms $X_{HAS}^{t_n}$ to $X_{HLS}^{t_n}$.
- (ii) Mapping to robot latent space: the $G_{Tr}(t, r)$ mapping is used to obtain the relevant RLS representation, X_{RLS} .
- (iii) Action reproduction: the formulated inverse transformation INV_{Tr} maps the X_{RLS} to RAS, formulating a set of points $X_{RAS} \in \mathbb{R}^8$.
- (iv) The planer's output specifies the action to be implemented (i.e. reach, grasp, move), and the requested execution times. The latter determines the speed of action execution i.e. low, normal, high.
- (v) Action is reproduced by the robot at the appropriate speed. Successful completion of the action is visually verified by a human observer (e.g. successful grasp of an object). In case of unsuccessful robotic action, the session is dropped.

4.2 Experimental Results

The methodological framework presented in the previous sections of this Chapter has been implemented and exhaustively tested in order to assess its performance in realistic cases. In this section, we report relevant results that quantify the effectiveness of action reproduction at the learned or different execution speeds.

In the current study the relevant experimental set-up regarding the learning phase involved kinesthetic teaching in order to demonstrate certain action behaviors to the robotic arm. In all our experiments a six-joint arm manipulator was used, namely the JACO robotic arm, by Kinova Robotics. Its joints can be controlled independently either with position-control or with torque-control. Accordingly, the compliant mode of the JACO arm was used in order to physically steer the arm to execute various motion trajectories.

The learning set is composed by sequences of paired poses executed by the two agents, i.e. actor human arm and JACO robot arm, respectively. More specifically, we recorded a set of 30 primitive right-arm-movements of reaching, grasping and placing an object of 100 frames each. Each demonstrated action is performed either at low, normal, or high speed. Given that for a human actor it is rather unrealistic to exactly perform actions at certain speeds, we employed the following convention in our experiments. At first, in an off-line session, the human demonstrator performed repeatedly the 30 actions in order to train with respect to the execution speeds. This resulted at a consistency in the execution times within a 20% margin. In other words, a sample reach/grasp/move action at normal speed, after training would have a duration of $10 \pm 2\text{sec}$. Similarly, durations at high-to-low speed were performed on the average at $5 - \text{to} - 16\text{sec}$, respectively. Subsequently, we used a threshold-based categorization to assign learned actions as low, normal or high speed ones. The corresponding thresholds were set at durations of (a) 7sec to discriminate between high and normal speed actions, and (b) 14sec to distinguish among normal and low speed actions. In order to end up with an un-biased population of the demonstrated actions, out of the 30 recorded actions $1/3$ were executed at each of the three specified execution speeds.

Demonstration of the above outlined 30 actions, and concurrent kinesthetic teaching to the JACO arm, gave rise to the establishment of the HAS and RAS spaces that were instantiated with the recorded samples from the actions' configuration spaces. More specifically, both HAS and RAS assumed 3000 configuration poses (samples) at the end of the training phase (30 actions \times 100 frames each). Following the learning phase, as described above, GPLVM was employed in order to compress the high dimensional acting spaces to the latent ones. As a result, HAS and RAS point clouds have been transformed to HLS and RLS representations, respectively.

Figure 4.5 illustrates the resulting latent spaces. It is of utmost importance that the derived representations in HLS and RLS reveal well-separated sectors that correspond to the

three implemented speeds. In other words, the sole information of action execution speed seems adequate to classify the action representations in the latent space in practically non-overlapping clusters. Interestingly, the latter holds true for both derived latent representations, namely HLS and RLS. This result should come as no surprise since the information conveyed by speed is quite discriminative, at least as opposed to spatial information that for the studied human actions designates small differences in the recorded trajectories. Furthermore, each point in the latent space represents an 11D vector in the acting space of human and an 8D vector for the robot case. In other words, every latent point includes the information of the human or the robot state, the grasping configuration and the timestamp. To this end, it is worth noting that the formulated latent spaces feature important and at the same time useful properties. Different actions result in well separated trajectories in the latent space, and also the neighboring property of a 3d trajectory is maintained given the continuous curves that are derived in the latent space. Furthermore, actions at different speeds are depicted in distinct point clouds in the latent space.

Besides the rather straightforward classification of actions in the latent spaces as exemplified above, additional noteworthy remarks can also be drawn from Fig. 4.5. More specifically, for each action in HAS (and its corresponding in RAS) recorded at a certain speed, we have “mentally” derived the two analogous actions in the other two implemented speeds. As a result, the three actions instantiated the same spatial actions at the three speeds, i.e. low, normal and high. Subsequent derivation of the latent representations gave rise to motion paths as the ones illustrated in Fig. 4.5 with the marked trajectories.

As can be observed, actions in the latent spaces are characterised by spatial continuity, that is neighbouring points in the acting space result in neighbouring points in the latent space. In other words, the physical continuity of an action’s 3D trajectory is also maintained in the derived latent representation. This is in full accordance with the findings in our previous work [25], where we considered the latent representations of actions at a single execution speed.

4.2.1 Performance Metrics of Spatio-temporal Accuracy

In order to quantitatively assess the proposed spatio-temporal LfD methodological framework, we adopt in this work similar evaluation metrics as in [25], i.e. HLS and RLS consistency and repeatability (E_{HLS} , E_{RLS}), robotic end-effector trajectory (E_{EF}), and temporal consistency (E_t). To this end, let a novel action X^N performed from the human demonstrator at normal speed. From X^N , the spatially equivalent actions at low (X^L) and high (X^H) speed are artificially generated. Additionally, from each of the three actions X^N , X^L and X^H , a set of M actions $\mathbf{X}^{i,N}$, $\mathbf{X}^{i,L}$, $\mathbf{X}^{i,H}$, $i = 1, \dots, M$ is produced by keeping the temporal components of the actions unaltered and randomly perturbing the spatial ones. Accordingly, M action trajectories are produced for each speed category that are variations of the same

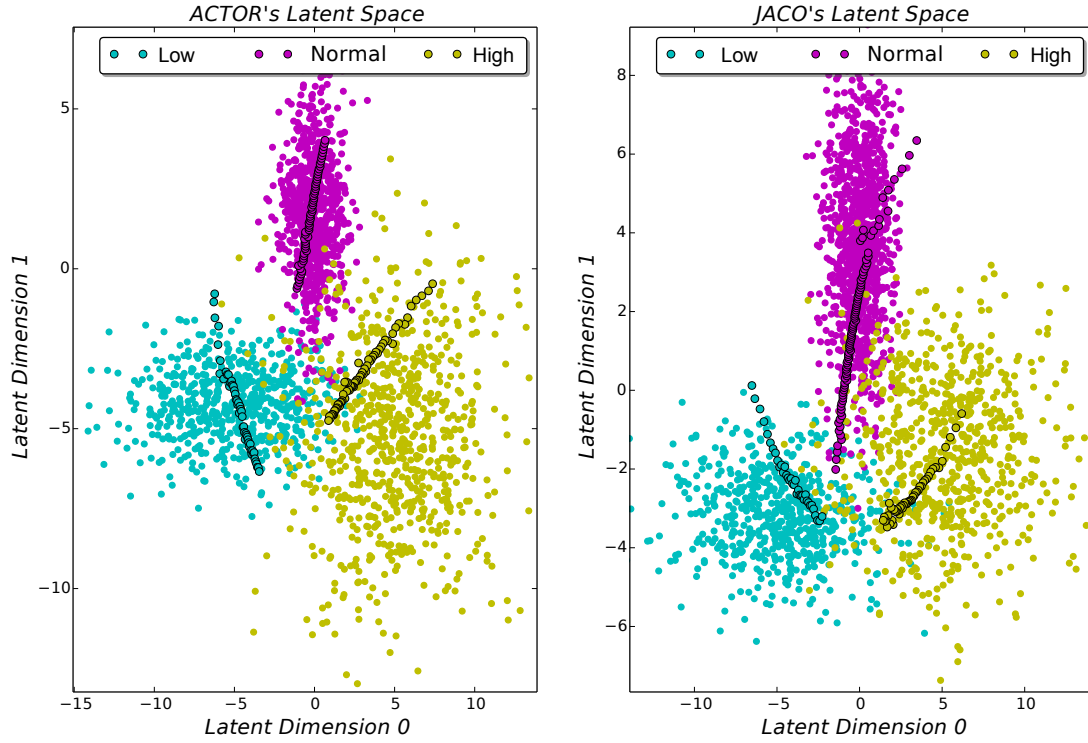


Figure 4.5: Formulated latent space representations for human actor (left) and JACO arm (right), respectively. Actions performed at normal speed are represented by magenta circles, whereas slow and fast actions (in low and high speed respectively) are depicted with cyan and yellow circles, respectively.

initial human action; a value of $M = 50$ has been used in our experiments. Subsequently, each one is projected via GPLVM to HLS, and G_{T_r} is then employed to map the latter representations to RLS. Finally INV_{T_r} is invoked to inversely transform the trajectories to RAS, effectively resulting to the motions being reproduced by the robotic arm along with their execution times.

HLS Consistency and Repeatability - E_{HLS}

Let the three sets of actions in HLS, depicted from sets of low, normal, and high speed, respectively, $\mathbf{X}_{HLS}^{i,L}$, $\mathbf{X}_{HLS}^{i,N}$, $\mathbf{X}_{HLS}^{i,H}$, $i = 1, \dots, M$. Let also $\bar{\mathbf{X}}_{HLS}^{i,L}$, $\bar{\mathbf{X}}_{HLS}^{i,N}$, $\bar{\mathbf{X}}_{HLS}^{i,H}$ be their corresponding mean trajectories. E_{HLS} is defined as:

$$E_{HLS} = \mathbb{E} \left[\frac{1}{M} \sum_{i=1}^M \left\| \mathbf{X}_{HLS}^{i,K} - \bar{\mathbf{X}}_{HLS}^{i,K} \right\|_{\mathbf{S}^{-1}}^2 \middle|_{\{K=L,N,H\}} \right] \quad (4.2)$$

where distances in above eq. 4.2 are Mahalanobis distances expressed in the corresponding ellipsoids of HLS (Fig. 4.5 left). $\mathbb{E}[\cdot]$ denotes the mean value of the three quantities that result for K values marked as N, L, H .

RLS Consistency and Repeatability - E_{RLS}

In a similar manner as above, E_{RLS} is obtained as the relevant sum of Mahalanobis distances expressed in RLS:

$$E_{RLS} = \mathbb{E} \left[\frac{1}{M} \sum_{i=1}^M \left\| \mathbf{X}_{RLS}^{i,K} - \bar{\mathbf{X}}_{RLS}^{i,K} \right\|_{\mathbf{S}^{-1}}^2 \middle|_{\{K=L,N,H\}} \right] \quad (4.3)$$

where symbols in eq. 4.3 above are interpreted as in eq. 4.2.

Robotic end-effector trajectory - E_{EF}

A Euclidean distance metric is calculated as an index of imitation of the end-effector's movement. More specifically, E_{EF} is assumed to quantify the precise reproduction of a demonstrated act by the robotic end-effector, and it is defined as the 3D-error in the latter's trajectory. By this metric, only the spatial information is isolated, as the execution time is not considered in the relevant equation. In other words, this measure computes the Euclidean (3D) differences in trajectories between the robotic action reproduction and the demonstrated one, assuming the same execution times; it is obtained as:

$$E_{EF} = \mathbb{E} \left[\frac{1}{M} \sum_{i=1}^M \left\| \mathbf{X}_{H_{EF}}^{i,K} - \mathbf{X}_{R_{EF}}^{i,K} \right\|^2 \middle|_{\{K=L,N,H\}} \right] \quad (4.4)$$

where $\mathbf{X}_{H_{EF}}^{i,K}, \mathbf{X}_{R_{EF}}^{i,K}$ indicate the end effector trajectories of the human actor and the robotic arm, respectively, at K speed, i.e. low, normal, and high.

Temporal consistency - E_t

A final metric is evaluated to describe the differences rendered in the execution times. More specifically, let $t_{HAS}^{i,N}$ be the normal execution time of human action $\mathbf{X}_{HAS}^{i,N}$ and similarly for the execution times at low and high speeds. Let also $t_{RAS}^{i,N}$ be the execution time of the robotic reproduced action $\mathbf{X}_{RAS}^{i,N}$, with similar definitions again for reproduced actions at low and high speeds. The differences in the execution times averaged over all actions at a certain speed give rise to E_t as follows:

$$E_t = \mathbb{E} \left[\frac{1}{M} \sum_{i=1}^M \left\| t_{HAS}^{i,K} - t_{RAS}^{i,K} \right\|^2 \right]_{\{K=L,N,H\}} \quad (4.5)$$

The complete set of thirty demonstrated actions was employed for assessment purposes.

The values obtained in this case, for the above described performance metrics, are summarized below in Table 4.1 as percentage figures. Following common practice in the area [4], [25], mean values of the obtained errors are presented, along with the relevant standard deviations.

Table 4.1: Mean error values (μ) and standard deviation (σ) of the computed metrics for each group of speed, namely Low (L), Normal (N) and High (H)

Metrics	Low (L)		Normal (N)		High (H)	
	μ	σ	μ	σ	μ	σ
E_{HLS}	2.7 %	0.1%	3.2%	0.1%	3.4%	0.2%
E_{RLS}	2.9 %	0.1%	4.6%	0.3%	5.1%	0.2%
E_{EF}	2.7 %	0.2%	3.7%	0.3%	4.4%	0.3%
E_t	2.4 %	0.1%	2.8%	0.3%	3.2%	0.3%

As can be observed, all performance metrics assumed very low values, indicating accurate and effective spatio-temporal representation and reproduction of actions. Interestingly, for the Low (L) case the errors are slightly smaller, becoming gradually larger as speed increases. This is rather expected, since lower speed values give rise to smoother trajectories, which in turn facilitate more accurate reproduction of relevant actions.

4.2.2 Use Case Application - Service Scenario

The second category of experiments regards the validation of robotic performance in a realistic service scenario. In particular, the available reaching, grasp, move actions are exploited to address time-constrained tray filling. The examined scenario is inspired by restaurant standing queues with customers served one at a time. Robot aims at filling the tray within the requested time frame, in order to serve humans on time. Humans are supposed to wait in front of the serving queue. The simplified serving considered here assumes two cups and one bowl to be placed on each tray. We consider varying times of requested tray filling, centered at 2.0 minutes (the average period of customer arrival). In short, when serving a customer is delayed, the system tries to compensate this latency by asking for faster filling of future trays. Following this formulation/scenario, the repetitive tray filling task must be implemented at varying time limits and hence robot action speeds.

We have developed a simple setup that enables tray filling in naturalistic conditions (see Fig. 4.6). It is noted here, that the proposed approach and the Daisy Planner in particular, manage the temporal constraints on tasks by enabling both the increment and the decrement of speed. In general both options are required and the proposed work provides a systematic approach to encompass both speed adaptations, and also demonstrates the latter in real application scenarios. To this end, a service scenario application case is used to demonstrate the validity of the approach, showing at the same time its ability of generalization and natural flow.

Additionally, for quantitative assessment of the action execution times, we conducted 20 repetitions of the tray filling task; cases where grasp failures were encountered were dropped out and the experiment was repeated. Accordingly, we ended up with 20 successful task completions for which we contrasted the actual execution times against the commanded ones by the planner. Time differences above 10% were regarded as failures. Interestingly, only 3 executions did not meet the latter criterion, and were marked unsuccessful. Given the complexity and variability of the studied scenario, the accomplished result is considered highly promising and indicative of the method's potential. A better appreciation of the described experiments can be acquired by the supplementary video in high resolution at <https://youtu.be/WGG0vI6NiMU>.

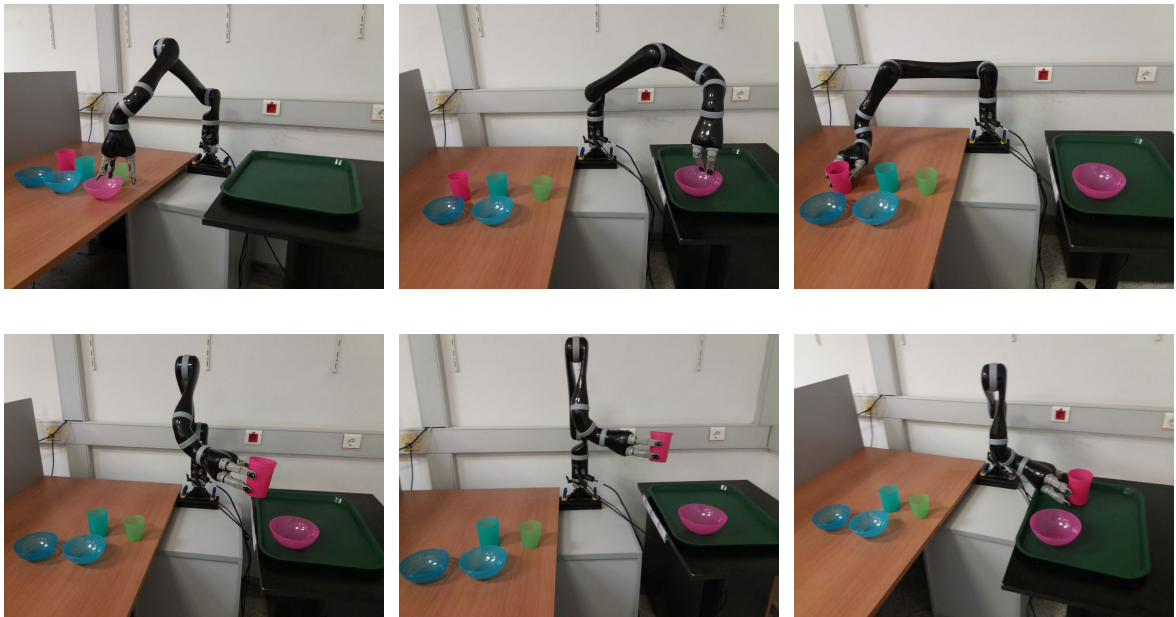


Figure 4.6: Snapshots from the serving scenario.

4.3 Chapter Highlights

In the current Chapter we introduced a novel spatio-temporal formulation that compactly represents spatial and temporal aspects of studied actions. The latter is accomplished by assuming latent space representations and further combined with a time-informed task planner to effectively schedule actions in the course of a complete task. Our examination has revealed very useful and interesting properties of the formulated latent representations, namely (i) well-separated latent representations for actions executed at different speeds, (ii) neighboring points in the configuration space are mapped to neighboring points in the latent space, thereby preserving action continuity in the latent depiction, and (iii) errors in the spatial and temporal domains due to the latent transformation are very small and in practice do not affect the method's performance.

The described framework has been employed in the execution of a realistic service scenario. Our results demonstrate the successful involvement of the robotic system in the accomplishment of relevant tasks, whereby learned action behaviors are appropriately executed at varying speeds. The main contributions of the proposed method are summarized as follows:

- Enhance robots that learn from demonstration to execute actions at variable speeds;
- Provide insight to the key role of temporal information in obtaining compact, latent representations of human arm motions;
- Demonstrate the reversibility of the spatio-temporal aspects of actions from the low dimensional space to the actual action space;
- Combine temporal planning with LfD.

Having dealt with the additional parameter of time, a natural next step is to extend this methodological framework in the case of multi-robot collaborative systems, where time-constrained task fulfillment is a critical issue. In addition the method's extensibility by incorporating further dynamic parameters, such as force, will be further examined in the following Chapter, aiming at more challenging tasks as in the case of lifting and manipulation of fragile and sensitive objects.

Chapter 5

Force-based Object Manipulation

Machine Learning methods has been widely employed for robotic reproduction of human behaviors [119], [167], [136] with corresponding implementations in various application domains. In our recent works [25], [24] we developed and evaluated an imitation framework, termed IMFO (IMitation Framework by Observation), that is based on the compact, low-dimensional representation of both human and robot arm motions, which are properly associated to facilitate learning of movements in various reproduction speeds. Similar approaches have also been studied in [4], [168], [169], [157], [12].

Industrial applications often depend on robots that need to manipulate fragile objects, in a precise and sensitive way. Such manipulations require a suitable force to be applied to the object; that is, the grasping force needs to be sufficiently large to lift the object and at the same time it cannot be too large to avoid object deformations and/or destruction. Accordingly, here, we study the more involved case of force-based object manipulation, that is the case where the suitable required force to lift and manipulate a delicate object varies for different objects and object positions relative to the arm. More specifically, we present a Supervised Learning scheme, termed SLF (Supervised Learning scheme for Force-based manipulation), to learn both the trajectory and the suitable gripper's force for rigorous manipulation of such objects.

In the proposed scheme, data acquisition results in a high dimensional dataset of an adequate number of lift and manipulation actions (data points), recorded in a GAZEBO simulated environment that facilitates ground truth availability. During learning, two alternative neural network topologies are employed and compared, a 3-layer Deep Neural Network (DNN) and a Recurrent Neural Network (RNN), to effectively derive the input-output relationship. The mechanism by which the two types of NNs represent the learning process is different. In the case of DNN, a feed forward neural network is implemented, whereas in the RNN topology the hidden layers are fed by the previous step as an additional input into the next step. While the RNN builds up memory in this process, it is not looking for the same patterns over successive time steps. The latter happens to be the case for a DNN that is looking for the same patterns over different spatial regions. To this end,

we further investigate whether the temporal dependence of the recorded trajectories, that is encoded as memory in the RNN topology, strengthens the learning process.

The proposed SLF accomplishes to generalize with respect to the suitable force required to be applied on a delicate object for effective lifting and manipulation. The latter constitutes the main contribution of the current work. Moreover, our approach achieves to formulate a generalized trajectory to a target position, and reproduce at the same time a successful pick-and-place movement. The above are successfully demonstrated in simulated and real experimental sessions conducted with a YuMi robotic arm (Fig. 5.1).

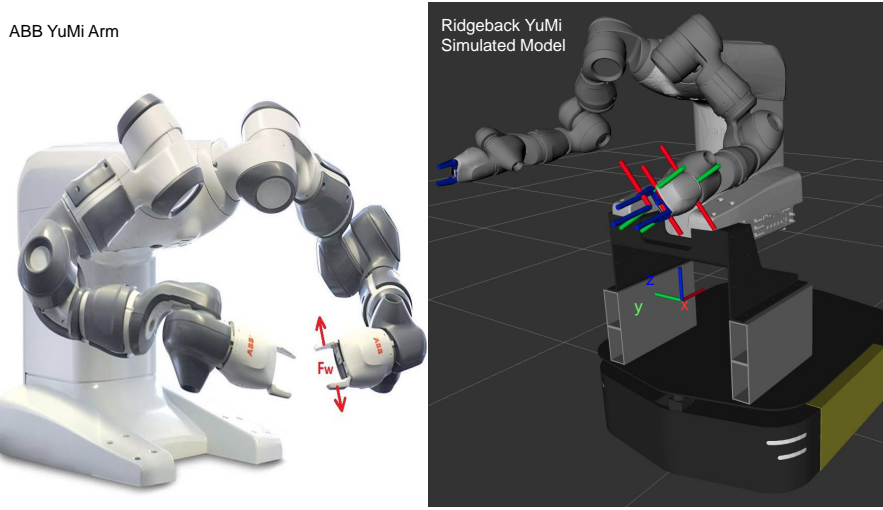


Figure 5.1: **Left:** ABB YuMi Arm. **Right:** Ridgeback-YuMi robot model in Gazebo simulator. Red-Green-Blue lines denote the XYZ axes, respectively, for the world, the left arm's end-effector and the wrist.

5.1 SLF Formulation

In the current work, we put forward a learning scheme (SLF), that is used to derive the grasping pose and the lifting force in order to manipulate delegate objects. Conceptually, the proposed formulation is comprised by three fundamental phases, as depicted in Fig. 5.2: (i) data acquisition phase via supervised trial-execution in simulation, (ii) training phase using a proper Neural Network topology, and (iii) execution phase both in simulation and real environment. Accordingly, a robotic system is initially set to execute grasping actions in simulation, applied to objects that vary in size, mass and location in workspace. In the latter actions the grasping pose is dictated by the supervisor, whereas the proper lifting force is obtained in a trial-and-error fashion, starting from a low force-value and gradually increasing it. Subsequently, training is facilitated via the employment of ap-

appropriate Neural Network (NN) structures. Two alternative NNs are exploited, both with their own merits. Specifically, Deep Neural Networks (DNN) and Recurrent Neural Networks (RNN) are used to facilitate the learning process and generate sequences of commands, directly fed to a low-level control loop. DNNs are known structures that succeed in encoding the spatial characteristics of the recorded movements, whereas RNNs better handle the temporal information as an inherent memory feature. The trained system accomplishes object-manipulation via proper grasping pose and lifting force, proving thus suitable for the case of sensitive objects. Following sim-to-real transfer, operation in real environments is achieved in addition to simulated ones, generalizing also for objects not included in the trial sessions.

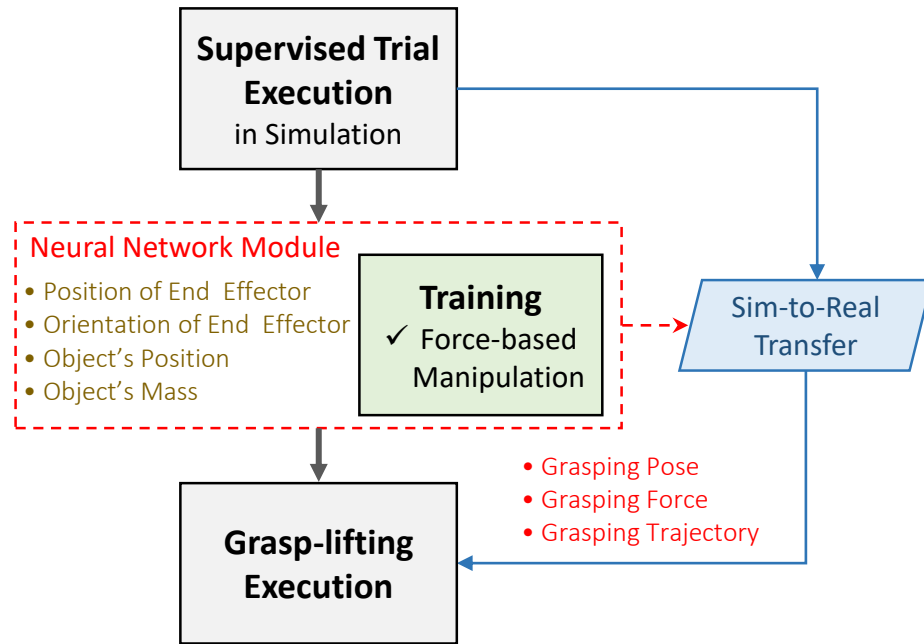


Figure 5.2: Conceptual scheme of the proposed SLF.

5.1.1 Supervised Trial Execution in Simulation

In order to facilitate training, a large dataset is collected that serves as input to the learning process. For data acquisition, the YuMi robot is set to perform several pick-and-place movements in the GAZEBO simulated environment, i.e. *reach the target, grasp it, and lift/place down*. The zero velocity crossing algorithm [170] is employed in order to segment an arm movement into the three distinct actions.

Trial sessions differ in the grasping configuration pose with respect to the mass of the object. Figure 5.3 illustrates the three grasping poses used during learning for each class of objects. All the grasping poses respect the kinematic constraints of the arm, i.e. the YuMi

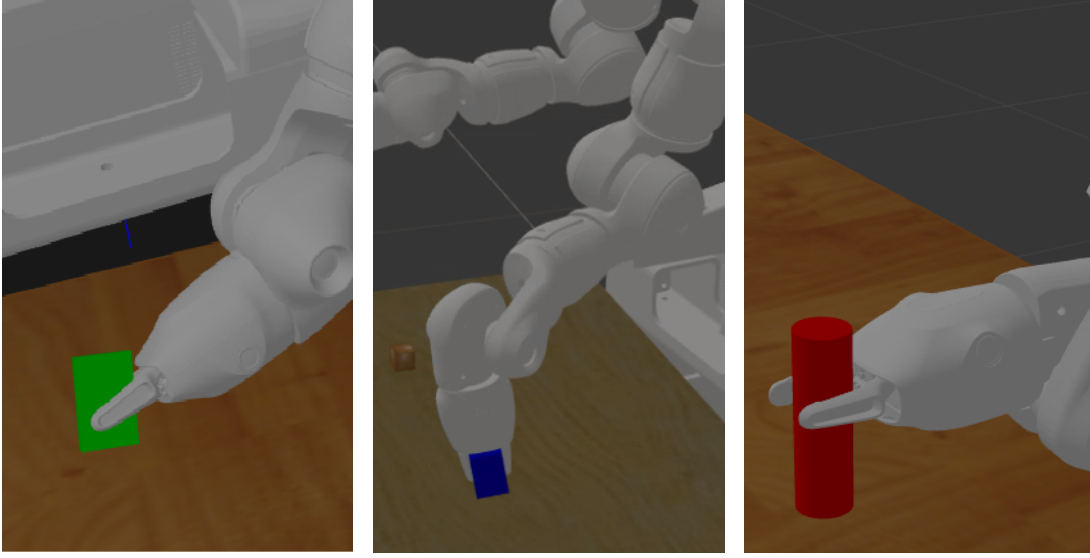


Figure 5.3: **Left:** Configuration grasping poses for green objects, marked as “light” grasping *Pose#1*. **Middle:** the “normal” grasping *Pose#2*. **Right:** the “heavy” one *Pose#3*.

arm needs to apply different forces to lift the same object (mass), when the arm is fully extended or not. As already mentioned above, during training the required lifting force is obtained via trial-and-error, starting from a low force-value and gradually increasing it. The grasping poses are selected by imitating the demonstrator’s real grasping pose while grasping a light, normal or heavy object.

Further to that, we note that the YuMi arm can lift objects in the range of $(0, 0.5] \text{kg}$. Accordingly, three groups of objects are used for training, as *Class* : $\{\text{Object Color, Mass, Grasping Pose}\}$:

- *Light* : $\{\text{Green, } (0.00, 0.16] \text{kg, Pose\#1}\}$
- *Normal* : $\{\text{Blue, } (0.16, 0.33] \text{kg, Pose\#2}\}$
- *Heavy* : $\{\text{Red, } (0.33, 0.50] \text{kg, Pose\#3}\}$

Given that the appropriate force in order to successfully lift an object depends on the object’s weight and the friction coefficient μ_o , we have employed in the current study a number of objects with similar friction coefficients, i.e. approximately $\simeq 0.4$. Accordingly, the network is trained to estimate the force based on the object’s weight, factoring out μ_o . Evidently, objects with different values of μ_o could have been considered with appropriate modifications in the network topology. This has been pursued in the current work, given experimental inadequacy to input in real-time μ_o to the neural network. However, estimating the friction coefficient is often not trivial. To this end, our approach is independent of

this estimation step by considering friction implicitly.

Completion of the trial sessions results in the compilation of the training set; the latter consists of N trajectories of T frames each. For each training data point, a 19-dimensional vector of the XYZ-position (i.e. $x_{ee_t}^n, y_{ee_t}^n, z_{ee_t}^n$) and the quaternion XYZW-rotation of the robot's arm end-effector (i.e. $qx_{ee_t}^n, qy_{ee_t}^n, qz_{ee_t}^n, qw_{ee_t}^n$), the target-object position (namely x_t^n, y_t^n, z_t^n), its mass (m), the robot's joint angles ($q1_t^n, \dots, q7_t^n$) and the force attributes of the gripper, $F_{w_t}^n$, is recorded as:

$$\mathbf{D}_t^n = \{x_{ee_t}^n, y_{ee_t}^n, z_{ee_t}^n, qx_{ee_t}^n, qy_{ee_t}^n, qz_{ee_t}^n, qw_{ee_t}^n, x_t^n, y_t^n, z_t^n, m, q1_t^n, q2_t^n, q3_t^n, q4_t^n, q5_t^n, q6_t^n, q7_t^n, F_{w_t}^n\}$$

where $t \in [1, T]$ refers to the time frame of the movement and $n \in [1, N]$ is the n^{th} trial session of the total N ones.

5.1.2 Training to Facilitate Force-based Manipulation

In this part of the proposed SLF, two alternative approaches are implemented and tested for their accuracy and validity for the learning process. More specifically, we investigate the role of (a) pattern similarities over different spatial regions, and (b) temporal dependency of the recorded sequences of poses (trajectories). The former is accomplished via a 3-layer Deep Neural Network (DNN) implementation, while a Recurrent Neural Network (RNN) is employed for the latter.

Irrespectively of the training process, the Neural Network (NN) module learns how to plan a task by repeating successful task sequences for multiple objects and also figures out a good grasp configuration choice among infinite possible ones in the joint space to effectively grasp and lift the given object.

Deep Neural Network: In the current case, training is accomplished via a DNN structure that is set to learn the grasping pose and lifting force of the industrial robot YuMi. DNN input is composed of 11 variables, namely robot's end-effector position and orientation, target position and mass $\{x_{ee}, y_{ee}, z_{ee}, qx_{ee}, qy_{ee}, qz_{ee}, qw_{ee}, x, y, z, m\}$ (target's orientation is considered fixed), whereas the DNN outputs 8 variables that denote the joint states of the 7-joint YuMi arm along with the force value of the gripper $\{q_1, q_2, q_3, q_4, q_5, q_6, q_7, F_w\}$. The joint angles $\{q_1 \dots q_7\}$ for the output data, are calculated by using a home-made Inverse Kinematics analytical solution for the YuMi model, based on available ROS KDL package [171]. Moreover, the force value F_w is the resultant force of the two fingers force components as they are highlighted in Fig 5.1 (left) with the red arrows. The number of the hidden layers and neurons are experimentally set to 3 and 7, respectively. The neurons in two adjacent layers are connected with weights randomly initialized within the range

$[-1, 1]$. During the learning process, the back propagation algorithm is applied for training the weights. Fig. 5.4 summarizes the overall structure of the proposed NN.

In turn, we minimize a cost function C , that is given as the mean squared error between the estimated and the actual values of the output variables:

$$C(\mathbf{W}_x^y) = \frac{1}{2} \|h_w(\mathbf{x}) - \mathbf{y}\|^2, \quad (5.1)$$

where, w is the set of the weights in the network to be trained, \mathbf{y} is the output label, and $h_w(\mathbf{x})$ is a hypothesis function yielding an estimated output. The overall cost function for a batch training is defined as:

$$C(\mathbf{W}) = \frac{1}{N} \sum_k C(\mathbf{W}_{x^k}^{y^k}) + \frac{\lambda}{2N} \sum_l \|\mathbf{W}\|_F^2 \quad (5.2)$$

where, N is the training set size, D is the depth of the NN, λ is a regularization parameter and the term $\|\mathbf{W}\|_F^2$ is the Frobenious norm, defined as:

$$\|\mathbf{W}\|_F^2 = \sum_{i=1}^{n_{l-1}} \sum_{j=1}^{n_l} (\mathbf{w}_{ij}^n)^2 \quad (5.3)$$

where, n is the number of the nodes at the l^{th} layer, and $\mathbf{w}_{ij}^n \in \mathbf{W}$ is the weight of the edges between a node i in the layer $l-1$ and a node j in the layer l . We want to obtain the optimal

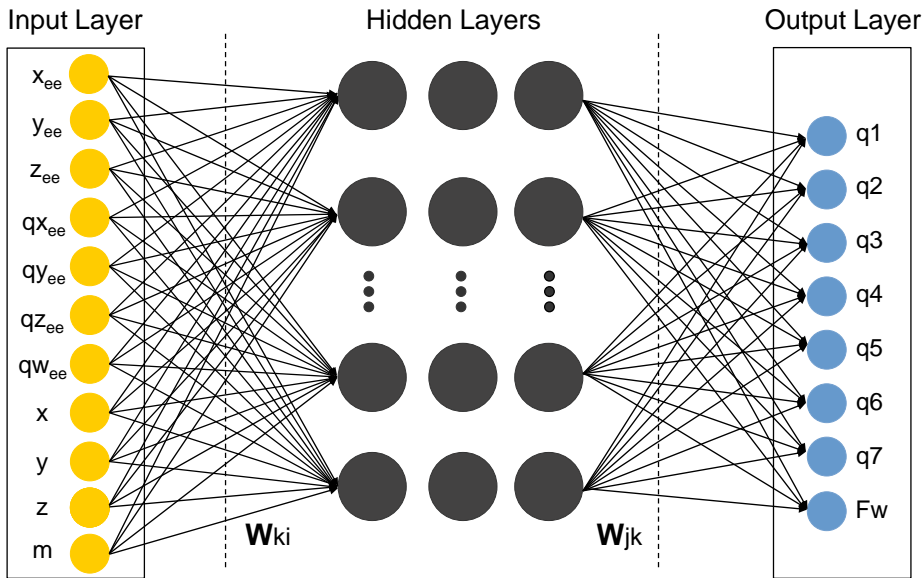


Figure 5.4: Block diagram of the proposed DNN topology.

parameter set \mathbf{W}^* to achieve the minimization of the objective function as follows:

$$\mathbf{W}^* = \arg \min_w C(\mathbf{w}) \quad (5.4)$$

which can be achieved by the back propagation algorithm. In the latter the weight vectors are updated from the top layer to the bottom one by using the stochastic gradient method:

$$\mathbf{w}_{ij}^n = \mathbf{w}_{ij}^{n-1} + \delta \frac{\partial C(\mathbf{W})}{\partial \mathbf{w}_{ij}^{n-1}} \quad (5.5)$$

where δ is an adaptation parameter, experimentally set to 0.5.

Recurrent Neural Network: Recurrent Neural Networks (RNNs) are quite popular models that have shown great promise in many learning tasks [172]. The main idea behind RNNs is the exploitation of sequential information. In a traditional NN, as described above, we assume that all inputs (and outputs) are independent of each other. RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on previous computations. In other words, they are supposed to have a “memory” that captures information about what has been calculated so far. In theory, RNNs can make use of information in arbitrarily long sequences, but in practice they are limited to looking back only a few steps.

In the proposed scheme, we train sequences of poses that have a temporal dependence among them. Subsequently, we employ a task-specific version of an RNN topology that is appropriate for SLF as shown in Fig. 5.5. In this implementation the inputs and outputs are the sequences of the demonstrated data, in full analogy with those used in the DNN formulation.

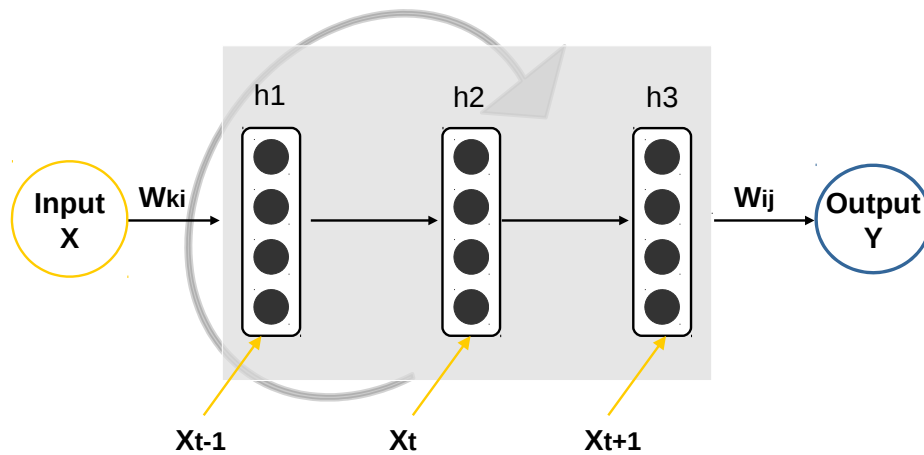


Figure 5.5: Schematic block diagram of the proposed RNN topology.

5.1.3 Grasp-lifting Execution

After learning the weight parameters, two types of executions are performed, both in simulation and in real robot using sim-to-real transfer. *Sim-to-Real Transfer Learning*, as it is often termed, offers a great potential to bridge the gap between simulation and real data in machine learning process control [173]. To this end, in the case of real robot we take advantage of the large number of trials that can be acquired through simulation. In turn, the following steps are employed to effectively reproduce a generalized lifting movement by the robotic system (Fig. 5.6).

- i An external camera system detects the position of the target object; the mass of the object is assumed known. In practice the latter is color-coded in our implementation.
- ii Through the trained NN module the output variables are predicted, e.g. the joint states of the arm and the desired lifting force.
- iii Finally, the arm executes the predicted trajectory to the goal pose and applies the required predicted force to successfully lift the object.

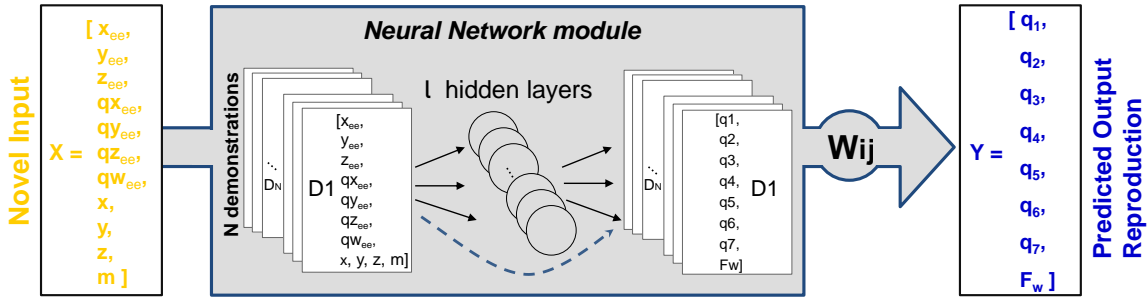


Figure 5.6: Integrated system design.

By this process, given a set of novel input data, we derive the corresponding output values that are appropriate for the task at hand. The described learning system achieves to: (i) implicitly implement a kinematic model of the YuMi arm, (ii) predict the output in high accuracy, achieving at the same time both trajectory and force predictive control, (iii) adapt and generalize for the case of novel objects that were not used during training. In the following section we present detailed experimental results that demonstrate the above.

5.2 Experimental Results

The proposed Supervised Learning scheme has been implemented and experimentally validated in both simulated and real robot environments. In the current section we present

at first quantitative results, that demonstrate the accuracy and robustness of the proposed force-based manipulation scheme in simulation. Further to that, evaluation results are outlined for the case of a YuMi robotic platform that is set to manipulate appropriate objects.

5.2.1 Simulation Experiments

In order to train the system and obtain quantitative assessment results, a large amount of data should be recorded. To this end, we assume the YuMi arm in a simulated Gazebo environment, executing a variety of pick-and-place movements.

To effectively accomplish training of the NN module, we employed two sets of data, one to train the module, and one to validate it. The complete dataset is comprised of 50 arm-motion trajectories for each class of objects, of 1000 data points each, hence in total 150,000 data points constitute the simulated data set. Every recorded trajectory is composed of three distinct motions, namely Reach-Grasp-Place. The above dataset is divided in two subsets: (a) 45 arm-motion trajectories are randomly selected and assigned to the training dataset, and hence the latter contains 135,000 data points; (b) the remaining 5 arm-motion trajectories are assigned to the test dataset which contains 15,000 data points.

Training of the simulated grasping and lifting is accomplished in a trial-and-error fashion. Accordingly, for a given object, the gripper attempts to grasp and lift it, initially by applying the smallest possible force from the three force-profiles that are employed. If this is a successful attempt, it is recorded as such and used as a training case. In many cases this fails, and hence when the gripper closes and attempts to lift the object the latter slips and remains on the table (Fig. 5.7-top row). This attempt is then dropped, the applied gripper force is increased to the next higher one, and grasping and lifting is attempted again. Success or failure would again lead to the same actions as above; in the latter case (failure), the gripper force is increased to the highest value, and the task is repeated for the last time (Fig. 5.7-bottom row). The above training procedure has been repeated two times for both NN module implementations, i.e. DNN and RNN.

Figure 5.8 summarizes the *Loss* and *Accuracy* plots derived from the training phase for the DNN and RNN modules, respectively. The *Loss* plots are calculated by employing the Root Mean Square Error (RMSE) metric. In short, the *Loss* function is a commonly adopted metric in artificial NNs, used to measure the inconsistency between the predicted values and actual ones. In most cases it results in a decreasing curve that drops significantly after a sufficient number of training epochs. As can be observed from Fig. 5.8, for both DNN and RNN implementations, *Loss* has dropped considerably after approximately 2000 epochs, and reached almost a plateau at 10,000 epochs. While in both cases *Loss* has achieved a final small value, for the case of DNN the obtained RMSE value of 0.03 is superior to the

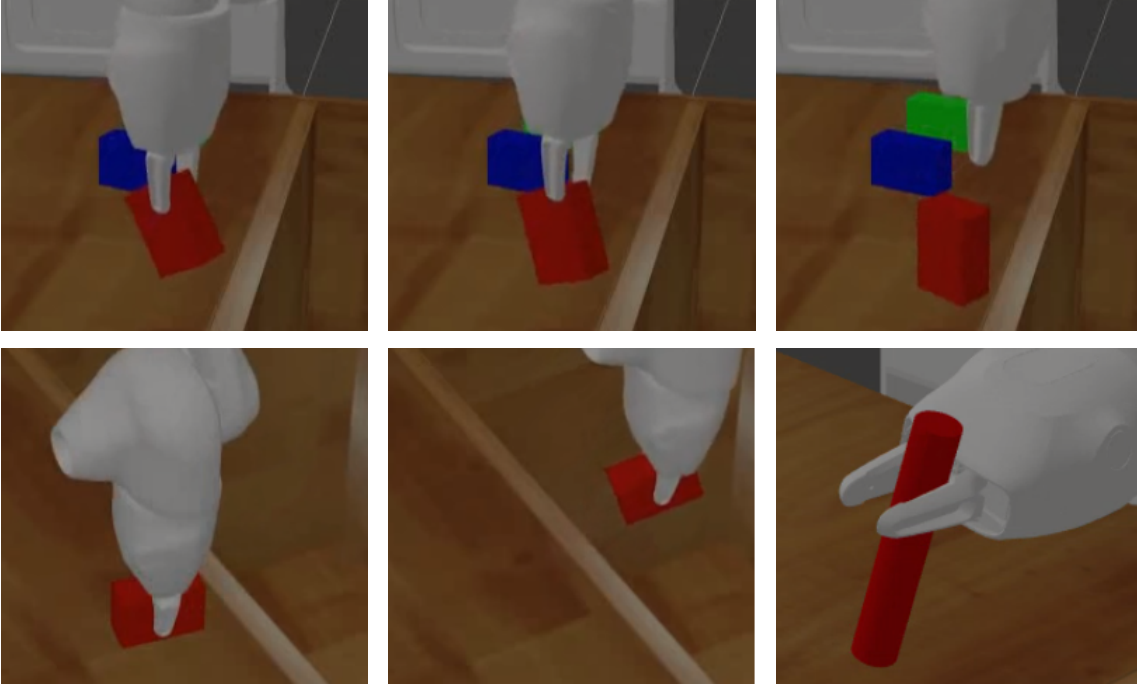


Figure 5.7: **Top row:** Snapshots from failure case of lifting object because of slipping. **Bottom row:** Successful reproduction of lifting motions after training of the proposed scheme.

0.28 for the case of RNN. Similarly, the *Accuracy* figure is used to quantify the error between a learned action (training data point) and the reproduced one during training. The relevant *Accuracy* plots for DNN and RNN are also depicted in Fig. 5.8, verifying robust performance of both networks already after approximately 2000 training epochs. The final achieved *Accuracy* figures of 0.91 for DNN and 0.80 for RNN are sufficiently appropriate for the studied task. The results summarized in Fig. 5.8 indicate that the DNN implementation has accomplished better training when compared to the RNN.

Analogous results were obtained when measuring RMSE for the seven arm joints ($q_1 \cdots q_7$) and the gripper force F_w . Given the simulated environment, ground truth information is available for all involved quantities, and hence RMSE can be computed. Figure 5.9 illustrates RMSE obtained for both modules, i.e. DNN and RNN. Evidently, in all cases the prediction errors are very small, indicating appropriate training. Still, DNN has consistently resulted in smaller RMSE values, suggesting superior performance with respect to RNN. Interestingly, as can be observed, the larger errors correspond to joint q_2 , which is expected since q_2 is mostly used for the arm motions in our experiments.

The results outlined above and summarized in Figs. 5.8, 5.9 indicate that whereas both

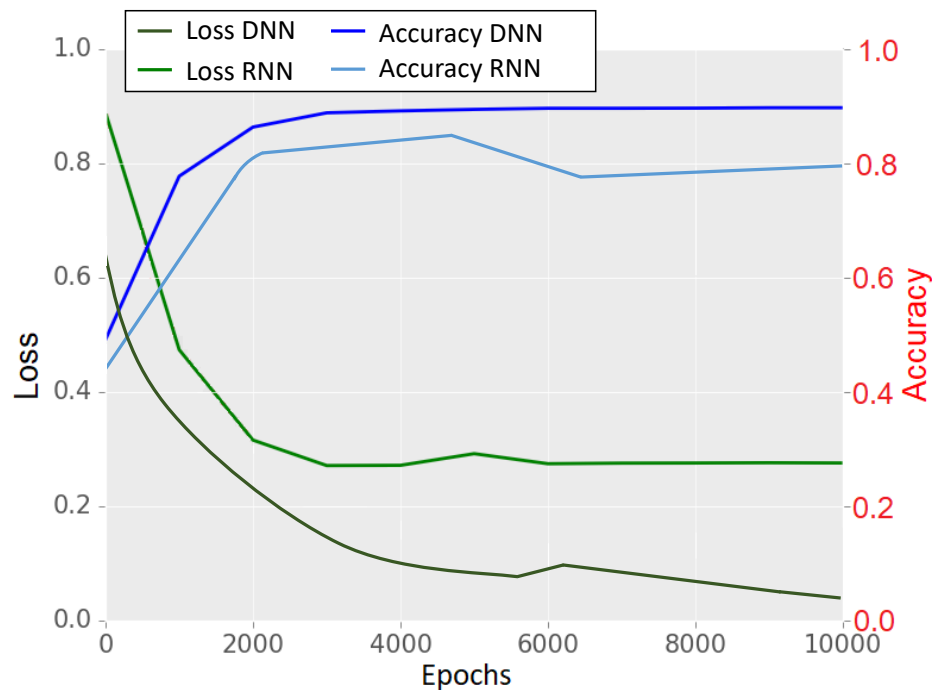


Figure 5.8: Learning curves for the cases of DNN and RNN training.

NN formulations, i.e. DNN and RNN, gave rise to similar results, DNN is consistently slightly more accurate. Overall, for the 7 arm-joints the RMSE error for the DNN implementation is approximately 2%, whereas RNN training resulted in total error of 3.5%. Errors in the gripper force are very small, 0.1% and 0.5%, respectively for DNN and RNN. Accordingly, the DNN formulation is adopted and used in our system, since it consistently outperforms RNN. In essence, the internal temporal dependence of the RNN circuit does not seem to contribute sufficiently to the training at hand, whereas the spatial structure of the DNN better adapts to the characteristics of the object manipulation task.

5.2.2 Transfer Learning for Real Robot Experiments

To further evaluate the performance of our force-based lifting scheme, we conducted a set of experiments with the ABB YuMi manipulator comprising left-hand reach-grasp-place movements. The training set consisted of 30 real-world lifting attempts (10 for each object class) that were conducted via kinesthetic guidance by a human operator. Figure 5.10 illustrates the mean trajectories recorded for each class of objects.

It is noted here that in order to effectively record during kinesthetic teaching the suitable force that the human teacher applies in each successful grasping manipulation, we augmented the YuMi gripper with two Force Sensing Resistors (FSR). Figure 5.11 shows

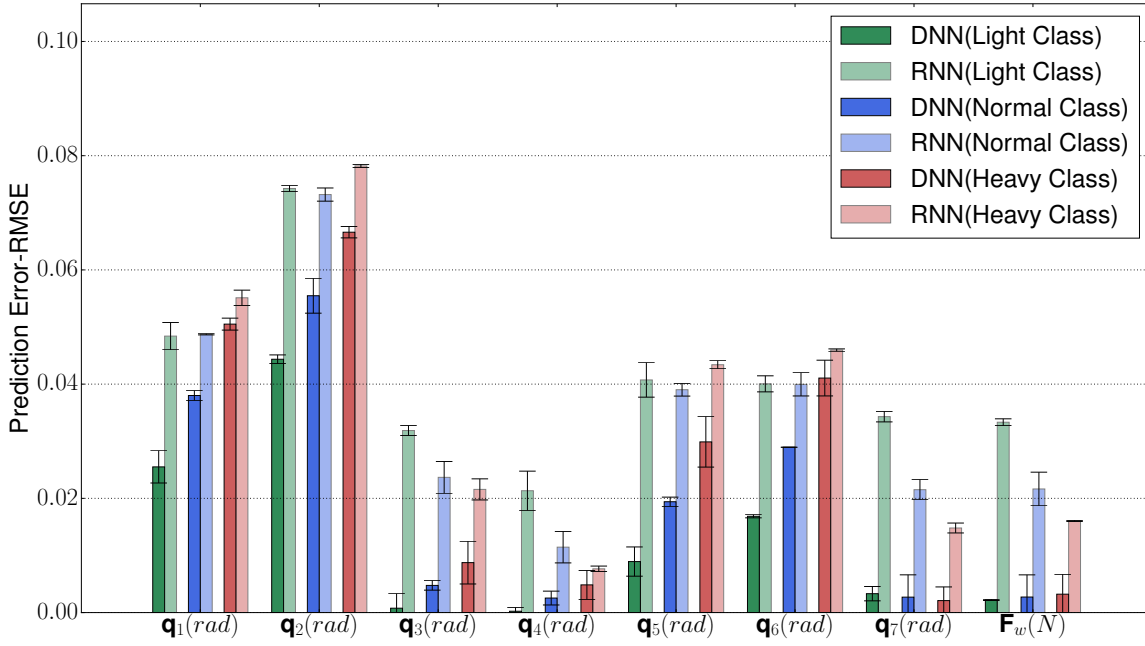


Figure 5.9: RMSE for the six arm joints and the wrist's force from the 1000 simulations; green bars indicate the prediction errors for the green class of objects, blue bars for the blue one and red bars for the red one; black lines indicate the standard deviation from the corresponding mean values; bold bars indicate the errors with the DNN implementation and light bars with the RNN in all cases.

the demonstrator teaching kinesthetically the arm to grasp the 3 objects used to represent each class, i.e. a black box of $0.01kg$ mass for the light class (Fig. 5.11-left); a remote control of $0.1kg$ for the normal one (Fig. 5.11-middle); a green box of $0.5kg$ that represents the heavy class of objects (Fig. 5.11-right). Additionally, the integrated FSR-circuit that was embedded on the left arm of the YuMi robot can be observed in Fig. 5.11.

Training via kinesthetic teaching is inherently a very slow process, and can involve only a limited number of sessions, i.e. object manipulations. Accordingly, and in order to sufficiently train the NN module, we employed the transfer learning approach by combining the recorded data from the simulated experiments with the data from the real ones. In this context, the NN module (DNN as derived above) was trained using the training set from the simulation experiments (135,000 points) and the one from the kinesthetically trained YuMi arm (30,000 points). The obtained results are illustrated in Fig. 5.12, showing very similar *Loss* and *Accuracy* plots with the DNN ones (Fig. 5.8).

Following *Sim-to-Real transfer learning*, we conducted an experiment to assess the trained system in real reach-grasp-lift scenarios. To that end, we used 4 novel objects

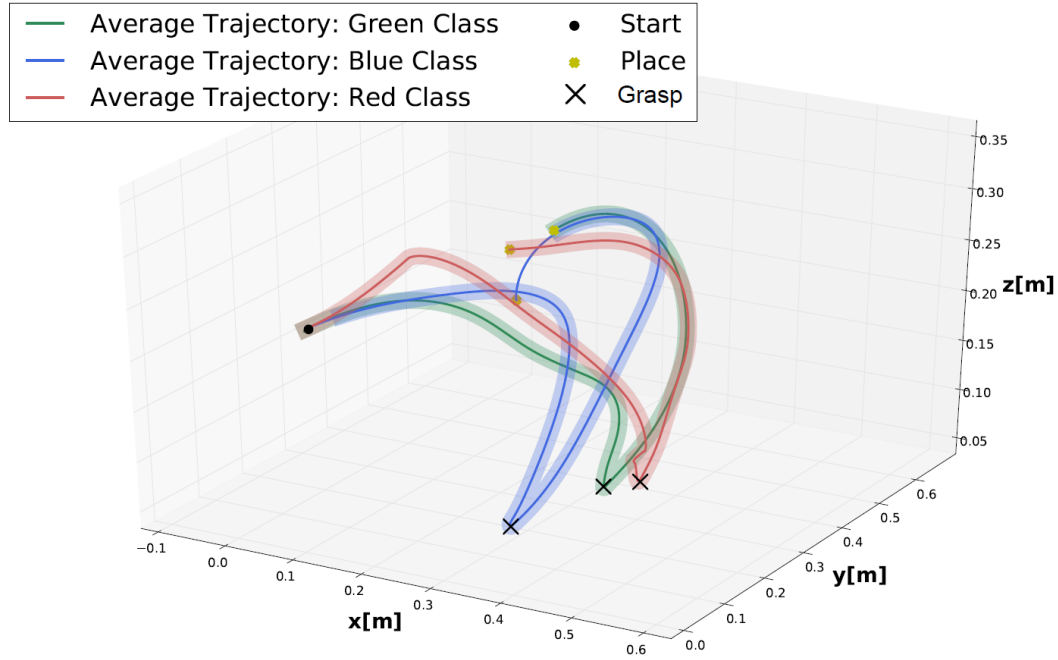


Figure 5.10: Mean trajectories of object lifting manipulation during training via kinesthetic teaching. Green Curve: average trajectory for reach-grasp-lift the green class object. Blue Curve: average trajectory for reach-grasp-lift the blue class object. Red Curve: average trajectory for reach-grasp-lift the red class object. Standard deviations are marked with the lighter thick line for each trajectory. Black circle denotes the starting point, whereas black and yellow crosses indicate the grasp and place points, respectively, as they were computed via the zero velocity crossing algorithm.

(varying on mass), namely green box (0.015kg), empty cup (0.03kg), tape (0.2Kg), bottle (0.42kg) and repeatedly executed the manipulation task 25 times for each object. Table 5.1 summarizes the results for all repetitions for the 4 novel objects. As can be observed, the results show very accurate reach and grasp movements, despite the fact that the 4 objects were not used during training.

A final, more challenging experiment, was conducted where the manipulated object was an *egg*. As in the case of all previous experiments, YuMi was set to reach, grasp and place the egg-object, by applying the proper required force for effectively lifting it. Obviously, applied forces with lower or higher values would cause the egg to break; in the former case due to the egg slipping from the gripper and in the latter due to the egg being squeezed by the gripper. More specifically, a small egg of 0.046kg weight (considered as light class) was used, and placed on the workspace-table in front of the YuMi arm. The experiment was repeated 10 times and succeeded in 9 times with the predicted force at

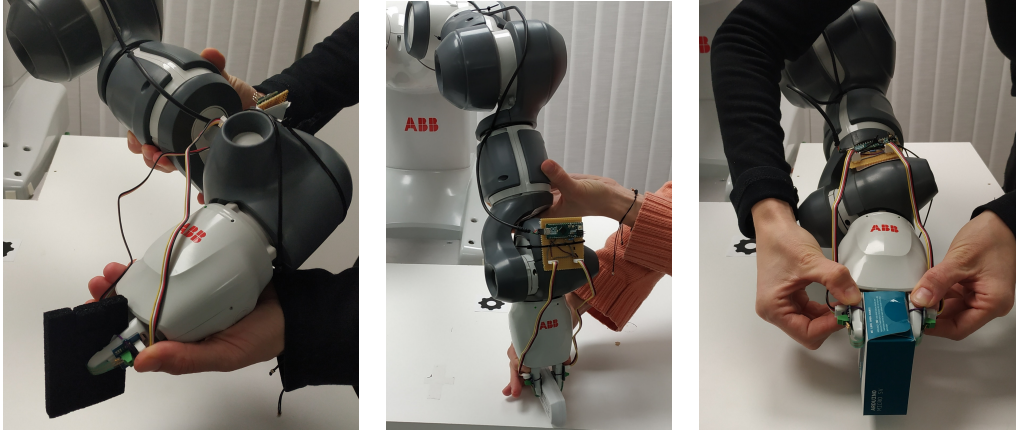


Figure 5.11: **Left:** a black box of 0.01kg mass for the light class; **Middle:** a remote control of 0.1kg for the normal one; **Right:** a green box of 0.5kg that represents the heavy class of objects.

5.22N. One failure case did happen where the predicted force was higher, at 6.5N, and the egg was smashed.

Indicative results from our experiments are showcased in the supplementary video, and relevant snapshots are also shown in Figures 5.13 and 5.14 . Overall, the obtained results demonstrate high accuracy in learning and predicting the required suitable force,

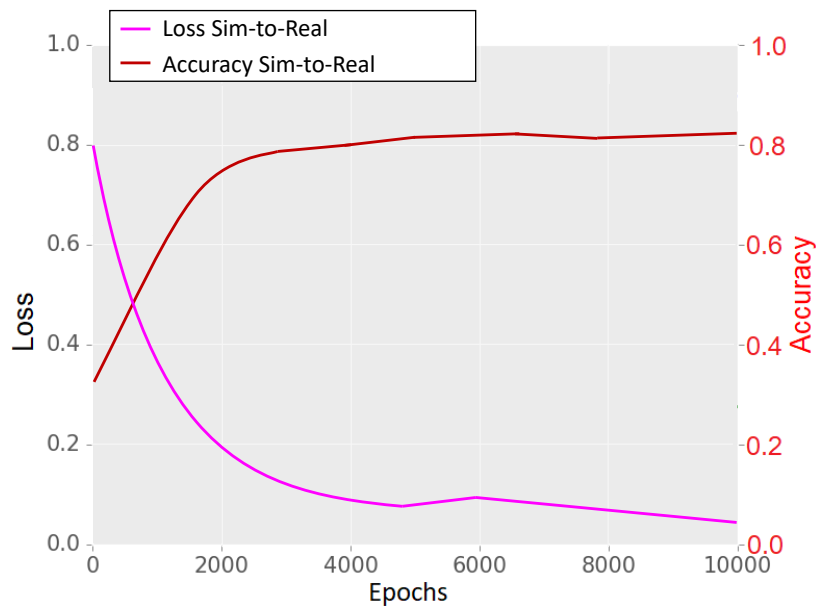


Figure 5.12: Sim-to-Real learning curve.

Table 5.1: Experimental Results for novel object manipulation

Objects	Successful Attempts	Failures	
		Slip	Squeeze
<i>Green Box</i>	24	0	1
<i>Empty Cup</i>	23	0	2
<i>Tape</i>	23	1	1
<i>Bottle</i>	22	1	2

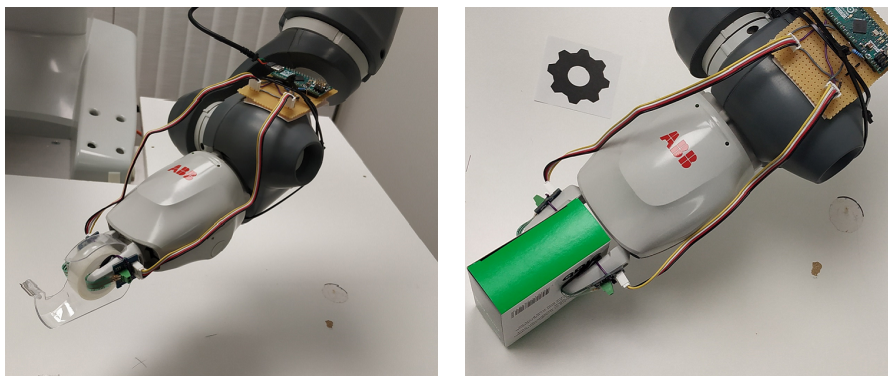


Figure 5.13: Two representative grasp-lifting shots from the objects used for validation, namely tape and green box, are shown in this figure.

along with the arm's motion trajectory, to manipulate delicate objects. All the presented experiments can be also visualized in high resolution at https://youtu.be/1_lv0yAywXA.

5.3 Chapter Highlights

In this Chapter, we presented a novel supervised learning scheme to effectively accomplish reach-grasp-lift tasks for delicate objects. After suitable training, the reaching trajectory, grasping pose and the appropriate applied force are learnt and effectively used for object manipulation tasks.

The proposed scheme employs an NN as the primary module that facilitates training to reach and grasp with the proper required force. We have formulated two appropriate modules, namely DNN and RNN, and experimentally have shown the DNN to perform very accurately. The latter has been adopted in our implementation, and based on *Sim-to-Real transfer learning* we have accomplished robust and proper manipulation of novel objects with the suitable required force. Quantitative and qualitative experimental results attest the above; in addition, the proposed SLF has succeeded in the most challenging



Figure 5.14: Snapshots from the implemented egg-lifting experimental setup.

manipulation of an egg object.

In our future work, we plan to apply the proposed integrated learning scheme to routine tasks in realistic setups, where force/torque feedback is required. Relevant tasks include manipulation of food items, such as tomatoes and pies, and grasping and placing of deformable objects. In this context, we aim at including in our formulation additional object attributes, besides the mass, such as shape, orientation, size, etc. Moreover, we plan to extend force-based skill learning to robots in an active learning for auto-correction and on line update of recovered movements. Also, the proposed SLF can potentially be combined with slipping avoidance techniques as in [115], to offer an integrated scheme that caters for suitable force-based grasping, enhanced with intrinsic slippage prevention. Finally, a longer-term extension of the approach regards dual-arm robots, multi-robot systems and human-robot collaborative setups where force-based manipulation is more involved.

Chapter 6

Discussion

6.1 Conclusions

A major issue in motor learning regards the difficulty in acquiring kinematic capacities that are inherently high dimensional. To deal with this problem, a direction of research in this thesis focused on obtaining an appropriate mapping of the high-dimensional motor space in a subspace of lower dimensionality, called latent space. In order to establish a latent space encapsulating the main characteristics of the motion, several linear and nonlinear methods have been proposed as already presented in Chapter 2. Among linear methods, Principal Component Analysis (PCA) is one of the most common approaches used in various fields of research where the datasets consist of human motion data. In this work, in order to tackle the inherent nonlinearities of the motion, we developed an LfD approach based on the non-linear dimensionality reduction technique, namely Gaussian Process Latent Variable Model (GPLVM).

Capitalizing on the GPLVM-based encoding of observed actions in latent representations, we then formulated and implemented a robust LfD framework, termed Imitation Framework by Observation (IMFO). As a typical LfD scheme, IMFO comprises three distinct phases: (i) observation and data acquisition, (ii) latent space formulation, and (iii) robotic reproduction. By modeling the reciprocal interaction of the two ends, i.e. observed and reproduced actions, the proposed formulation effectively accomplishes to map the observed world to the robot's one. Accordingly, IMFO succeeds in endowing robotic systems with human-like action capabilities. Our extensive examination has revealed very useful and interesting properties of the formulated latent representations, namely: (i) well-separated latent representations for different actions, (ii) neighboring points in the configuration space are mapped to neighboring points in the latent space, thereby preserving action continuity in the latent depiction, and (iii) errors in the spatial and temporal domains due to the latent transformation are very small and in practice do not affect the method's performance.

It is interesting to note that research in the LfD field has rather overlooked two particular and relevant topics. On one hand, most of the work in LfD has focused on how to repre-

sent and reproduce a given action task, and only a few researchers have worked on finding possible solutions to the "what to imitate" problem. On the other hand, learning has been applied in trajectory following tasks, where only spatial information -coming mostly from vision systems- has been used as input to encode and reproduce the demonstrated skill. Nevertheless, it is worth emphasizing that such learning systems may miss relevant data in more general scenarios, like tasks implying physical contacts, fine manipulation skills or multi-agent skills with temporal constraints.

In turn, our research investigated the role of temporal planning in LfD, by enhancing the basic formulation of IMFO framework with the feature of action execution time. Accordingly, we succeeded to encode both the spacial and temporal aspects of the learned motor behaviors. This spatio-temporal framework has been extensively tested in the execution of a realistic serving scenario, where execution time is critical for the customer's idle time. Obtained results regarding reproduction of the spatio-temporal characteristics of the demonstrated actions, validate the successful involvement of the robotic system in the accomplishment of relevant tasks, whereby learned action behaviors are appropriately executed at varying speeds.

During the last year of this PhD, our internship in ABB Corporate Research Center in Vasteras, Sweden, under the supervision of Dr. Pietro Falco, resulted in a formulation regarding a learning scheme for dexterous manipulation of fragile objects. The proposed, Supervised Learning scheme for Force-based lifting and manipulation, has been validated using an ABB YuMi robotic platform. The conducted experiments attest for the system's accuracy and validity by effectively learning the appropriate force required for a successful object lifting.

Overall, the current thesis contributed to LfD by proposing a comprehensive framework (IMFO) to deal with action observation, encoding and reproduction. The compact representation of actions in a latent space lies in the heart of IMFO and constitutes a key development of our work. Interestingly, IMFO has been successfully advanced to also include temporal aspects of relevant actions as well as features involved in force-based lifting and manipulation. By systematically pursuing the above, we believe we have succeeded in accomplishing to:

- Effectively learn and reproduce action behaviors, irrespective of the robotic embodiment;
- Deal with novel actions in HRI/HRC contexts;
- Reveal spatial and temporal aspects of the studied actions, effectively facilitating action learning and reproduction at different speeds;
- Provide insight to the key role of spatio-temporal information in the obtained latent representations of both human and robot arm motions;

- Predict the suitable force that has to be applied on a sensitive object for effective lifting and manipulation.

The above are also evidenced by the extensive experimental results obtained in the context of this thesis.

6.2 Directions for Future Work

Evidently, the formulation of the proposed LfD methodological framework appears to be a tangible technical contribution in domains such as robot learning and imitation, motor control, HRI and HRC. Nevertheless, work in the current thesis, besides addressing specific and challenging problems, has opened up future research directions in multiple directions.

At first, it seems rather straightforward to opt for the extension of the proposed framework to formulate a generic methodology for full body imitation through latent representation. Through the current formulation, we have set the basis to imitate human arm-motions by mere observation. Since no limiting assumptions underly the latter, full body imitation lends itself as a direct extension, relying on contemporary full body tracking systems [174], [175].

Further to the above, the application of the proposed integrated learning scheme to routine, yet challenging tasks in realistic scenarios and industrial setups, whereby force/torque feedback is crucial, is a fruitful direction. Relevant tasks include manipulation of sensitive food items and grasping and placing of deformable objects [176], [177]. The evaluation metrics introduced in this thesis can be systematically employed in the above cases to rigorously assess the performance in the named scenarios.

Another promising direction for future work is the combination of our approach with obstacle avoidance methods. In other words, assuming the presence of objects in the robot's workspace, the representation of obstacles in the latent space should systematically be investigated in order to enhance the current imitation framework with the ability of auto-correction and recovering movements for online adaptation to unexpected perturbations, i.e. obstacles, novel environments, etc.

The issues addressed in our study regard mostly the single-robot LfD problem, i.e. developing imitation strategies from multiple demonstrations. More involved scenarios should consider multiple agents, and hence noteworthy extensions regard dual-arm robots, multi-robot systems and advanced human-robot collaborative setups where force-based manipulation is more involved.

Broadly speaking, LfD technologies developed in the current PhD thesis address specific applications, but at the same time bear potential for far-reaching impact. The advent of smart robot companions, hybrid-agent systems, and human-robot symbiotic setups, calls for robust LfD implementations. In this context, the proposed framework along

with all developed modules may be particularly beneficial and form the seed for more enhanced systems. The directions for future work proposed in the current section augment nicely contemporary technologies, ultimately leading to skilled AI agents. Imitation capacities in turn can significantly promote the behavioral traits of the latter, and pave the way towards robots acceptable in man-made environments.

Bibliography

- [1] B. Argall, S. Chernova, M. Veloso, and B. Browning, “A survey of robot learning from demonstration,” *Rob. Auton. Syst.*, vol. 57, no. 5, pp. 469–483, 2009.
- [2] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, “Learning force control policies for compliant manipulation,” *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 4639–4644, 2011.
- [3] A. Ijspeert, J. Nakanishi, and S. Schaal, “Movement imitation with nonlinear dynamical systems in humanoid robots,” *Proc. 2002 IEEE Int. Conf. Robot. Autom. (Cat. No. 02CH37292)*, vol. 2, pp. 1–6, 2002.
- [4] S. Calinon, F. Guenterlorent, and A. Billard, “On learning, representing, and generalizing a task in a humanoid robot,” *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 37, pp. 286–298, 2007.
- [5] P. Lucia, K. Umezawa, Y. Nakamura, and A. Billard, “Learning Robot Skills Through Motion Segmentation and Constraints Extraction,” *Proc. Collab. Manip. Work. ACM/IEEE Int. Conf. Human-Robot Interact. (HRI 2013)*, 2013.
- [6] S. Schaal, “Robot learning from demonstration,” *Adv. Neural Inf. Process. Syst.*, no. 9, pp. 1040–1046, 1997.
- [7] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovi, “Style-based inverse kinematics,” *ACM Trans. Graph.*, vol. 23, pp. 522–527, 2004.
- [8] D. B. Grimes and R. P. N. Rao, “Learning actions through imitation and exploration: Towards humanoid robots that learn from humans,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5436, pp. 103–138, 2009.
- [9] D. Grimes, D. Rashidand, and R. P. N. Rao, “Learning Nonparametric Models for Probabilistic Imitation,” *Adv. Neural Inf. Process. Syst. 19*, vol. 19, pp. 521–528, 2007.
- [10] C. Breazeal and B. Scassellati, “Robots that imitate humans,” *Trends Cogn. Sci.*, vol. 6, no. 11, pp. 481–487, 2002.
- [11] Z. Zhu and H. Hu, “Robot learning from demonstration in robotic assembly: A survey,” *Robotics*, vol. 7, no. 2, 2018.

- [12] F. Ficuciello, P. Falco, and S. Calinon, "A brief survey on the role of dimensionality reduction in manipulation learning and control," *IEEE Robotics and Automation Letters*, vol. 3, p. 2608–2615, jul 2018.
- [13] M. S. Chesters, "Human visual perception and ROC methodology in medical imaging," *Physics in Medicine and Biology*, vol. 37, pp. 1433–1476, jul 1992.
- [14] F. Paas and J. Sweller, "An evolutionary upgrade of cognitive load theory: Using the human motor system and collaboration to support the learning of complex cognitive tasks," *Educational Psychology Review*, vol. 24, pp. 27–45, Mar 2012.
- [15] G. Sandini, A. Sciutti, and F. Rea, *Movement-Based Communication for Humanoid-Human Interaction*, pp. 1–29. 01 2017.
- [16] R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel, "Natural human-robot interaction using speech, head pose and gestures," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE)*, pp. 2422–2427 vol.3, 2004.
- [17] A. X. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel, "Learning force-based manipulation of deformable objects from multiple demonstrations," in *IEEE Intern. Conf. on Robotics and Automation (ICRA)*, pp. 177–184, May 2015.
- [18] M. L. Infante and V. Kyrki, "Usability of force-based controllers in physical human-robot interaction," in *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 355–362, 2011.
- [19] K. Iqbal, "Mechanisms and models of postural stability and control," *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, vol. 2011, pp. 7837–40, 08 2011.
- [20] H. J. Chiel, L. H. Ting, Ö. Ekeberg, and M. J. Z. Hartmann, "The brain in its body: Motor control and sensing in a biomechanical context," *Journal of Neuroscience*, vol. 29, no. 41, pp. 12807–12814, 2009.
- [21] M. Maniadakis, E. Aksoy, T. Asfour, and P. Trahanias, "Collaboration of heterogeneous agents in time constrained tasks," in *Proc. IEEE-RAS Intern. Conf. on Humanoid Robots (Humanoids)*, 2016.
- [22] M. Maniadakis and P. Trahanias, "Time-informed, adaptive multi-robot synchronization," in *Proc. Simulation of Adaptive Behavior (SAB)*, 2016.

- [23] M. Koskinopoulou, M. Maniadakis, and P. Trahanias, "Speed adaptation in learning from demonstration through latent space formulation," *Robotica*, p. 1–13.
- [24] M. Koskinopoulou, M. Maniadakis, and P. Trahanias, "Learning spatio-temporal characteristics of human motions through observation," in *Advances in Service and Industrial Robotics* (N. A. Aspragathos, P. N. Koustoumpardis, and V. C. Moulianitis, eds.), (Cham), pp. 82–90, Springer International Publishing, 2019.
- [25] M. Koskinopoulou and P. E. Trahanias, "A methodological framework for robotic reproduction of observed human actions: Formulation using latent space representation," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th Intern. Conf. on*, pp. 565–572, IEEE Press, 15–17 Nov. 2016.
- [26] M. Koskinopoulou, S. Piperakis, and P. E. Trahanias, "Learning from demonstration facilitates human-robot collaborative task execution," in *The Eleventh ACM/IEEE Intern. Conf. on Human Robot Interaction*, pp. 59–66, IEEE Press, 2016.
- [27] S. Piperakis, M. Koskinopoulou, and P. E. Trahanias, "Nonlinear state estimation for humanoid robot walking," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3347–3354, 2018.
- [28] S. Piperakis, M. Koskinopoulou, and P. E. Trahanias, "Nonlinear state estimation for humanoid robot walking," Madrid, Spain, 2018.
- [29] "Goal-directed imitation for robots: A bio-inspired approach to action understanding and skill learning," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 353 – 360, 2006. The Social Mechanisms of Robot Programming from Demonstration.
- [30] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation.," *Philos. Trans. R. Soc. Lond. B. Biol. Sci.*, vol. 358, pp. 537–547, 2003.
- [31] Y. Kuniyoshi, M. Yasuo, and H. I. Inaba, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," *IEEE Trans. Robot. Autom.*, vol. 10, no. 6, pp. 799–822, 1994.
- [32] E. Oztop and M. Kawato, "Models for the control of grasping," *Sensorimotor Control Grasping*, pp. 110–124, 2009.
- [33] R. I. Rumiati and H. Bekkering, "To imitate or not to imitate? How the brain can do it, that is the question!," *Brain Cogn.*, vol. 53, no. 3, pp. 479–482, 2003.
- [34] G. Rizzolatti and M. Arbib, "Language within our grasp," *Trends Neurosci.*, vol. 21, no. 5, pp. 188–194, 1998.

- [35] S. Schaal, J. Peters, and J. Nakanishi, "Learning movement primitives," *Robot. Res.*, pp. 1–10, 2005.
- [36] J. Ernesti, L. Righettit, M. Do, T. Asfour, and S. Schaal, "Encoding of Periodic and their Transient Motions by a Single Dynamic Movement Primitive," *12th IEEE-RAS Int. Conf. Humanoid Robot.*, pp. 57–64, 2012.
- [37] A. J. Ijspeert, J. Nakanishi, and S. Schaal *Adv. Neural Inf. Process. Syst.* 2002.
- [38] J. Nakanishi, J. Morimoto, G. C. Gordon, M. Kawato, and S. Schaal, "Learning from demonstration and adaptation of biped locomotion," *Rob. Auton. Syst.*, pp. 79–91, 2004.
- [39] K. Kronander, S. M. Khansari-Zadeh, and A. Billard, "Learning to control planar hitting motions in a minigolf-like task," *2011 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 710–717, 2011.
- [40] E. Gribovskaya, S. M. Khansari-Zadeh, and A. Billard, "Learning Non-linear Multivariate Dynamics of Motion in Robotic Manipulators," *Int. J. Rob. Res.*, vol. 30, no. 1, pp. 80–117, 2011.
- [41] S. Calinon and A. Billard, "Learning of Gestures by Imitation in a Humanoid Robot," *Imitation Soc. Learn. Robot. Humans Anim. Behav. Soc. Commun. Dimens.*, pp. 153–177, 2007.
- [42] A. Ude, C. Atkeson, and M. Riley, "Programming full-body movements for humanoid robots by observation," *Rob. Auton. Syst.*, vol. 47, no. 2-3, pp. 93–108, 2004.
- [43] S. Kim, C. Hwan, B. You, and S. Oh, "Stable whole-body motion generation for humanoid robots to imitate human motions," in *2009 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS 2009*, pp. 2518–2524, 2009.
- [44] E. Sauser, B. Argall, and A. Billard, "The life of iCub, a little humanoid robot learning from humans through tactile sensing," *Human-Robot Interact. (HRI), 2011 6th ACM/IEEE Int. Conf.*, p. 393, 2011.
- [45] S. M. Khansari-Zadeh and A. Billard, "Learning Stable Non-Linear Dynamical Systems with Gaussian Mixture Models," *IEEE Trans. Robot.*, vol. 27, pp. 943–957, 2011.
- [46] S. M. Khansari-Zadeh, K. Kronander, and A. Billard, "Modeling robot discrete movements with state-varying stiffness and damping: A framework for integrated motion generation and impedance control," *Robot. Sci. Syst.*, pp. 1–10, 2014.
- [47] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Learning Dynamical System Modulation for Constrained Reaching Tasks," *Proc. IEEE Humanoids*, pp. 444–449, 2006.

- [48] M. Hersch, F. Guenterlorent, S. Calinon, and A. Billard, "Dynamical system modulation for robot learning via kinesthetic demonstrations," *IEEE Trans. Robot.*, vol. 24, pp. 1463–1467, 2008.
- [49] M. Liarokapis, C. P. Bechlioulis, P. K. Artemiadis, and K. J. Kyriakopoulos, "Deriving Humanlike Arm Hand System Poses," *Journal of Mechanisms and Robotics*, vol. 9, no. 1, 2017.
- [50] L. Peternel and J. Babic, "Humanoid robot posture-control learning in real-time based on human sensorimotor learning ability," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 5329–5334, 2013.
- [51] J. Babic, J. G. Hale, and E. Oztoprhan, "Human sensorimotor learning for humanoid robot skill synthesis," *Adapt. Behav.*, vol. 19, no. 4, pp. 250–263, 2011.
- [52] P. Pook and D. Ballard, "Recognizing teleoperated manipulations," *Int. Conf. Robot. Autom.*, pp. 578–585, 1993.
- [53] J. D. Sweeney and R. Grupen, "A Model of Shared Grasp Affordances from Demonstration," *Proc. 2007 7th IEEE-RAS Int. Conf. Humanoid Robot. HUMANOIDS 2007*, pp. 27–35, 2008.
- [54] S. Schaal, "Is imitation learning the route to humanoid robots," vol. 3, no. 6, pp. 233–242, 1999.
- [55] G. M. E. L. Sauser, B. D. Argall, and A. Billard, "Iterative learning of grasp adaptation through human corrections," *Rob. Auton. Syst.*, vol. 60, no. 1, pp. 55–71, 2012.
- [56] K. Kronander and A. Billard, "Online learning of varying stiffness through physical human-robot interaction," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 1842–1849, 2012.
- [57] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous Helicopter Aerobatics through Apprenticeship Learning," *Int. J. Rob. Res.*, vol. 29, no. 13, pp. 1608–1639, 2010.
- [58] D. H. Grollman and O. Jenkins, "Dogged learning for robots," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 2483–2488, 2007.
- [59] S. Calinon, I. Sardellitti, and D. G. Caldwell, "Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies," *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS 2010 - Conf. Proc.*, pp. 249–254, 2010.
- [60] R. A. Peters, "Robonaut task learning through teleoperation," *ICRA'03. IEEE*, pp. 2806–2811, 2003.

- [61] M. Pardowitz, S. Knoop, R. Dillmann, and R. Zollner, "Incremental learning of tasks from user demonstrations, past experiences, and vocal comments," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 37, no. 2, pp. 322–332, 2007.
- [62] A. G. Billard, S. Calinon, and R. Dillmann, *Learning from Humans*. Cham: Springer International Publishing, 2016.
- [63] A. Shon, K. Grochow, A. Hertzmann, and R. Rao, "Learning shared latent structure for image synthesis and robotic imitation," *Adv. Neural Inf. Process. Syst.*, vol. 18, p. 1233, 2006.
- [64] J. Kober, E. Oztophan, and J. Peters, "Reinforcement learning to adjust robot movements to new situations," *IJCAI Int. Jt. Conf. Artif. Intell.*, pp. 2650–2655, 2011.
- [65] H. Veeraraghavan and M. Veloso
- [66] P. E. Rybski, K. Yoon, J. Stolarz, M. Veloso, and M. Manuela, "Interactive Robot Task Training through Dialog and Demonstration," in *Proceeding ACM/IEEE Int. Conf. Human-Robot Interact.*, pp. 49–56, 2007.
- [67] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," *2009 IEEE Int. Conf. Robot. Autom.*, 2009.
- [68] S. M. Khansari-Zadehand and A. Billard, "Imitation learning of globally stable nonlinear point-to-point robot motions using nonlinear programming," *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS 2010 - Conf. Proc.*, pp. 2676–2683, 2010.
- [69] S. M. Khansari-Zadeh and A. Billard, "Learning to Play Mini-Golf from Human Demonstration using Autonomous Dynamical Systems," *Proceeding 28th Int. Conf. Mach. Learn.*, 2011.
- [70] S. Stavridis, D. Papageorgiou, and Z. Doulgeri, "Dynamical system based robotic motion generation with obstacle avoidance," *IEEE Robotics and Automation Letters*, vol. 2, pp. 712–718, April 2017.
- [71] S. M. Khansari-Zadeh, K. Kronander, and A. Billard, "Learning to Play Minigolf: A Dynamical System-Based Approach," *Adv. Robot.*, vol. 26, no. 17, pp. 1967–1993, 2012.
- [72] L. Lukic, J. Santos-Victor, and A. Billard, "Learning robotic eye–arm–hand coordination from human demonstration: a coupled dynamical systems approach," *Biol. Cybern.*, vol. 108, no. 2, pp. 223–248, 2014.

- [73] S. Kim, E. Gribovskaya, and A. Billard, "Learning motion dynamics to catch a moving object," *2010 10th IEEE-RAS Int. Conf. Humanoid Robot. Humanoids 2010*, pp. 106–111, 2010.
- [74] A. Ude, T. A. A. Gams, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *IEEE Trans. Robot.*, vol. 26, no. 5, pp. 800–815, 2010.
- [75] B. Nemec, M. Tamosiunait, F. Wotter, and A. Ude, "Task adaptation through exploration and action sequencing," pp. 610–616, 2009.
- [76] M. S. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Auton. Robots*, vol. 32, no. 4, pp. 433–454, 2012.
- [77] F. Stulp, E. Oztoprhan, P. Pastor, , M. Beetz, and S. Schaaz, "Compact models of motor primitive variations for predictable reaching and obstacle avoidance," *9th IEEE-RAS Int. Conf. Humanoid Robot. HUMANOIDS09*, pp. 589–595, 2009.
- [78] S. Kim, A. Shukla, and A. Billard, "Catching Objects in Flight," *IEEE Trans. Robot.*, vol. PP, pp. 1–17, 2014.
- [79] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance," *2009 IEEE Int. Conf. Robot. Autom.*, pp. 2587–2592, 2009.
- [80] J. Kober and J. Peters, "Policy search for motor primitives in robotics," *Mach. Learn.*, vol. 84, pp. 171–203, 2011.
- [81] M. S. Khansari-Zadeh and A. Billard, "Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions," *Rob. Auton. Syst.*, vol. 62, no. 6, pp. 752–765, 2014.
- [82] D. Grimes, R. Chalodhorn, and R. P. N. Rao, "Dynamic Imitation in a Humanoid Robot through Nonparametric Probabilistic Inference," *Robot. Sci. Syst.*, pp. 1–6, 2006.
- [83] D. B. Grimes and R. P. N. Rao, "Learning nonparametric policies by imitation," *2008 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, pp. 2022–2028, 2008.
- [84] P. Aaron, D. B. Grimes, C. L. Baker, W. Matthew, S. Zhou, and R. P. N. Rao, "Probabilistic gaze imitation and saliency learning in a robotic head," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2005, pp. 2865–2870, 2005.

- [85] A. M. Ghalamzan E., C. Paxton, G. D. Hager, and L. Bascetta, "An incremental approach to learning generalizable robot tasks from human demonstration," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5616–5621, May 2015.
- [86] F. Guenter, M. Hersch, S. Calinon, and A. Billard, "Reinforcement learning for imitating constrained reaching movements," *Advanced Robotics*, vol. 21, no. 13, pp. 1521–1544, 2007.
- [87] Ye Gu, A. Thobbi, and W. Sheng, "Human-robot collaborative manipulation through imitation and reinforcement learning," in *2011 IEEE International Conference on Information and Automation*, pp. 151–156, June 2011.
- [88] M. Falahi and M. Jannatifar, "Using orthogonal basis functions and template matching to learn whiteboard cleaning task by imitation," in *ICCKE 2013*, pp. 289–294, Oct 2013.
- [89] V. Kruger, D. Herzog, and S. Baby, "Primitive-Based Modeling and Grammar," *IEEE Robot. Autom. Mag.*, no. June, 2010.
- [90] S. Calinon, F. Guenterlorent, and A. Billard, "On learning the statistical representation of a task and generalizing it to various contexts," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2006, no. Icra, pp. 2978–2983, 2006.
- [91] P. Evrard, E. Gribovskaya, S. Calinon, A. Kheddar, and A. Billard, "Teaching physical collaborative tasks: Object-lifting case study with a humanoid," *9th IEEE-RAS Int. Conf. Humanoid Robot. HUMANOIDS09*, pp. 399–404, 2009.
- [92] O. C. Jenkins and M. J. Mataric, "Automated derivation of behavior vocabularies for autonomous humanoid motion," *Int. Jt. Conf. Auton.*, pp. 225–232, 2003.
- [93] S. Calinon and A. Billard, "Recognition and Reproduction of Gestures using a Probabilistic Framework combining PCA, ICA and HMM," *Proc. ICML*, pp. 105–112, 2005.
- [94] D. Kulic, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *Int. J. Rob. Res.*, vol. 31, no. 3, pp. 330–345, 2012.
- [95] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemom. Intell. Lab. Syst.*, vol. 2, pp. 37–52, 1987.
- [96] A. P. Shon, K. Grochow, and R. P. N. Rao, "Robotic imitation from human motion capture using Gaussian processes," *5th IEEE-RAS Int. Conf. Humanoid Robot*, 2005.

- [97] N. D. Lawrence, "Gaussian Process Latent Variable Models for Visualisation of High Dimensional Data," *Adv. Neural Inf. Process. Syst.* 16, pp. 329–336, 2004.
- [98] S. S. Schaal and C. G. Atkeson, "Constructive incremental learning from only local information," *Neural Comput.*, vol. 10, no. 8, pp. 2047–84, 1998.
- [99] D. Grollman and O. Jenkins, "Incremental learning of subtasks from unsegmented demonstration," *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS 2010 - Conf. Proc.*, pp. 261–266, 2010.
- [100] D. Nguyen-Tuong and J. Peters, "Local Gaussian process regression for real-time model-based robot control," *2008 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, pp. 380–385, 2008.
- [101] M. Schneider and W. Ertel, "Robot Learning by Demonstration with local Gaussian process regression," *Intell. Robot. Syst. (IROS), 2010 IEEE/RSJ Int. Conf.*, pp. 255–260, 2010.
- [102] S. M. Khansari-Zadehand and A. Billard, "BM: An iterative algorithm to learn stable non-linear dynamical systems with Gaussian mixture models," *Proc. - IEEE Int. Conf. Robot. Autom.*, no. 1, pp. 2381–2388, 2010.
- [103] S. Calinon, *Robot Programming by Demonstration*. CRC Press, Inc, 1st ed., 2009.
- [104] M. Pomplun and M. J. Mataric, "Evaluation metrics and results of human arm movement imitation," 2000.
- [105] A. Alissandrakis, C. L. Nehaniv, and K. Dautenhahn, "Correspondence mapping induced state and action metrics for robotic imitation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, pp. 299–307, April 2007.
- [106] A. Alissandrakis, N. Otero, J. Saunders, K. Dautenhahn, and C. Nehaniv, "Helping robots imitate: Metrics and computational solutions inspired by human-robot interaction studies," in *In Gray, J. and Nefti-Meziani, S., editors, Advances in Cognitive Systems*, pp. 127–167, IET Press, 01 2010.
- [107] R. Chalodhorn, D. B. Grimes, K. Grochow, and R. P. N. Rao, "Learning to walk through imitation," *IJCAI Int. Jt. Conf. Artif. Intell.*, pp. 2084–2090, 2007.
- [108] D. Grimes, R. Chalodhorn, and R. P. N. Rao, "Dynamic Imitation in a Humanoid Robot through Nonparametric Probabilistic Inference," *Robot. Sci. Syst.*, pp. 1–6, 2006.
- [109] S. Schaal, C. Atkeson, and S. Vijayakumar, "Scalable Techniques from Nonparametric Statistics for Real Time Robot Learning," *Appl. Intell.*, vol. 17, pp. 49–60, 2002.

- [110] D. Grollman and A. Billard, "Learning from failure ," *HRI 20011*, no. 2, pp. 145–146, 2011.
- [111] M. Li, H. Yin, K. Tahara, and A. Billard, "Learning Object-level Impedance Control for Robust Grasping and Dexterous Manipulation," *Int. Conf. Robot. Autom.*, pp. 6784–6791, 2014.
- [112] C. G. Atkeson and S. Schaal, "Learning tasks from a single demonstration," in *Int. Conf. Robot. Autom.*, vol. 2, pp. 1706–1712, 1997.
- [113] C. Atkeson and S. Schaal, "Robot juggling: implementation of memory-based learning," *IEEE Control Syst. Mag.*, vol. 14, no. 1, pp. 57–71, 1994.
- [114] P. Falco, A. Attawia, M. Saveriano, and D. Lee, "On policy learning robust to irreversible events: An application to robotic in-hand manipulation," *IEEE Rob. and Autom. Letters*, pp. 1482–1489, 2018.
- [115] G. D. Maria, P. Falco, C. Natale, and S. Pirozzi, "Integrated force/tactile sensing: The enabling technology for slipping detection and avoidance.," in *ICRA*, pp. 3883–3889, IEEE, 2015.
- [116] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, pp. 99–134, 1998.
- [117] P. Evrard, E. Gribovskaya, S. Calinon, A. Billard, and A. Kheddar, "Teaching physical collaborative tasks: object-lifting case study with a humanoid," in *Humanoids*, pp. 399–404, IEEE, 2009.
- [118] A. Gupta, C. Eppner, S. Levine, and P. Abbeel, "Learning dexterous manipulation for a soft robotic hand from human demonstrations," in *2016 IEEE/RSJ Intern. Conf. on Intel. Robots and Systems, IROS*, pages = 3786–3793, year = 2016,.
- [119] M. Li, Y. Bekiroglu, D. Kragic, and A. Billard, "Learning of grasp adaptation through experience and tactile sensing," in *Proceedings of the IEEE/RSJ Intern. Conf. on Intel. Robots and Systems IROS*, 2014.
- [120] J. Rey, K. Kronander, F. Farshidian, J. Buchli, and A. Billard, "Learning motions from demonstrations and rewards with time-invariant dynamical systems based policies," *Autonomous Robots*, vol. 42, pp. 45–64, Jan 2018.
- [121] M. Kamel, K. Alexis, and R. Siegwart, "Design and modeling of dexterous aerial manipulator," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4870–4876, Oct 2016.

- [122] J. Peng, J. Wang, and Y. Wang, "Neural network based robust hybrid control for robotic system: an h_∞ approach," *Nonlinear Dynamics*, vol. 65, pp. 421–431, Sep 2011.
- [123] M. Stachowsky, T. Hummel, M. Moussa, and H. A. Abdullah, "A slip detection and correction strategy for precision robot grasping," *IEEE/ASME Trans. on Mechatr.*, vol. 21, pp. 2214–2226, 2016.
- [124] H. P. Singh and N. Sukavanam, "Stability analysis of robust adaptive hybrid position/force controller for robot manipulators using neural network with uncertainties," *Neural Computing and Applications*, vol. 22, pp. 1745–1755, Jun 2013.
- [125] M. Khansari, K. Kronander, and A. Billard, "Modeling robot discrete movements with state-varying stiffness and damping: A framework for integrated motion generation and impedance control," in *Robotics: Science and Systems*, 2014.
- [126] T. Adachi, K. Fujimoto, S. Sakaino, and T. Tsuji, "Imitation learning for object manipulation based on position/force information using bilateral control," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3648–3653, Oct 2018.
- [127] Z. Yang, J. Peng, and Y. Liu, "Adaptive neural network force tracking impedance control for uncertain robotic manipulator based on nonlinear velocity observer," *Neurocomputing*, pp. 1846–1851, 2018.
- [128] J. Li, L. Liu, Y. Wang, and W. Liang, "Adaptive hybrid impedance control of robot manipulators with robustness against environment's uncertainties," *IEEE Intern. Conf. on Mechatronics and Automation (ICMA)*, pp. 1846–1851, 2015.
- [129] F. J. Abu-Dakka, L. Rozo, and D. G. Caldwell, "Force-based variable impedance learning for robotic manipulation," *Robotics and Autonomous Systems*, vol. 109, pp. 156–167, 2018.
- [130] L. Rozo, J. Silverio, S. Calinon, and D. Gordon Caldwell, "Learning controllers for reactive and proactive behaviors in human-robot collaboration," *Frontiers in Robotics and AI*, vol. 3, pp. 1–111, 2016.
- [131] K. Kronander and A. Billard, "Learning compliant manipulation through kinesthetic and tactile human-robot interaction," *IEEE Transactions on Haptics*, vol. 7, pp. 367–380, 11 2014.
- [132] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Advanced Robotics*, vol. 25, pp. 581–603, 2011.

- [133] M. Li, H. Yin, K. Tahara, and A. Billard, “Learning object-level impedance control for robust grasping and dexterous manipulation,” in *2014 IEEE Intern. Conf. on Robotics and Automation (ICRA)*, pp. 6784–6791, May 2014.
- [134] K. Kronander and A. Billard, “Online learning of varying stiffness through physical human-robot interaction,” in *2012 IEEE Intern. Conf. on Robotics and Automation*, pp. 1842–1849, May 2012.
- [135] M. Rambow, T. Schaub, M. Buss, and S. Hirche, “Autonomous manipulation of deformable objects based on teleoperated demonstrations,” in *IEEE/RSJ Intern. Conf. on Intel. Robots and Systems*, 2012.
- [136] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, “Combining self-supervised learning and imitation for vision-based rope manipulation,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2146–2153, May 2017.
- [137] J. Sanchez, J. A. Corrales Ramon, B. Bouzgarrou, and Y. Mezouar, “Robotic manipulation and sensing of deformable objects in domestic and industrial applications: A survey,” *The International Journal of Robotics Research*, vol. 37, pp. 688 – 716, 06 2018.
- [138] T. Gašpar, M. Deniša, and A. Ude, “Knowledge acquisition through human demonstration for industrial robotic assembly,” in *Advances in Service and Industrial Robotics* (K. Berns and D. Görge, eds.), (Cham), pp. 346–353, Springer International Publishing, 2020.
- [139] D. R. Myers, M. J. Pritchard, and M. D. J. Brown, “Automated programming of an industrial robot through teach-by showing,” in *Proceedings 2001 ICRA. IEEE Intern. Conf. on Robotics and Automation*, vol. 4, pp. 4078–4083, May 2001.
- [140] M. Ehrenmann, R. Zollner, O. Rogalla, and R. Dillmann, “Programming service tasks in household environments by human demonstration,” in *Proceedings. 11th IEEE Intern. Workshop on Robot and Human Interactive Communication*, pp. 460–467, Sep. 2002.
- [141] H. Dang and P. K. Allen, “Robot learning of everyday object manipulations via human demonstration,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1284–1289, Oct 2010.
- [142] B. D. Argall and A. G. Billard, “A survey of tactile human-robot interactions,” *Robot. Auton. Syst.*, pp. 1159–1176, 2010.

- [143] M. Sigalas, M. Pateraki, and P. Trahanias, "Full-body pose tracking—the top view re-projection approach," *IEEE transactions on pattern analysis and machine intelligence*, pp. 1569–1582, 2015.
- [144] N. Lawrence and A. Hyvarinen, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *Journal of Machine Learning Research*, pp. 1783–1816, 2005.
- [145] E. Levina and P. J. Bickel, "Maximum likelihood estimation of intrinsic dimension," in *NIPS*, 2004.
- [146] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Third Intern. Conf. on 3D Digital Imaging and Modeling (3DIM)*, 2001.
- [147] N. D. Monnig, B. Fornberg, and G. Meyer, Francois, "Inverting non-linear dimensionality reduction with scale-free radial basis interpolation," tech. rep., 2013.
- [148] E. Amorim, E. V. Brazil, J. P. Mena-Chalco, L. Velho, L. G. Nonato, F. F. Samavati, and M. C. Sousa, "Facing the high-dimensions: Inverse projection with radial basis functions," *Computers & Graphics*, pp. 35–47, 2015.
- [149] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philos. Trans. R. Soc. Lond. B. Biol. Sci*, pp. 537–547, 2003.
- [150] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," *IEEE Int. Conf. Robot. Autom. 2009. ICRA '09*, pp. 763–768, 2009.
- [151] F. Dimeas, F. Fotiadis, D. Papageorgiou, A. Sidiropoulos, and Z. Doulgeri, "Towards progressive automation of repetitive tasks through physical human-robot interaction," in *Human Friendly Robotics* (F. Ficuciello, F. Ruggiero, and A. Finzi, eds.), (Cham), pp. 151–163, Springer International Publishing, 2019.
- [152] H. Baltzakis, M. Pateraki, and P. Trahanias, "Visual tracking of hands, faces and facial features of multiple persons," *Machine Vision and Applications*, pp. 1–17, 2012.
- [153] M. Sigalas, M. Pateraki, and P. Trahanias, "Robust articulated upper body pose tracking under severe occlusions," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Chicago, USA), Sept. 2014.
- [154] S. Piperakis, E. Orfanoudakis, and M. Lagoudakis, "Predictive Control for Dynamic Locomotion of Real Humanoid Robots," in *Intel. Robots and Systems (IROS 2014), 2014 IEEE/RSJ Intern. Conf. on*, pp. 4036–4043, 2014.

- [155] G. Bradski, “opencv toolbox,” *Dr. Dobbs's Journal of Software Tools*, 2000.
- [156] M. Maniadakis, E. Hourdakis, M. Sigalas, S. Piperakis, M. Koskinopoulou, and P. Trahanias, “Time-aware multi-agent symbiosis,” *Submitted for publication in Journal Frontiers in Robotics and AI*, Oct. 2019.
- [157] R. Vuga, B. Nemec, and A. Ude, “Speed adaptation for self-improvement of skills learned from user demonstrations,” *Robotica*, pp. 2806–2822, 2016.
- [158] S. Calinon, A. Pistillo, and D. G. Caldwell, “Encoding the time and space constraints of a task in explicit duration hidden Markov model,” in *Proc. IEEE/RSJ Intl Conf. on Intel. Robots and Systems (IROS)*, (San Francisco, CA, USA), pp. 3413–3418, September 2011.
- [159] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, “Dynamical movement primitives: Learning attractor models for motor behaviors,” *Neural Computation*, vol. 25, pp. 328–373, 2013.
- [160] L. Rozo, P. Jimenez, and C. Torras, “A robot learning from demonstration framework to perform force-based manipulation tasks,” *Intel. Service Robotics*, pp. 33–51, 2013.
- [161] M. Ewerton, G. Maeda, G. Neumann, V. Kisner, G. Kollegger, J. Wiemeyer, and J. Peters, “Movement primitives with multiple phase parameters,” May 2016.
- [162] M. Guo, C. P. Bechlioulis, K. J. Kyriakopoulos, and D. V. Dimarogonas, “Hybrid control of multiagent systems with contingent temporal tasks and prescribed formation constraints,” *IEEE Transactions on Control of Network Systems*, vol. 4, pp. 781–792, Dec 2017.
- [163] R. Dechter, I. Meiri, and J. Pearl, “Temporal constraint networks,” *Artificial Intelligence*, pp. 61–95, 1991.
- [164] P. Morris, “Dynamic controllability and dispatchability relationships,” in *Integration of AI and OR Techniques in Constraint Programming*, Lecture Notes in Computer Science, pp. 464–479, Springer Intern. Publishing, 2014.
- [165] D. Dubois and H. Prade, *Possibility Theory. An approach to Computerized Processing of Uncertainty*. Plenum Press, New York, 1988.
- [166] B. Khadar, A. Rajesh, R. Madhusudhan, M. V. Ramanaiah, and K. Karthikeyan, “Statistical optimization for generalised fuzzy number,” *Int. Journal of Modern Engineering Research*, pp. 647–651, 2013.

- [167] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Control, planning, learning and imitation with dynamic movement primitives," in *IROS*, pp. 1–21, Max-Planck-Gesellschaft, 2003.
- [168] J. Aleotti and S. Caselli, "Robust trajectory learning and approximation for robot programming by demonstration," *Robotics and Autonomous Systems*, vol. 54, pp. 409 – 413, 2006. The Social Mechanisms of Robot Programming from Demonstration.
- [169] A. Shon, D.B. Grimes, C. Baker, and R. P. N. Rao, "A probabilistic framework for model-based imitation learning," *Proc. Twenty-Sixth Annu. Conf. Cogn. Sci. Soc.*, pp. 1237–1242, 2004.
- [170] A. Fod, M. J. Matarić, and O. C. Jenkins, "Automated derivation of primitives for movement classification," *Autonomous Robots*, vol. 12, pp. 39–54, Jan 2002.
- [171] H. Bruyninckx, "Open robot control software: the orocos project," in *Proceedings 2001 ICRA. IEEE Intern. Conf. on Robotics and Automation*, vol. 3, pp. 2523–2528, May 2001.
- [172] G. Magistris, A. Munawar, and P. Vinayavekhin, "Teaching a robot pick and place task using recurrent neural network," 12 2016.
- [173] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," *IEEE Intern. Conf. on Robotics and Automation (ICRA)*, pp. 1–8, 2018.
- [174] D. Michel, C. Panagiotakis, and A. A. Argyros, "Tracking the articulated motion of the human body with two rgb-d cameras," *Machine Vision Applications*, vol. 26, no. 1, pp. 41–54, 2015.
- [175] D. Michel, A. Qammar, and A. Argyros, "Markerless 3d human pose estimation and tracking based on rgb-d cameras: an experimental evaluation," in *Conference: International Conference on Pervasive Technologies Related to Assistive Environments (PETRA 2017)*, 06 2017.
- [176] "An exploratory study on the automated sorting of commingled recyclable domestic waste," *Procedia Manufacturing*, vol. 11, pp. 686 – 694, 2017. 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27–30 June 2017, Modena, Italy.
- [177] F. You, M. Mende, D. Štogl, B. Hein, and T. Kröger, "Model-free grasp planning for configurable vacuum grippers," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4554–4561, 2018.

