UNIVERSITAT POLITÈCNICA DE CATALUNYA

Doctoral Programme

AUTOMATIC CONTROL, ROBOTICS AND COMPUTER VISION

Ph.D. Thesis

ROBOT LEARNING FROM DEMONSTRATION OF FORCE-BASED MANIPULATION TASKS

Leonel Dario Rozo Castañeda

Advisors:

Carme Torras and Pablo Jiménez

May 2013

Universitat Politècnica de Catalunya

Doctoral Programme:
Automatic Control, Robotics and Computer Vision

This thesis has been completed at:
Institut de Robòtica i Informàtica Industrial, CSIC-UPC

Advisors:
Carme Torras and Pablo Jiménez

Author:
Leonel Dario Rozo Castañeda

To my dearly beloved parents Mery and Leonel, and my precious sister Jakeline. You are my main motivation.

“Nothing in the world can take the place of persistence. Talent will not; nothing is more common than unsuccessful men with talent. Genius will not; unrewarded genius is almost a proverb. Education will not; the world is full of educated derelicts. Persistence and determination alone are omnipotent”

Calvin Coolidge

Acknowledgements

Along this research road, I have met wonderful people who played a very important role in achieving this goal. First of all, I would like to thank my advisors Carme Torras and Pablo Jiménez for their continuing encouragement, coordination, supervision and understanding help. Carme and Pablo, you are wonderful people, I learned from you not only aspects about research on Robotics, but also life lessons, which helped me to become a better researcher and man. Thanks to you I had the honor and pleasure to work at the *Institut de Robòtica i Informàtica Industrial*. I did enjoy and love working at such a good research institute, where I met really nice, unforgettable and endearing people. Another thanks goes to my friends María L. Sarmiento, Michael Villamizar, Diego Pardo, Agustin Ortega, Ernesto Teniente, Gonzalo Ferrer, Farzad Husain and Patrick Grosh, you made these five years and a half memorable, funny and pleasant! Thanks for that! Special thanks to Anaís Garrell, your support, words and encouragement helped me to better manage those times when things do not go as expected.

I am also very grateful to Sylvain Calinon, who leaded my research stay at the *Istituto Italiano di Tecnologia*, his novel ideas and concepts about robot learning were really valuable and constructive. I had great moments during the time I spent there, and also found a very kind atmosphere. All this because of friendly people I met in Italy, thanks Davide de Tommaso, Tohid Alizadeh, Milad Malekzadeh, Marco San Biagio, Simona Ullo, and Arash Ajoudani.

Luces, you have been by my side through the end and hardest part of this long way. Your endless love, unconditional support and encouraging words have always been what I have needed day by day. Not everyone is as lucky

as me to be blessed with a woman like you. Thank you so much for coming into my life and standing by my side through thick and thin. Te amo!

Finally, I would like to thank my close colombian friends, parents and sister: JuanK, Fercho y Jeison, gracias por su amistad incondicional, siempre me han acompañado en los momentos más importantes de mi vida. Pa', tu ejemplo de superación, de trabajo y responsabilidad fueron claves para poder llegar hasta aquí, gracias por ello! Jakeline, tu mayor enseñanza en mi vida ha sido la nobleza y rectitud para enfrentar la vida, eso sin duda lo he llevado siempre en mi mente durante este periodo lejos de todos. Ma', eres el motor de mi vida, una fuente de motivación y enseñanzas diarias. Gracias por tu apoyo, comprensión, paciencia y palabras de aliento que me has brindado durante toda mi vida, pero sobretodo a lo largo de este sueño que hoy culmino. Gran parte de lo que soy hoy en día te lo debo a ti. Las palabras siempre se quedaran cortas para expresarte lo que significas para mi. Los amo y este triunfo se los dedico especialmente a ustedes, mi familia.

Moltes gracies Barcelona! Grazie mille Genova!

Abstract

One of the main challenges in Robotics is to develop robots that can interact with humans in a natural way, sharing the same dynamic and unstructured environments. Such an interaction may be aimed at assisting, helping or collaborating with a human user. To achieve this, the robot must be endowed with a cognitive system that allows it not only to learn new skills from its human partner, but also to refine or improve those already learned. In this context, learning from demonstration appears as a natural and user-friendly way to transfer knowledge from humans to robots. This dissertation addresses such a topic and its application to an unexplored field, namely force-based manipulation tasks learning. In this kind of scenarios, force signals can convey data about the stiffness of a given object, the inertial components acting on a tool, a desired force profile to be reached, etc. Therefore, if the user wants the robot to learn a manipulation skill successfully, it is essential that its cognitive system is able to deal with force perceptions.

The first issue this thesis tackles is to extract the input information that is relevant for learning the task at hand, which is also known as the *what to imitate*? problem. Here, the proposed solution takes into consideration that the robot actions are a function of sensory signals, in other words the importance of each perception is assessed through its correlation with the robot movements. A Mutual Information analysis is used for selecting the most relevant inputs according to their influence on the output space. In this way, the robot can gather all the information coming from its sensory system, and the perception selection module proposed here automatically chooses the data the robot needs to learn a given task.

Having selected the relevant input information for the task, it is necessary to represent the human demonstrations in a compact way, encoding the relevant characteristics of the data, for instance, sequential information, uncertainty, constraints, etc. This issue is the next problem addressed in this thesis. Here, a probabilistic learning framework based on hidden Markov models and Gaussian mixture regression is proposed for learning force-based manipulation skills. The outstanding features of such a framework are: *(i)* it is able to deal with the noise and uncertainty of force signals because of its probabilistic formulation, *(ii)* it exploits the sequential information embedded in the model for managing perceptual aliasing and time discrepancies, and *(iii)* it takes advantage of task variables to encode those force-based skills where the robot actions are modulated by an external parameter. Therefore, the resulting learning structure is able to robustly encode and reproduce different manipulation tasks.

After, this thesis goes a step forward by proposing a novel whole framework for learning impedance-based behaviors from demonstrations. The key aspects here are that this new structure merges vision and force information for encoding the data compactly, and it allows the robot to have different behaviors by shaping its compliance level over the course of the task. This is achieved by a parametric probabilistic model, whose Gaussian components are the basis of a statistical dynamical system that governs the robot motion. From the force perceptions, the stiffness of the springs composing such a system are estimated, allowing the robot to shape its compliance. This approach permits to extend the learning paradigm to other fields different from the common trajectory following. The proposed frameworks are tested in three scenarios, namely, *(a)* the ball-in-box task, *(b)* drink pouring, and *(c)* a collaborative assembly, where the experimental results evidence the importance of using force perceptions as well as the usefulness and strengths of the methods.

Resumen

Uno de los principales retos en Robótica es crear robots que puedan interactuar con seres humanos de una forma natural compartiendo el mismo entorno, el cual puede ser altamente dinámico y no estructurado. Dicha interacción podría estar enfocada a asistir, ayudar o colaborar en ciertas tareas con un usuario humano. Para alcanzar dicho objetivo, el robot debe estar dotado de un sistema cognitivo que le permita no solo aprender nuevas habilidades, sino también refinar o mejorar aquellas ya aprendidas con anterioridad. En este contexto, el aprendizaje por demostración aparece como un método intuitivo por el cual un humano puede transferir conocimiento a un robot. Esta disertación se basa en dicha idea y su aplicación a un campo poco explorado: el aprendizaje de tareas de manipulación basadas en fuerza. En esta clase de escenarios, las señales de fuerza pueden transmitir información sobre la rigidez de un objeto, las componentes inerciales que actúan sobre una herramienta, una referencia de fuerza a seguir, etc. Por lo consiguiente, si el usuario desea que el robot aprenda una tarea de manipulación satisfactoriamente, es esencial que su sistema cognitivo sea capaz de trabajar con percepciones hápticas.

El primer punto que esta tesis abarca es la extracción de información relevante para el aprendizaje a partir del conjunto de datos de entrada, lo cual se conoce como el problema de *¿Qué imitar?*. La solución que se propone aquí considera que las acciones del robot son función de las señales sensoriales, en otras palabras, la importancia de cada percepción es evaluada a través de su correlación con los movimientos del robot. Un análisis de información mutual es utilizado para seleccionar las entradas más relevantes de acuerdo a su influencia sobre las variables de salida del problema. De este modo, el robot puede reunir toda la información proveniente de su sistema

sensorial, la cual será analizada por un modulo de selección de percepciones con el fin de escoger automáticamente los datos que el robot necesita para aprender una tarea especifica.

Una vez seleccionada la información de entrada relevante para la tarea, es necesario representar en una forma compacta las demostraciones dadas por un usuario, codificando las características más importantes, por ejemplo, información secuencial, incertidumbre, restricciones, etc. Este aspecto es el siguiente problema que se aborda en esta tesis. Se propone entonces una estructura de aprendizaje probabilística basada en el modelo oculto de Markov y en la regresión por mezclas de Gaussianas, para aprender habilidades de manipulación basadas en fuerzas. Las propiedades más destacadas de dicha estructura son: *(i)* es capaz de trabajar con señales de fuerza ruidosas y con incertidumbre gracias a su naturaleza probabilística, *(ii)* aprovecha la información secuencial dada por el modelo de Markov con el fin de resolver el "aliasing" sensorial y las discrepancias temporales, y *(iii)* explota las variables de la tarea para codificar aquellas habilidades basadas en fuerza en las cuales las acciones del robot sean moduladas por un parámetro externo. Por lo tanto, el sistema de aprendizaje resultante es capaz de codificar y reproducir de forma robusta diferentes tareas de manipulación.

Finalmente, esta tesis va un paso más allá proponiendo una estructura novedosa para el aprendizaje por demostración de comportamientos basados en impedancia. Los aspectos claves de dicho sistema son la combinación de información de fuerza y visión para codificar las demostraciones de forma compacta, y la capacidad de modular el nivel de "compliance" del robot a lo largo de la tarea. Lo anterior se logra por medio del uso de un modelo probabilístico paramétrico, cuyas componentes Gaussianas son la base de un sistema dinámico que gobierna el movimiento del robot. A partir de las percepciones hápticas, la rigidez de los muelles que componen el sistema dinámico es estimada, permitiendo de esta forma modular la rigidez resultante del robot. Este enfoque permite extender el paradigma de aprendizaje a otros campos diferentes a los enfoques que tratan el problema típico de codificación de trayectorias cinemáticas. Las estructuras de aprendizaje

propuestas son evaluadas en tres escenarios, a saber, *(a)* la tarea bola-en-caja, *(b)* servir bebidas, y *(c)* ensamble colaborativo. Los resultados experimentales evidencian la importancia de utilizar percepciones de fuerza, así como también la utilidad y fortalezas de los métodos planteados.

Resum

Un dels reptes en Robòtica és la creació de robots capaços d'interactuar amb els humans de manera natural i compartint el mateix entorn, el qual pot ésser altament dinàmic i no estructurat. Aquesta interacció pot estar enfocada per a assistir, ajudar o col·laborar en tasques concretes amb un usuari. Per tal d'assolir el citat objectiu, el robot ha d'estar dotat d'un sistema cognitiu que permeti aprendre noves habilitats, i alhora ha de ser capaç de refinar o millorar aquelles que han estat prèviament apreses. En aquest context, l'aprenentatge per demostració apareix com un mètode intuïtiu amb el qual l'humà pot transferir coneixement a un robot. Aquesta dissertació es basa en l'esmentada idea, a més, la seva aplicació es presenta en un camp poc explorat: l'aprenentatge de tasques de manipulació basades en forces. En aquest tipus d'escenaris, les senyals de força poden transmetre informació sobre la rigidesa d'un objecte, les components inercials que actuen sobre una eina, una referència de força que ha de ser seguida, etc. Conseqüentment, si l'usuari desitja que el robot aprengui una tasca de manipulació satisfactòriament, és essencial que el seu sistema cognitiu sigui capaç de treballar amb percepcions hàptiques.

El primer punt que aquesta tesi comprèn és l'extracció d'informació rellevant per a l'aprenentatge a partir d'un conjunt de dades d'entrada, el què es coneix com el problema *Què imitar?*. La solució que es proposa en aquest treball és considerar que les accions del robot són funcions de les senyals sensorials, en altres termes, la importància de cada percepció és avaluada mitjançant la seva correlació amb els moviments del robot. Un anàlisi d'informació mútua és utilitzat per a seleccionar les entrades més rellevants d'acord a la seva influència sobre les variables de sortida del problema. D'aquesta manera, el robot pot recopilar tota la informació procedent

del seu sistema sensorial, la qual serà analitzada mitjançant un mòdul de selecció de percepcions amb la finalitat d'escollir automàticament les dades que el robot necessita per a aprendre una tasca específica.

Havent seleccionat la informació d'entrada rellevant per a la tasca, és necessari representar d'una forma compacta les demostracions donades per l'usuari, codificant les característiques més importants, per exemple, informació seqüencial, incertesa, restriccions, etc. Aquest aspecte és el següent problema que s'estudia en aquesta tesi. Conseqüentment, s'estudia una estructura d'aprenentatge probabilística basada en el model ocult de Markov i en la regressió per mescla de Gaussianes, per tal d'aprendre habilitats de manipulació basades en forces. Les propietats més destacades de la citada estructura són: *(i)* és capaç de treballar amb senyals de força amb soroll i incertesa gràcies a la seva naturalesa probabilística, *(ii)* aprofita la informació seqüencial donada pel model de Markov amb la fi de resoldre el "aliasing" sensorial i les discrepàncies temporals, i *(iii)* explota les variables de la tasca per a codificar aquelles habilitats basades en força, amb les quals les accions del robot són modulades per un paràmetre extern. Per tant, el sistema d'aprenentatge resultant és capaç de codificar i reproduir de forma robusta diferents tasques de manipulació.

Finalment, aquesta tesi doctoral va un pas més enllà proposant una estructura innovadora per a l'aprenentatge per demostració de comportaments basats en impedància. Els aspectes claus del citat sistema són la combinació d'informació de forces i visió per a codificar les demostracions de forma compacta, i la capacitat de modular el nivell de "compliance" del robot durant la realització de la tasca. Tot això s'aconsegueix per mitjà de l'ús d'un model probabilístic paramètric, les components del qual són Gaussianes, aquestes són la base del sistema dinàmic que governa el moviment del robot. A partir de les percepcions hàptiques, la rigidesa dels molls que componen el sistema dinàmic és estimada, permetent d'aquesta manera modular la rigidesa resultant del robot. Aquest enfocament permet estendre el paradigma d'aprenentatge a altres camps diferents, els quals tracten els problemes típics de codificació de trajectòries cinemàtiques. Les estructures

d'aprenentatge proposades són avaluades en tres escenaris diferents, *(a)* la tasca pilota-en-caixa, *(b)* servir begudes, i *(c)* acoblament col·laboratiu. Els resultats experimentas evidencien la importància d'utilitzar percepcions de força, així com també la utilitat i fortalesa dels mètodes plantejats.

Contents

List of Figures

Chapter 1

Introduction

You open the fridge and, by only lifting slightly the Tetra Brik, you already know whether it contains enough milk for your breakfast, and how carefully you have to pour the liquid into the bowl. Think of all the force-based manipulations involved in inserting a key in the keyhole and opening the lock, or the push-and-pull needed to close the drawer of an old chest of drawers. Visualize yourself moving together with a friend a piece of furniture across a cluttered environment. In all these cases, force-based sensory information is crucial for the successful completion of the task. Force signals can convey data about the stiffness of a given object, the inertial components acting on a tool, a desired force profile to be reached, etc. Therefore, when a human wants to teach a robot such manipulation skills from examples, it is necessary to endow the robot with a learning framework able to exploit the force information generated during the execution of the task. This thesis analyzes this problem and proposes a set of computational structures appropriate to learn and reproduce force-based manipulation tasks. All the work is performed in the context of three scenarios, where manipulation is the central theme.

This chapter presents the motivations and objectives of this thesis in Sections 1.1 and 1.2, respectively. The different learning frameworks developed along this thesis are described in Section 1.3, and the three experimental scenarios are shown in Section 1.4. Lastly, Sections 1.5 and 1.6 respectively explain the contributions of this research and the organization of this dissertation.

1.1 Motivation

One of the main concerns in Robotics is how robots may become our companions and inhabit our environments in the most “natural” way. Such target is still far, but one of the routes to achieve it is to endow robots with safe and user-friendly interaction skills, so that they can behave in such a way humans can feel comfortable sharing their surroundings with them. This involves to have robots with cognitive capabilities allowing them to understand, analyze and react according to the interaction with a human [34, 36]. In this context, learning mechanisms are needed to acquire knowledge from such an interaction process. Learning from demonstration (**LfD**) is one approach in which a robot can learn a specific task from human examples (e.g., imitating human gestures [23], pouring a drink [154], lifting an object [50], etc.).

While most works in the field have focused on learning the kinematics of motions, little work concerning force-based skills has been done. Force signals are crucial when contact with the environment takes place, mostly in manipulation of objects [108], and physical interaction with a partner [130]. Research has shown that the human central nervous system is composed of internal models that control the interactions between the body and its surroundings [107, 165]. Some of these models are dedicated to predicting the outcome or anticipating force resulting from an individual’s conscious action. Inspired by the relevance of force information in the human performance of tasks, the core of this thesis is to study and analyze how robots may learn force-based tasks from human demonstrations.

It is worth noting that depending exclusively on one perception channel greatly limits the information that a robot may be provided with during the execution of any task. Humans are endowed with a very rich multimodal perception system [57], and in contrast, research on LfD has focused on learning tasks using unimodal sensing. Taking inspiration from these facts, this thesis also proposes to merge vision and haptic information in order to improve the learning process by exploiting the different kinds of data provided by distinct sensory channels. Note that when performing physical collaboration tasks, humans communicate strongly and often subtly via multiple channels like gaze, speech, gestures, movement and posture [113, 152]. Therefore, robots jointly working with humans may widely benefit from this sensory combination.

Recent progress in physical human-robot interaction (**pHRI**) showed that active and safe workspace sharing is possible in principle. Encouraged by these results, researchers have been focused on encompassing safety issues based on biomechanical analysis, human-friendly hardware design, and interaction control strategies [4, 64]. Nevertheless, it is equally relevant to develop and validate cognitive key components that enable robots to track, understand and predict human motions in a weakly structured dynamic environment in real-time. In this sense, humans may also teach through examples a robot to carry out a collaborative task, where not only trajectories are important, but also the role of the robot (e.g., leader or follower) and its behaviors (e.g., reactive or proactive). In this context, this thesis addresses an interesting and challenging issue regarding impedance-based behaviors learning in pHRI scenarios, where the new control schema of recently developed torque-controlled robots can be exploited. Here, the reformulation of learning algorithms as well as multimodal perception systems are crucial to achieve such a goal.

1.2 Objectives

According to the current state of the research in LfD and the gaps in this field to build safe, smart and user-friendly robots, the problems that this thesis aims to solve are summarized below:

1. **To identify the relevant features in force-based manipulation tasks from sensory information with the aim of including them as inputs in the learning stage.**
   Although the learning algorithms might work successfully using input training data with redundant and irrelevant variables, it is useful and more suitable to recognize those perceptions that are relevant in manipulation tasks with the aim of reducing the complexity of the problem and data dimensionality as well as improving the performance in the learning stage. The identification of relevant input variables works in the direction of answering the central question of *What to imitate?*

2. **To propose extendable end-to-end LfD frameworks for teaching force-based manipulation tasks to robots.**

The characteristics of force-based skills will be analyzed and further on taken into account for selecting the learning methods best suited to deal with force signals. This will permit to build entire learning frameworks able to encode and reproduce a large set of manipulation tasks, so that future researchers can use and/or modify them easily.

3. **To merge haptic and visual data by exploiting the unique information that each perception channel provides, and taking advantage from wisely combining them to obtain richer and more precise state and task descriptions.**
   Haptic and vision data mostly provide distinct types of information when a robot is carrying out a manipulation task. In this sense, it is quite relevant to exploit such information for learning a larger range of skills without increasing significantly the complexity of the encoding and reproduction methods.

4. **To extend the LfD paradigm for encoding and reproducing impedance-based robot behaviors.**
   The learning frameworks that have been developed for precise reproduction of reference trajectories need to be re-thought and adapted to the new scenarios and tasks where the recently developed torque-controlled robots may work. Force sensing and stiffness estimation can be exploited for learning and reproducing a different kind of skills in pHRI, namely, impedance-based behaviors.

1.3 Overview of the proposed learning frameworks

Along the development of this thesis, three learning frameworks have been proposed in order to provide a whole structure to be used for encoding and reproducing force-based tasks with distinct characteristics. The next chapters are aimed at explaining in detail each module of these frameworks, justifying the selection of each method and algorithm, and showing their performance when dealing with haptic perceptions. Brief descriptions and illustrations of the proposed frameworks are given next.

The first learning framework is shown in Figure 1.1. Here, the first module is aimed at *processing* the data coming from the force sensor (details in Section 3.1), which is relevant when the robot manipulates objects and also when its signals are fed back to

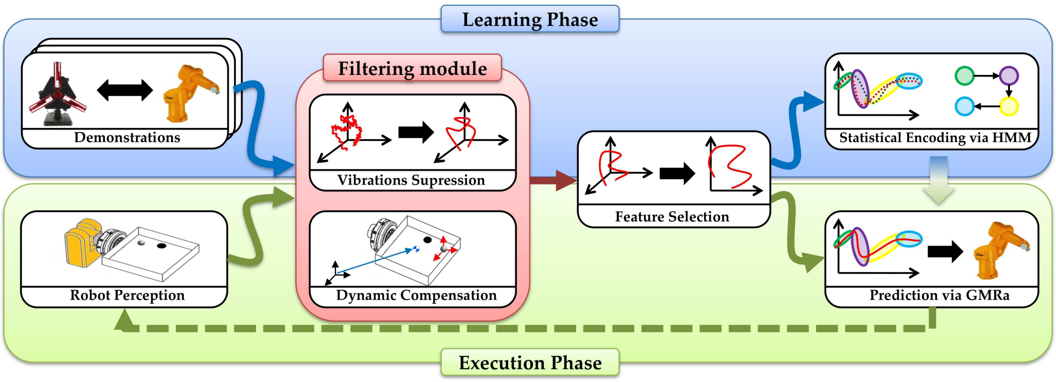


Figure 1.1: Learning framework based on hidden Markov models and Gaussian mixture regression able to deal with perceptual aliases.

the teacher, for instance, through a haptic device (see Sections 1.4.1 and 1.4.2). The processed signals are then analyzed by the *feature selection* module. This is in charge of selecting the most relevant input variables for the problem at hand through a Mutual Information analysis, as explained in Section 3.3. After this, in the learning phase, a hidden Markov model (**HMM**) encodes the resulting training data. The obtained probabilistic model is then used at the reproduction phase along with a modified version of Gaussian mixture regression (**GMR**) for computing the desired control commands to be sent to the robot. These two phases are described in Section 4.3.1 and 4.3.2, respectively. This whole structure is suitable to learn simple force-based manipulation tasks, it exploits the sequential information implicit in the training data and it is also able to deal with perceptual aliases (i.e., those tasks with an underlying multivalued function behavior).

Figure 1.2 illustrates the second proposed framework, which is basically an extension of the one previously described. The main difference lies in the concept of *task variables modulation* (also known as task parametrization), where the main idea is that the robot actions now also depend on a specific parameter of the task. In this sense, this framework exploits such task variables by encoding the demonstrations through a parametric version of the classic HMM (details in Section 4.4). Therefore, this framework keeps the same advantages provided by its predecessor, but it is now able to learn force-based skills that are modulated by a given parameter. Hence, this improved version can learn a larger range of force-based tasks.

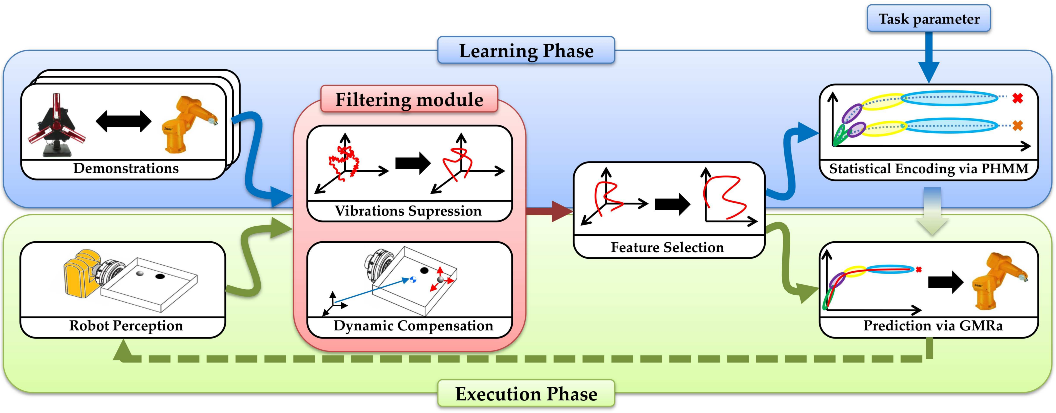


Figure 1.2: Extension of the framework shown in Figure 1.1, where force-based parameters are exploited for encoding purposes.

Note that the previous frameworks do not take into consideration other kind of sensory information different from force signals. This aspect might limit the range of manipulation tasks where these approaches may be applied to, for instance, those where vision data are needed. Hence, the third proposed framework shown in Figure 1.3 copes with this problem by merging vision and haptic information through a task-parametrized probabilistic model, as explained in Section 5.1. This novel learning structure takes inspiration from the task parametrization used in the previous framework (Figure 1.2) in order to modulate a model encoding force-based skills with parameters given by a vision sensory system. Such model is then used to represent a statistical dynamical system, which is aimed at encoding impedance-based behaviors by shaping the robot impedance through the *stiffness estimation* module (see Section 5.2). The whole learning framework thus allows to extend the force-based learning to other type of scenarios, e.g., collaborative tasks (see Section 1.4.3).

1.4 Experimental scenarios

To test the different proposed learning frameworks, three experimental setups were constructed. The first two structures (Figures 1.1 and 1.2) were used to teach a robotic manipulator to carry out two distinct manipulation tasks using exclusively haptic data. In both scenarios a human user holding the end-effector of a 6-DoF haptic interface (Delta device from Force Dimension [59]) teleoperates a robotic arm (RX60

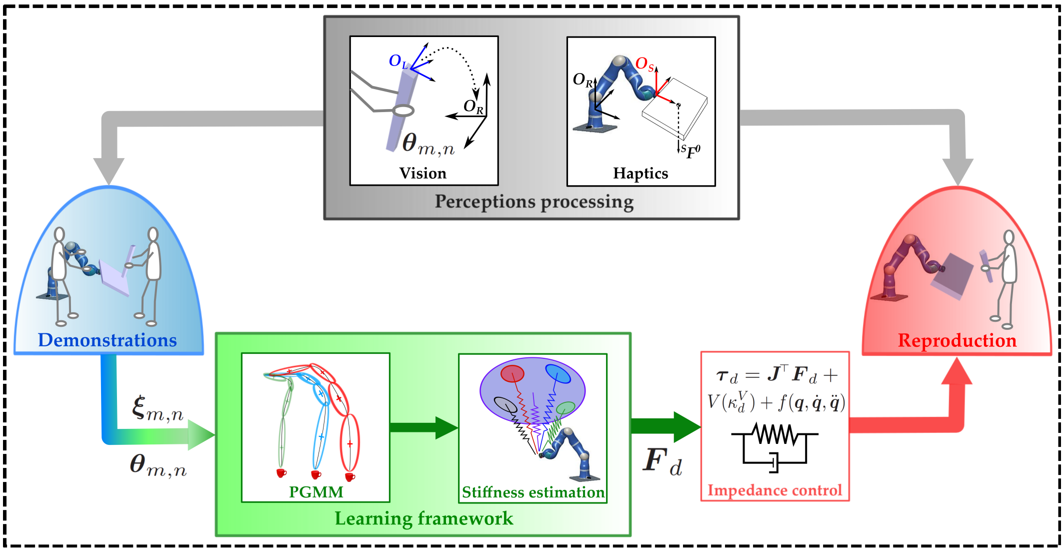


Figure 1.3: Learning framework combining vision and force information for encoding impedance-based behaviors.

from Stäubli) which has a force-torque sensor (Schunk FTC-050) placed on its wrist. The description of these tasks is given below. The third experimental setup aimed at testing the learning structure shown in Figure 1.3 is introduced in Section 1.4.3.

1.4.1 Ball-in-box task

In this task the robot holds a plastic container with a steel sphere inside it, as shown in Figure 1.4. At the demonstration phase, the teacher repeatedly carries out the task to be learned, which consists of taking the ball out of the box through the hole, following a specific motion strategy: *Starting at some predefined initial positions, the ball is driven towards the wall adjacent to the hole, and then forced to roll along this wall to the hole* (see bottom right box in Figure 1.4). During the demonstrations, the teacher feels at the end-effector of the haptic device the force-torque sensed at the robotic wrist. Also, note that the user has an additional information source by watching the scene directly. No visual data are provided to the robot. It is worth highlighting that this particular manipulation task has been chosen because it is well-defined and simple enough to permit analyzing each stage of the proposed LfD frameworks separately and in depth.

The inputs of this manipulation task are defined as the wrench $\boldsymbol{\vartheta} = \{F_x\ F_y\ F_z\ T_x\ T_y\ T_z\}$,

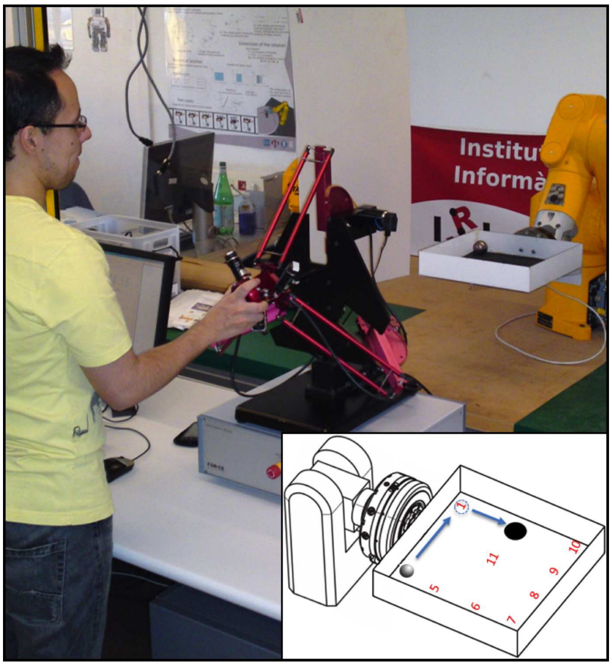


Figure 1.4: Experimental scenario of the *ball-in-box* task.

i.e., the sensed forces and torques in the robot's frame, and the outputs correspond to the joint velocities of the robot defined by $\boldsymbol{\omega} = \{\omega_1 \ldots \omega_{N_q}\}$, where N_q is the number of joints of the robot. Despite the task to be learned seems simple at first sight, the entire process implies to solve several technical and research issues. Regarding the acquisition of suitable training data, first it is necessary to take into account that the box is not a rigid structure and it vibrates when the robot moves. Second, it is important to provide the teacher with focused haptic feedback so that it does not distract him/her from task demonstrations. Note that these are general issues since robot tools, grippers and hands are a source of noise that affects the force-torque sensor readings. Moreover, it is also important to consider the possible delays that typical teleoperation systems undergo [118]. In this sense, the position-force architecture of the experimental setting works at a frequency of 1000 Hz, which showed to be stable according to the task requirements and also provided high fidelity force reflection to the user.

This task was used to analyze how several learning algorithms behave in the force domain, and subsequently it was also used to evaluate the performance of the first proposed framework (Figure 1.1). Results regarding the feature selection process, encoding and reproduction of this task are given in Sections 3.4.1, 4.3.3.1 and 4.3.3.2, respectively.

1.4.2 Pouring task

The second task consists of pouring drinks. Here, the robotic arm holds a 1 liter plastic bottle full of tiny metallic spheres, which play the role of a fluid (this solution was adopted to avoid spilling liquid during tests and, in the end, the goal to learn is a given fluid-like dynamics, no matter which). The teacher teleoperates the robot in order to demonstrate how to pour 100 ml drinks into a plastic glass. In contrast to the illustrative example shown in Figure 1.5 where the human moves its arm in order to pour into all the glasses placed at different location, here every sample of the task starts from a unique predefined initial pose of the bottle, which is also the stop configuration once the robot has poured a drink. Initially, the bottle is completely full, and the teacher carries out several demonstrations until the bottle is empty. Thus, the initial force-torque values for each example vary according to how much "fluid" has been poured previously. Note that in a real situation, a human carrying out the same task must turn the bottle in such a way that the fluid is poured, which definitely depends on the quantity of fluid inside it, as illustratively shown in Figure 1.6. It is worth to highlight that such changes in the input variables at the beginning of the demonstrations are similar to those observed in the *ball-in-box* task for each initial position of the sphere inside the container.

Note that in this task, the teacher is also able to watch the scene directly, thus he/she can know the location of the glass in the robot workspace. Such information is not provided to the robot during the execution phase because the glass position is predefined in advance and fixed across the examples.[1] The input variables also are the wrench $\boldsymbol{\vartheta}$, but the output variables are the joint robot position defined by $\boldsymbol{q} = \{ q_1 \ldots q_{N_q} \}$ at instant $t+1$ for the given $\boldsymbol{\vartheta}$ at t. Such change of output variables for this task is aimed at showing the generic significance of the second proposed learning framework (Figure 1.2) for different representations of the task state.

[1] Note that a camera system may also be used to know the location of the glass in the robot frame, so that the demonstrations would also be dependent on this parameter.

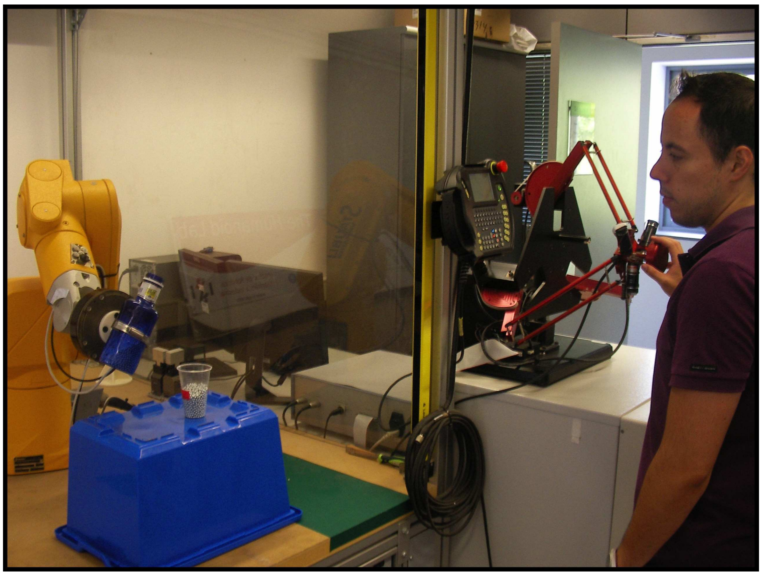


Figure 1.5: Experimental scenario of the *pouring task*.

It should be noted that the proposed task is challenging and has aroused the research community's interest recently. For instance, Tamosiunaite *et al.* [154] tackled the same problem using reinforcement learning, which was applied to improve the initial encoding obtained from human demonstrations modeled through dynamic motion primitives. Moreover, Cakmak and Thomaz [20] taught a humanoid robot to pour through an active learning framework, wherein the robot was allowed to ask questions regarding the task at hand. The proposed experimental setup and learning frameworks significantly differ from these works in that the human-robot interaction is through a haptic device, the demonstrations are encoded by a probabilistic model that exploits the task parameters and the perception system senses only the forces-torques generated over the execution of the skill. Section 3.4.2 explains the results of the feature selection process, while Sections 4.4.2.1 and 4.4.2.2 respectively present the learning and reproduction of this task.

1.4.3 Collaborative table assembly

In order to test the third proposed framework (Figure 1.3), a human-robot collaborative task is considered. Here, the robot's role is to hold a wooden table while the user's role is to screw the four legs to it. Figure 1.7 presents an example of assembly instructions

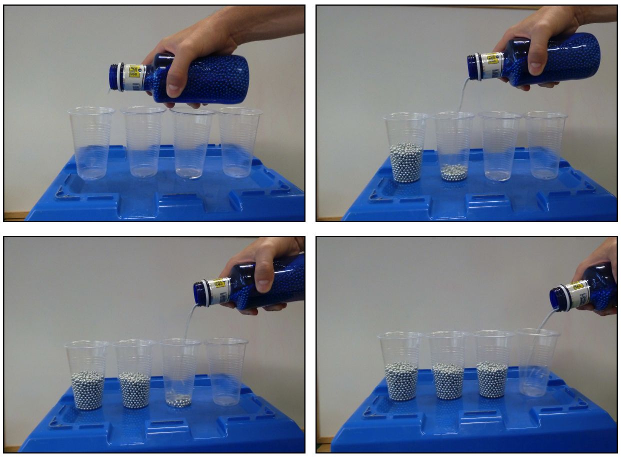


Figure 1.6: Illustration of pouring task carried out by a human in a real situation.

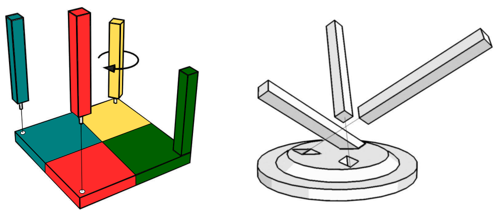


Figure 1.7: Assembly tasks characterized by different sequences, positions and orientations of components, with haptic and motion patterns that are specific to each item.

that can be found in "do it yourself" furniture catalogs. Here, two small tables require specific sequences of force and movement to get assembled. Learning such specificities is required for an efficient collaborative assembly. Instead of manually programming those specificities for each item, one would like the robot to extract those automatically from a set of demonstrations provided by two users collaborating together to assemble the different parts of the table as shown in Figure 1.8. After learning, the task can be reproduced by a single user, with the robot partner interacting appropriately with respect to the preferences of the user and the specificities of the item being assembled. Thus the information about the points of assembly is not provided to the robot, neither the different options, orientation of table legs, etc. The robot instead learns those specificities by being guided by one of the users through kinesthetic teaching.

A different experimental setup was built for this task. A KUKA lightweight 7-DoF robot (LWR) [3] is used here, which is controlled through the *Fast-Research Interface* [145], by using a Cartesian impedance controller. The position and orientation of the

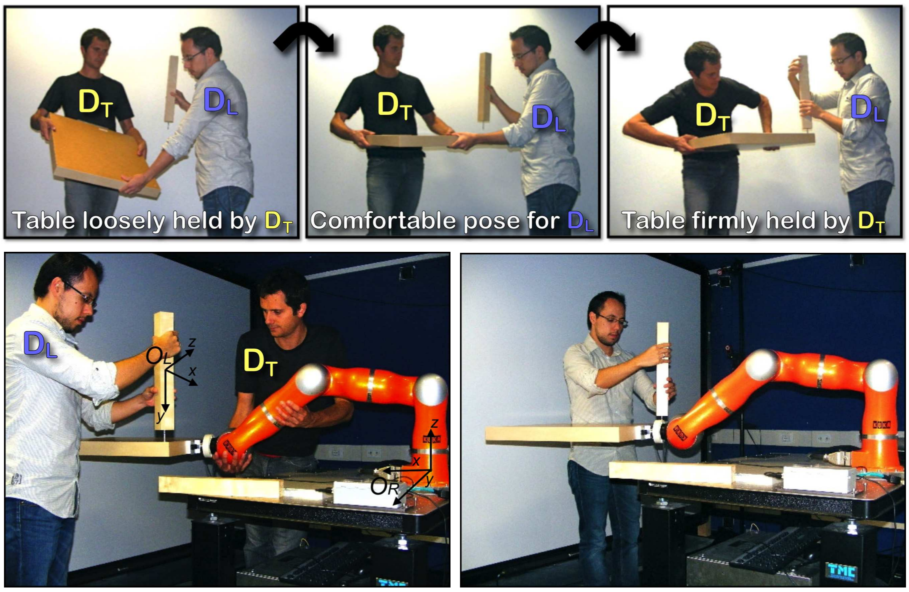


Figure 1.8: *Top*: Two humans assembling a wooden table. *Bottom*: (left) demonstration of the impedance-based behavior, (right) reproduction of the collaborative assembly task.

table legs are tracked with a marker-based NaturalPoint OptiTrack motion capture system that is composed of 12 cameras working at a rate of 30 frames per second. The robot is equipped with a six-axis force-torque sensor (ATI Mini45) attached between its wrist and the wooden table, measuring all the signals generated during the interaction of the human while moving the table and screwing the legs to it.

Regarding the task, two candidate frames of reference are considered in the experiment: the fixed robot frame $\mathcal{O}_R$ and the leg frame $\mathcal{O}_L$. It is assumed here that one leg is used and tracked at a time (after one leg has been assembled, the next leg is tracked). The collaborative scenario shown in Figure 1.8 consists of screwing the legs at the four corresponding positions on the table. D_T (i.e., the robot) is first compliant to allow D_L (i.e., the human user) to move the table freely in the workspace until comfortable position and orientation are found for the work to be performed next (compliant phase). When D_L grasps a leg and starts inserting it into one of the four screw threads in the table, D_T adopts a stiff posture, holding the table to facilitate D_L's part of the task (stiff phase).

It is worth to highlight that vision and haptic information combination is fundamental to this task. On the one hand, if only vision was used, it may occur that the leg was close enough to one of the screw threads but without being assembled. This would probably make D_T become stiff, wasting a high quantity of energy while D_L is occupied to precisely position the leg before the actual screwing process. Here, D_T should instead also regulate its stiffness in accordance with the sensed force pattern. On the other hand, if the D_T behavior was based only on force-based perceptions, the learning of the task could potentially fail because D_T would require to distinguish from noisy data what forces-torques correspond to interaction with D_L and which ones are produced by the leg screwing phases. This can be problematic because these patterns might be similar in some situations. Thus, this setting allows to stress the convenience of merging both perception channels with vision and haptic data coming together to help D_T and learn how its stiffness behavior should be shaped.

Note that vision information is used to define the task variables that modulate the probabilistic model while the force data compose the input vector $\boldsymbol{\vartheta}$. The output variables in the demonstration phase are the robot end-effector position $\boldsymbol{x}$ in $\mathcal{O}_R$. The robot is controlled at the execution phase through force commands obtained from a set of virtual spring systems (see Section 5.3.2). Results regarding the feature selection process, encoding and reproduction of this task are given in Sections 3.4.3, 5.3.1 and 5.3.2, respectively.

1.5 Contributions

The development of this thesis may be split in several phases. The first entails discovering the relevant information for learning a manipulation task, in other words, which perceptions are needed for encoding and reproducing a skill successfully. After this, the second step was to build a whole learning framework able to robustly encode and exploit force data. Lastly, the last stage of this research considered two challenging problems, namely, *(i)* merging vision and force information in a LfD framework, and *(ii)* learning impedance-based robot behaviors. These issues comprise the main contributions of this dissertation, which will be explained in detail next.

1. **Solving the *what to imitate?* issue using a Mutual Information feature selection approach**

In general terms, the robot skills are encoded by a policy representing a mapping between perceptions and actions, in other words, the robot output commands are conditioned by the input information coming from its sensory system. In this sense, a given input variable is more or less relevant for the task according to how much it influences the robot actions. Such idea is the basis of the approach presented in this thesis for solving the *what to imitate?* problem. Here, a selection algorithm based on Mutual Information analysis is proposed in order to choose the most relevant robot perceptions based on their correlation with the actions. Specifically, a conditional Mutual Information criterion is used, which allows not only to discover the most correlated input variables, but also to know how much information a specific perception provides with respect to a subset of already chosen inputs.

2. **Force-based LfD framework able to deal with perceptual aliases and parametrized skills**
   When a robot learns a task using force signals as input information, several issues need to be taken into account in order to design a framework appropriate for this kind of data (e.g., forces are noisy and may also show high time discrepancies). From a study of several learning algorithms, probabilistic methods showed to perform well in the force domain. This thesis then proposes a whole learning framework that deals with time discrepancies by exploiting the sequential information provided by hidden Markov models, which also allow to discern the correct robot action when perceptual aliasing occurs. The framework is then improved by switching the encoding model to its parametric version, which permits to learn skills that are modulated by a force-based parameter. The resulting framework is thus composed by an inputs selection module (previously described), an encoding phase carried out by a parametric hidden Markov model and a reproduction stage performed through modified Gaussian mixture regression that uses the sequential information embedded in the learning model.

3. **Merging vision and force information to compactly encode impedance-based behaviors**
   Several manipulation tasks are not exclusively based on force perceptions, but they also depend on other kind of information, for instance, that provided by

vision systems (i.e., location and orientation of objects of interest). Thus, it is interesting how these two sensory components can be merged in an optimal fashion, where the different data are exploited and combined for learning a skill. In this context, a parametrized Gaussian mixture model is used in this thesis in order to encode a force-based task where the variables modulating the model come from the vision system. Such encoding approach is then the basis to define a statistical dynamical system (i.e., a set of virtual springs) that controls the robot motion. Such representation allows to extend the LfD paradigm to other scenarios, for instance, those where the robot action is based on specific impedance-based behaviors, instead of merely following a given trajectory. Therefore, the proposed idea here is to shape the springs stiffness according to the vision and force inputs in such a way that the robot reaches different compliance levels according to the task requirements. The obtained novel learning structure is thus able to encode and reproduce impedance-based robot actions.

1.6 Organization

This thesis is structured in the following chapters:

- Chapter 2 presents the history and concepts of LfD. The most well-known learning algorithms as well as the different ways of transferring skills to a robot are also explained. Applications in the field are briefly described in this chapter.

- Chapter 3 shows how feature selection techniques can be used to select the most relevant perceptions to learn a given task. Specifically, the *Mutual Information* criterion is introduced here as a robust tool that analyzes how the robot perceptions influence its actions in order to carry out the selection process.

- Chapter 4 is aimed at analyzing how some state-of-the-art algorithms work on the force domain, and at proposing a compact learning framework able to learn skills from haptic inputs. Also, an extension of such framework is proposed to deal with *task variables*.

- Chapter 5 deals with *merging vision and force* in LfD, where instead of simply augmenting the observation vector, the input variables provided by the vision system are considered as task variables while the force data compose the input space.

- Chapter 6 presents the conclusions of thie thesis and summarizes the results achieved.

- Chapter 7 discusses new possible routes of research arising from the work presented in the previous chapters. Issues concerning robust temporal information encoding, impedance-based behaviors learning, role determination in pHRI using force information, among others, are discussed here.

Supplementary information concerning the list of publications of the author related to this thesis (with descriptions) can be found in Appendix A.

Chapter 2

Robot learning from demonstration: Past and current research

During the last years, the need of having robots in scenarios different from the industrial floor has increased. At the beginning, robots only worked in structured environments such as factories or research departments, thus engineers built programming routines knowing accurate information about the setting where the robot performed. However, recent robotic applications where robots may interact with humans and populate dynamic scenarios demand other types of programming approaches. Classical programming does not fulfill the new requirements that human-robot interaction (**HRI**) and changing environments demand, hand coded programs would have to consider and predict all possible human behaviors, as well as contemplate and have a response to all the possible changes in the surroundings of the robot. This is clearly not the environment where classical programming of robots is suited for.

Hence, new robot programming techniques should: *(i)* encapsulate the relevant features of the task, *(ii)* adapt to new and unseen conditions (i.e., good generalization capabilities) and *(iii)* be based on *user-friendly* systems in order to allow inexperienced users to program them in a more *natural* way (e.g., by providing gesture-based communication or natural language commands). An attractive idea is to program robots from demonstrations provided by humans and through the interaction with them. The goal is that robots can learn to execute specific tasks from this information and adapt to all changes generated in their surroundings. In this way, they will successfully carry out the given task regardless of environment changes.

As it was highlighted in [12] and [81], learning would be a promising approach for programming robots in dynamic environments. Learning can be performed in two non-mutually excluding ways, namely by direct interaction with the environment or from demonstrations carried out by a human or by another robot. The first approach is known as *reinforcement learning* where teaching derives from experience [79]. The second approach is known as *Programming by demonstration* (**PbD**), *learning from demonstration* (**LfD**), *imitation learning*, etc. Here, the robot learns from examples of the task given by a teacher. The robot generalizes through these samples generating an abstract task knowledge (at a high or low level). Through learning, robots will be able to accomplish a given task in a dynamic setting by adapting their actions based on the knowledge previously provided by a human expert and the current information acquired through their perception system.

This chapter first presents the history and concepts about LfD, where the evolution of this field is briefly reviewed. The most known learning algorithms as well as the different ways of transferring skills to a robot are explained. Applications will also be described, highlighting those where haptics and vision play a relevant role. Lastly, it will be shown how physical human-robot interaction (**pHRI**) offers new perspectives in LfD.

2.1 History and concepts

In response to the need of automating robot programming in industrial settings, LfD constitutes a suitable solution for avoiding tedious manual programming. First works were based on symbolic reasoning where a programmer demonstrated an action either manually or by teleoperating the robot, which afterwards reproduced exactly the shown task. This approach was also known as *teach-in*, *guiding* or *playback*. While demonstration was carried out, all information about the robot and its environment (e.g., positions and orientations of obstacles and targets) was stored in order to segment it into sub-goals and primitive actions to attain these sub-goals. After that, the demonstrated task was represented by means of a sequence of state-action-state transitions, which was the basis of the symbolic approaches later on. Such concepts are closely related to a branch of the artificial intelligence known as *high-level planning*. Here, given a description of the possible initial states of the world, a description of the desired goals and possible actions, the planning problem is to find a plan that is guaranteed to generate a sequence of actions that leads to one of the goal states.

Due to the presence of uncertainty in demonstrations and sensors, a method that consolidates all demonstrations was developed. Here, the state-action-state sequence was converted into *if-then* rules, where states and actions were described based on symbolic representations. Finally, the complete demonstration was represented as a graph [101, 103]. Such rules were very restrictive, thus the system showed poor generalization capabilities. This fact gave rise to the need of having more robust, compact and versatile tools for representing a task. Machine learning techniques showed up as a very promising solution.

2.1.1 Machine learning - Towards the robot learning paradigm

Machine learning is concerned about how to construct algorithms that automatically learn from experience (i.e., from data) [109]. Many such learning algorithms exist in literature, which can be classified into the following three groups with respect to the sort of feedback that the learner has access to: *supervised* techniques, *unsupervised* methods and *reinforcement learning*. In the first approach, the algorithm learns a function from labeled data, this means that a target exists for every input (e.g., artificial neural networks, decision trees, nearest neighbors method). For unsupervised techniques, an input data-set is presented but no feedback about it is given. Thus, their goal can be considered as finding a representation of particular input patterns in a way that reflects the statistical nature of the whole set [38]. The third alternative, reinforcement learning, conceives the learner receiving feedback about the appropriateness of its response [79].

The inclusion of machine learning techniques in LfD plays a key role. In this framework, the training data consist of sensory and motor information (i.e., perceptions and actions) acquired from the perception and proprioceptive systems of the robot, respectively. This data stream is processed by a machine learning algorithm that generates actions as a function of sensory states, allowing a generalization on the samples in a more natural way [111]. A much better encoding of the task is achieved if the relevant

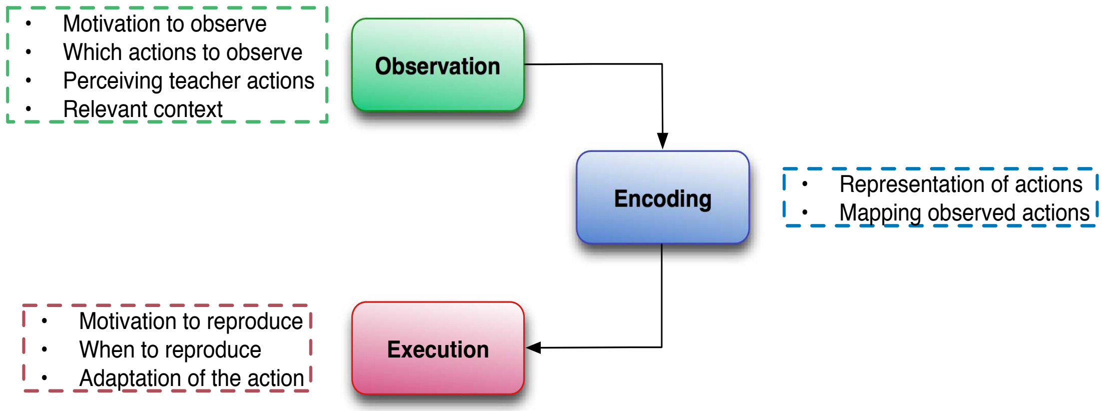


Figure 2.1: The three main phases in imitation learning according to [8].

actions and perceptions are recognized in advance, so as to remove actions that do not contribute to the learning process and to provide smoother action transitions [41, 80].

2.1.2 Imitation learning

The machine learning role in LfD and the analysis of specific neural mechanisms for visual-motor imitation in primates, as well as the evidence of developmental stages of imitation capacities in children led to name LfD as *imitation learning* or *learning by imitation*. This term was analyzed in depth by Bakker and Kuniyoshi [8], who tried to make a definition of what imitation is and to analyze if this is what robot imitation should be.

From a psychological point of view, Thorndike defines imitation as: *From an act witnessed learn to do an act* [155]. Based on this, Bakker and Kuniyoshi postulated that *imitation takes place when an agent learns a behavior from observing the execution of that behavior by a teacher*. This was the starting point for establishing the features of robot imitation: *(i)* adaptation, *(ii)* efficient communication between teacher and learner, *(iii)* compatibility with other learning algorithms and *(iv)* efficient learning in a society of agents. In addition, three processes were identified in robot imitation: sensing, understanding and doing, which can be redefined as: *observe an action, represent the action and reproduce the action*. These three main issues entail all current challenges in robot imitation, see Figure 2.1.

On the other hand, Schaal [140] introduced an inductive approach that deals with the problem of how information acquired from demonstration can be translated into an action, where he presents the concept of *movement primitives*: sequences of actions that accomplish a complete goal-directed behavior and allow to have a compact state-action representation. According to this, Schaal – based on the Piaget's theory [125] – describes imitation learning as a process where "a perceived action of the teacher is mapped onto a set of existing primitives in an assimilation phase. Then, the most appropriate primitives are adjusted by learning to improve the performance in an accommodation stage. If there is not a good match for any of the primitives in front the observed behavior, a new primitive must be created".

Another work defined imitation learning from a more biological perspective, where action and perception work in a joint way [105]. The proposed definition considered a *behavior-based control* where the most important aspects to solve are how to *interpret* and *understand* observed behaviors and how to integrate the perception and motion control systems to reconstruct what was observed. There are two relevant challenges here: *(i)* to recognize human behavior from visual input, *(ii)* to find methods for structuring the motor control system for general movement and imitation learning capabilities. These challenges may be solved by using a behaviors-based control system, where behaviors are real-time processes that take inputs from sensors or other behaviors and send output commands to effectors or other system behaviors, as Mataric proposes [105]. This approach may be implemented along with *Neuroscience* ideas to structure humanoid motor control, where *spinal fields* [14] and *mirror neurons* [39, 132] concepts are combined for defining a learning structure based on *perceptual-motor primitives*.

The topic of imitation learning has also been addressed by Breazeal and Scassellati in [17], who made clear differences between learning by imitation, learning to imitate, learning by demonstration, task-level imitation and true imitation. They also reviewed possible solutions for two of the main problems in imitation, namely: *What to imitate* (i.e., determining which aspects of the demonstration should be imitated), and *how to imitate* (i.e., determining how the robot would perform those parts of the demonstration that should be imitated). These challenging issues have been the main concerns of research in LfD during the last decades, but most of the efforts have been devoted to develop learning frameworks able to represent the demonstrated tasks successfully.

2.1.3 Control policies

The need of having an algorithm representing skills that will be transferred through imitation, as well as the way they are generated in a generic manner are issues of current research in LfD. Skills can be represented either at a low level where a non-linear mapping between sensory and motor information takes place, or at a high level where the skill is represented as a sequence of action-perception units [12]. An interesting work about finding a generic structure for LfD is explained by Schaal *et al.* in [143], where a computational formalization of imitation learning concepts based on motor control is put forward (assuming that the perception problem is already solved). They postulate that the motor control problem can be conceived as finding a *task-specific control policy* that maps relevant states (which can or cannot be functions of time) into motor commands.

Hence, imitation learning may be defined as the problem of how *control policies* can be learned by observing a demonstration. For instance, an approach known as *imitation by direct policy learning* tackles the problem by directly modeling the control policy through supervised learning of its parameters, which belong to the open variables of the mathematical method to be used (e.g., the weights of a neural network) [37, 143]. Another approach called *imitation by learning policies from demonstrated trajectories*, uses the demonstrated human movements as seeds for an initial policy which can be optimized by a self-improvement process or active teaching [110]. A third proposal is named *imitation by model-based policy learning*, here not a policy but a predictive model of the task dynamics is approximated from the demonstrated behavior [6]. Given knowledge of the task goal, the task-level policy of the demonstrated movement primitive can be then improved with reinforcement learning procedures based on the learned model.

2.2 Skills transfer

One of the key aspects in LfD is to provide a user-friendly method to teach the task to the robot, which greatly depends on hardware issues. Initially, most robots were controlled by very restricting admittance-based control laws and commanded by standard electric motors that constrained how humans could transfer a specific skill to a robotic learner. In the way towards enhanced user-friendliness, teleoperation and camera systems appeared as suitable tools for supporting the teaching process. Teleoperated robots constitute an adequate solution for teaching complex tasks in hostile, unsafe or inaccessible environments [124]. In this type of scenarios, a robot located in a remote place is teleoperated through a local interface driven by a human operator. These local interfaces usually have a display providing the user with visual feedback about the remote scene [158]. In addition, they may offer force/torque feedback through a haptic device, which largely enriches the information sent back to the teacher as well as provides a bidirectional communication channel between the partners [106]. One of the main drawbacks of LfD by teleoperation is time delays, which may make the teaching process slow and exhausting if these are significantly high.

As for camera systems, vision is one of most used systems for capturing human demonstrations because it is considered as the most natural way of *observing* actions in humans. In this context, the teacher carries out the samples of the skill to be learned while the robot records them using a set of cameras. However, these vision-based systems have to be very simplified to avoid all the well-known problems in computer vision (e.g., segmentation, occlusions, lighting), for instance by placing a set of markers on the teacher's body [166], and on objects to be manipulated during the task[1] [10]. After this, the next problem to be solved is the kinematic mapping to represent the demonstrations either in the joint or the operational space of the robot, which is a key problem when teaching humanoids [158].

As mentioned, vision systems extracted information from human motion, thus most information collected by the camera was discarded (e.g., color, contours, shadows). Because of this, another much simpler hardware has been designed to exclusively collect motion data, as optical – or magnetic – tracking systems do. The main advantage of these devices is that they provide very precise data about the position – and possibly orientation – of bodies defined by a set of markers, avoiding the typical image processing problems of cameras-based systems. However, the drawback of kinematic mapping (i.e., the correspondence problem) remains when humanoid robots are in the loop [90]. This problem refers to how the imitator knows what pattern of motor activation will make

[1]Currently, part of the research on computer vision is focused on finding invariant features on the objects of a given scene, which may be considered as markers that do not vary with rotations, scale changes, etc.

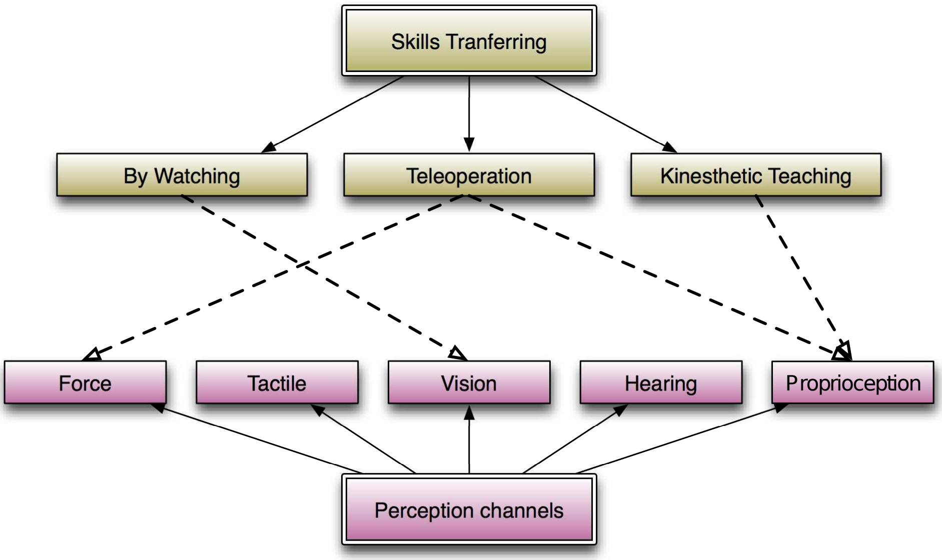


Figure 2.2: The most common skill transferring methods and perception channels used in imitation learning.

its action looks like that of the demonstrator [16], which is not straightforward if the samples of the task are not provided using the robot's own embodiment (e.g., through kinesthetic teaching). Furthermore, placing markers on the user body is not natural, and thus hardly used in domestic environments.

It is worth highlighting that *a strong relationship exists between the skill transferring method used by the teacher and the input perceptions sensed by the robotic learner*. In other words, the way in which the teacher transfers his/her knowledge about the task indirectly conditions at least one of the robot
ion channels to *observe* the human demonstrations (Figure 2.2). The above described systems are mainly used to transfer movements, this means that the learning process occurs at trajectory-level, where the robot must encode time-series data as streams of joint angles [98], or Cartesian positions of the end-effector [83].

With the formerly described robots hardware and control approaches, the learning process was not considered user-friendly, safe and natural as desired. However, insights in new control schemes and novel actuators (e.g., backdrivable motors) allowed to create much safer and more versatile robots, which can be manipulated and held by humans effortlessly [3, 133]. Also, small and light robotic arms or humanoids can be easily manipulated by setting the motors in a passive mode. This opens the door to a new skill transferring approach, namely: *kinesthetic teaching*. Such term refers in general to the procedure where the teacher is holding the robot, which is gravity compensated at its wrist and/or links, and moves the robotic arm along the trajectories that the robot has to follow during the execution of the task. This method has been greatly exploited to demonstrate trajectory-level skills [2, 24], where better and more precise samples can be provided as the teacher is much more involved in the learning process of the robot by experiencing the kinematic constraints at first hand.

2.3 Perception systems

As mentioned above, vision-based systems (e.g., cameras, optical tracking systems) have been the most used hardware to *observe* the teacher demonstrations and to perceive the robot environment.[1] Nevertheless, the robotic learner may be endowed with other type of sensors depending on the task to carry out [77]. For instance, proprioceptive sensors provide information about the internal state of the robot (e.g., motor encoders), which are very useful during kinesthetic [28], or teleoperated teaching [124]. Auditive perception has been also used in LfD settings, where a human enhances the examples of the task by telling words or sentences that provide more information about the current state of the demonstration [122]. Such systems have been extensively applied in social robotics, because this kind of HRI shares similarities with how humans interact daily.

When the task to learn involves contact with objects or surroundings (e.g., manipulation tasks), force-based perception systems are used to extract information such as reference force/torque profiles [148], or tactile data [35]. The importance of this kind of sensing increases as robots become safer, which allows them to physically interact with humans, where haptic communication takes place. Haptic cues have been shown to be a very rich source of information in human-robot collaborative scenarios, since they can convey intentions over the course of the task [150]. In this context, these force measurements are also used to learn and establish the roles of the partners [149].

[1] The robot may use a different set of perception systems during the demonstration phase and the reproduction of the task. For instance, the robot may observe demonstrations of trajectories to grasp a cup through an optical tracking system and a set of cameras, while in the execution of the task the robot would only need the cameras to know where the cup is located.

2.4 Representation of knowledge

Once the demonstration phase has finished, the data collected by the robot must be encoded through a model that represents the taught task in a compact way. Such representation depends on what the robot needs to learn from the human samples and on the complexity of the task. On the one hand, when a robot is desired to carry out a specific set of movements (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 represented at *low-level*. Here, the robot actions are directly determined by such variables, which govern motor commands to be sent to the robot controller. Hence an approximation of the perception-movement *mapping function* must be found [5]. 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. Continuous encoding models and regression-based techniques are fit for representing this kind of tasks.

On the other hand, very complex tasks are commonly split up in a set of sub-goals to be achieved by the robot. In such a case, the task representation is at *high-level*, where the sub-goals are represented as state-action pairs. Mostly, the learning process involves to discover 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 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. Discrete encoding algorithms and graph-based models are used to represent this set of tasks.

2.4.1 Encoding of skills

In Machine Learning literature [13, 109], it is possible to find a huge quantity of algorithms that can be adapted to LfD applications. The selection process of the model to be used should consider the type of tasks to encode and the level of representation of the teacher demonstrations. Learning techniques based on dynamical systems models and stochastic approaches stand out over more ad-hoc algorithms, because they provide a more general structure to encode different type of skills using the same basis framework.

2.4.1.1 Dynamical systems models

Regarding learning methods based on dynamical systems, Ijspeert *et al.* proposed to use nonlinear differential equations to form control policies in trajectory formation [75]. In this approach – known as dynamic movement primitives (**DMP**) – the dynamical system represents a whole 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 field can be constructed from demonstrations, which can automatically correct external perturbations and guarantee convergence to a goal state. DMP have been extensively used in imitation of reaching movements [74], for encoding rest-to-rest motions in articulated mobile robots [120], and also extended to manipulation tasks where the behavior of the primitives is influenced by perceptual cues [83].

The main idea behind DMP is to use simple dynamical systems (e.g., a set of first order systems) and transform them into a nonlinear system with determined attractor dynamics by means of a learnable autonomous forcing term. Such an approach has been the basis for further reformulations and modifications of DMP. For instance, Hoffmann *et al.* 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 [68]. On the other hand, Pastor *et al.* 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.* [83]. In this work the original formulation was modified by including a coupling with external variables, most of them considered as perceptual cues. Finally, a recent work presents a novel approach for joining several DMPs by overlapping kernels, in order to reproduce very complex trajectories composed of simpler movements that need to be mixed through smooth and natural transitions [92].

2.4.1.2 Stochastic approaches

Gaussian Mixture Models (**GMM**) are the basis of several LfD frameworks performing successfully in a variety of scenarios. 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.* used this mixture modeling to teach simple manipulation tasks to a humanoid robot [26]. 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 stage over the training datapoints is needed to obtain such type of data. Among the solutions, one can find Dynamic Time Warping (**DTW**) [25, 44] and Hidden Markov Models (**HMM**) [88, 127].

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. In LfD, HMMs have been used for teaching collaborative lifting tasks to a humanoid robot [50], for learning and reproduction of a bi-manual task and tennis table strokes [27], and as a basis for a hierarchical incremental learning of full body motion [91]. Recently, some extensions of the classical HMM formulation have been also applied to learn tasks from imitation. Krüger *et al.* [86] 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 [29], 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.

2.4.2 Reproduction of skills

Once the task model has been learned, it is necessary to reproduce the learned skill. 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 nonlinear equations used for encoding the demonstrations. In contrast, when encoding through probabilistic models, retrieving smooth trajectories is not that straightforward. First efforts addressed this problem using an *averaging* approach applied to an HMM, where generalized movements are retrieved by averaging over a large number of trajectories previously generated from the trained model [98]. Such an approach is very time consuming and computationally expensive, does not guarantee smoothness of the results, and it may also smooth out important peaks in the human motion.

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 keypoints for interpolating them [22]. The main drawback of this approach is the fact that the covariance information is ignored. Then, Calinon *et al.* introduced the use of Gaussian Mixture Regression (**GMR**) [55] to retrieve a time-based trajectory from a set of demonstrations encoded by a GMM. GMR 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 [21]. Recently, a quantum theory based GMR was proposed by Chatzis *et al.* [32], which allows for a significant performance increase in comparison with other state-of-the-art LfD methodologies.

In contrast to the above approaches, other regression-based methods have been also used to encode and reproduce robotic skills, mostly based on trajectory following. A popular technique is locally weighted regression (**LWR**) [7], a memory-based algorithm that combines the simplicity of linear least squares along with a weighting mechanism to learn nonlinear functions. Such an approach was the core for two further extensions, namely receptive field weighted regression (**RFWR**) [141], and locally weighted projection regression (**LWPR**) [159]. The first one dealt with the problem of moving from a batch process to an incremental learning strategy, but suffering from the curse of dimensionality. This drawback was overcome by LWPR, which was shown to operate efficiently in high dimensional spaces. On the other hand, Gaussian processes (**GP**) have been also applied to LfD tasks. Grimes *et al.* proposed to use GP for nonparametric forward model learning in whole-body motions of a humanoid robot [61]. Nonetheless, its main disadvantage is the high computational cost, preventing it to be used in very complex tasks or in incremental learning. 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 [119]. Grollman and Jenkins [62] compared this approach with LWPR in the context of LfD, where both techniques provided good function approximation capabilities. However, regarding hard memory and timing guarantees issues, sparse GP showed to be more suitable for real-time interaction. In contrast, Nguyen-Tuong *et al.* [116] 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. In the same vein, Schneider and Ertel 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 [144].

Researchers in the field of LfD agree that a widely adopted assumption to represent complex skills and nonlinear movements 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 [27], and methods whose core is the DMP [82]. These techniques differ in the way the linear systems are estimated and constrained, and in the way the activation weights are defined to combine the linear systems. This representation allows to develop more generic frameworks able to deal with complex tasks, as well as to extend the learning process to teach other type of human behaviors to robots, such as those based on impedance.

2.5 Applications - Experimental Scenarios

Works dealing with LfD have been carried out on different settings, where both sensory information and how the demonstrator provides the robot with samples differ for each application. Most of the efforts have focused on teaching a given path or trajectory to be followed by the robot. Such trajectory may correspond to a set of desired position/velocity profiles of the robot's end-effector as well as of its joints. This low-level learning has been successfully applied to manipulation tasks, grasping skills, gesture reproduction and whole body motion pattern imitation. However, the development of compliant robots brings new possibilities in imitation learning, by extending the skill transfer problem towards tasks involving force, and towards reactive systems able to cope with various sources of perturbation coming from the interaction with the user and the environment.

2.5.1 Manipulation tasks

A large group of researchers have focused on endowing robot 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 an appropriate test-bed to study LfD issues. Initially, such type of tasks were demonstrated to the robot by capturing the teacher samples exclusively using vision-based systems [94]. Dillman *et al.* used fuzzy sets and information theory to transfer manipulation skills to a robotic arm, where the demonstrations were recorded using a stereo vision system [41]. More sophisticated systems have been recently used to record manipulation human skills, such as magnetic or optical tracking systems. Shon *et al.* [147] used reflective markers attached to the teacher's body in learning a lifting task, where HMMs were applied to encode and recognize the demonstrations representing a forward model of the skill at hand. A similar perception system was used by Krüger *et al.* [86] to track human examples of *pick and place* movements, where abstract relationships between objects and robot actions were determined from a low-level representation based on a parametric version of HMMs.

In contrast to the aforementioned works, other researchers studied similar tasks but these were demonstrated using other skills transferring methods. For instance, assembly tasks were also analyzed in teleoperation settings, where force sensory patterns conditioned the robot actions, which can be modeled as a sequence of contact formations and desired transitions between them [148]. Telerobotics has been also exploited to teach complex manipulation tasks (e.g., drill-mating or chisel-pickup) to a humanoid robot for space applications [77, 124]. In recent times, this type of skills transfer has benefit from force-reflecting devices, which offers two significant advantages: *(i)* the teacher gets knowledge about how the robot is perceiving its environment based on force feedback, and *(ii)* learning is extended to the force domain. In this field, Howard and Park [70] took advantage of a haptic device to regulate the teacher behavior in demonstrating a manipulation task using guidance forces derived from visual input data. In this work, a discriminative model based on neural networks was used to learn the control sequences necessary for task execution.

Lifting tasks have been also demonstrated using haptic devices. Evrard *et al.* proposed a framework to teach to a humanoid robot how to lift a beam cooperatively with a human operator [50]. The authors used a GMM to encode a pure follower/leader role distribution while GMR was implemented to reproduce the manipulation skill. Force-based perceptions are considered as inputs for learning the task, assuming that the proposed framework is able to encode the dynamics of the motion, and the synchronization and adaptation processes. This extension of LfD to manage force-based demonstrations is crucial in manipulation tasks as noted by Kormushev *et al.* [84]. Their work dealt with the problem of teaching force skills demonstrated through kinesthetic teaching. Different force profiles of contact-based tasks (ironing and door opening) were demonstrated through a haptic device while the robot followed a previously learned trajectory. Both space and force constraints of the skills were represented as a mixture of dynamical systems based on virtual damped springs.

Force-reflecting devices have been also used to teach tasks in virtual environments where the haptic signals sensed by the human demonstrator correspond to forces computed from mathematical models instead of coming from sensory readings. In [43], a virtual *peg-in-hole* task was learned by encoding the human demonstrations through an HMM. After learning, a physical robot reproduced the task using LWR as an approximator for the trajectories encoded at each state of the model. Mayer *et al.* [106] also used virtual environments along with haptic devices, in this case to teach *knot-tying* tasks in a minimally invasive surgery context. Their skill transfer framework comprises a LfD module supported by a *scaffolding*[1] process, which assists the robot to extract sensory-motor primitives from the human demonstrations.

[1]Scaffolding refers to an assistance, which is generated from the superior knowledge of the teacher.

2.5.2 Human motion

As learning at low-level has for a long time been on the top of the agenda for LfD, many works have used such representation in human motion imitation. Calinon and Billard devoted their research to gesture recognition and reproduction, where HMMs [22] and incremental versions of GMMs [24] were tested to encode the teacher demonstrations, while reproduction was implemented using GMR. They also focused on the idea of how to find a representation of the data that encapsulates only the key aspects of the human action, and discards the intrinsic variability across people movements (i.e., the *what to imitate?* problem). This was tackled by applying principal components analysis (**PCA**) to reduce the redundancy in the human samples while reducing the dimensionality of the data. Akgun *et al.* [2] also addressed the problem of gesture imitation using GMMs, but their contribution lies on the way the demonstrations are provided. They proposed to provide a sparse set of consecutive *keyframes* that achieve the skill when connected together, unlike kinesthetic teaching where continuous uninterrupted samples are given.

Full body motion imitation has been another field of application. Ude *et al.* [158] dealt with the problem of programming the movements of a humanoid robot from data generated by human motion, which was sensed by an optical tracking system. They proposed a new approach to the formulation and optimization of joint trajectories for humanoids using *B-spline* wavelets. Kulić *et al.* presented an entire LfD framework to incrementally learn whole body human movements using factorial HMMs [56]. The robot encoded the different motion patterns using a separate model for each of them, which can also be used for recognition purposes. The models are automatically organized in a hierarchical tree where a clustering process takes place once a specific motion has been recognized, so that similar movements can be grouped and synthesized by a single model [88]. Kulić and her colleagues improved this work in recent times by reducing the complexity of the models to be classical HMMs and by learning a temporal relationship between motion primitives via the construction of a motion primitive graph [91].

Human motion is also essential for social robots, which are expected to interact with humans. Takano *et al.* modeled primitive nonverbal communication based on gestures through a hierarchical mimesis model represented by three groups of HMMs [153]. Such model integrated imitative learning with communication in a compact framework, extending LfD to social robotics applications. In contrast to this approach, communication between a human and a robot can also show up through physical interaction. Lee and Ott [99] presented a LfD structure to teach human gestures to a robot through observational learning and subsequent kinesthetic corrections. The key feature here is the definition of a motion refinement tube, which regulates the stiffness of the robot joints according to the variance encapsulated by an HMM. A customized impedance controller based on such tube allows deviations from a nominal trajectory, so that the robot can incrementally improve its representation of the task using the kinesthetic modifications performed by a human.

2.6 Physical human-robot interaction in LfD

Physical HRI (**pHRI**) opens the door to new possibilities and scenarios where a robot can help and collaborate in an active way to perform a task with a human. Collaborative tasks are a new branch of research in HRI, where haptic communication is an important component to determine the roles of each partner in the dyad (e.g., leader and follower) and to accomplish the goal successfully [130, 131]. pHRI offers two new important perspectives in LfD: *(i)* it permits to demonstrate skills by guiding kinesthetically the robot through the task, and *(ii)* it allows to transfer skills that are not only represented by position information, but also by force information.

2.6.1 Human-robot interaction based on haptic inputs

In the physical human-human interaction (**HHI**) context, haptic inputs have shown to be a rich and complex communication channel between the partners. Reed *et al.* [131] proposed to analyse the performance on HHI through a set of experiments showing that a dyad performs better and faster in collaboration than if the given task is carried out individually, which highlights the possible advantages of using a robot as a collaborative partner. In this scenario, haptic inputs are a very valuable information source when HRI includes physical contact between the robot and the human either directly or through an object [130]. For instance, these signals allow the robot to recognize human intentions in order to change its behavior accordingly, as well as to determine the role of the participants in the task [149].

In the field of intention recognition, Stefanov *et al.* [150] proposed to identify human intentions in a computer-assisted teleoperation setting by analysing the haptic interaction signals which are discretized using a threshold-based technique and then encoded through two HMMs representing the stages of the task, namely, transportation and positioning phases. A similar approach is introduced in [162] for a human-robot handshaking task, where an HMM-based high-level controller estimates human intentions and modifies the reference trajectories accordingly. The main idea is to imitate the compliance behavior seen in human-human handshaking by analyzing the haptic inputs and the human intentions. All these works addressed the problem of identifying the role for both partners during the execution phase, extending former approaches where each role was predefined [129, 161].

2.6.2 Impedance-based robot behaviors

During the last years, an increasing effort has been devoted to exploit the advantages provided by the impedance-based control of robots, working on the foundations given by Hogan in his seminal work [69]. In broad words, Hogan highlights the importance of considering robot control as a hybrid structure where position and force control must be developed in parallel. He proposed an energy-based model, which considers the dynamic interaction as a flow exchange where an element behaves as an admittance (i.e., the environment) and the other one as an impedance (i.e., the robot).[1] Considering that the robot behaves as an impedance has opened new research branches to control and build safer and friendlier robots to interact with human beings.

Impedance in humans has also been studied with the aim of gaining in-depth knowledge of the roles of the muscles, tendons, brain and spinal cord in modulating impedance when we interact with the environment. For instance, [18, 58] have investigated how humans modify their impedance – by activating specific muscles – to stabilize an unstable task or in trajectory tracking. New efforts have been devoted to mimic or copy these insights on how human impedance works with robots. In [53], the authors analyze how the impedance on the human arm is regulated through the muscles and propose to implement this behavior in a robot by means of feedforward and feedback commands.

[1]From a physical systems perspective, admittances accept effort inputs such as forces and yield motion, while impedances accept motion inputs and yield force outputs.

In this way, the robot is able to adapt its force and impedance while minimizing its trajectory tracking error. On the other hand, studies on transferring impedance control strategies to robots with variable impedance actuators have been developed [71]. In this work an apprenticeship learning approach is proposed to record the optimal behavior of the system from human movements. This approach is used along with optimal control techniques to transfer the behavior characteristics to the robot actuator.

In the field of HRI, Evrard and Kheddar [49] described an homotopy-theory-based model to modify the robot control signals in a collaborative lifting task with humans. In this work, a smooth switching function for the robot is determined to balance between two extreme roles: leader and follower. This is achieved by modifying the trajectory and impedance parameters of the robot on the basis of force-based perceptions. A similar task was tackled in [27, 60], where the robot learns the task using GMM to anticipate human intentions and to modify its behavior according to the force data. In [60], the learning stage is carried out by implementing a GMM-based encoding of the task, while motion adaptation is achieved by tuning the impedance parameters of the robot as a function of the errors in the trajectory, the velocity, and the force data.

Transferring correct compliant behaviors is an uprising challenge in LfD. Such behaviors can be controlled by the estimation of the robot compliance level which has for example been computed from robot position variability [28] or through human-stiffness estimation using electromyography signals [1]. Recently, impedance shaping has been exploited to refine robot motion primitives by physical interaction with a human teacher [99].

2.7 Chapter highlights

Looking at the previous sections, it is interesting to notice that research in the LfD field has missed two particular – and relevant – topics. On the one hand, most of the work on LfD has focused on how to represent and reproduce a given task, and only a few researchers have worked on finding possible solutions to the *what to imitate?* problem. This is a crucial part of any learning framework, because the proposed solution may make the skill learning considerably easier and faster. Such an issue is tackled in the next chapter, where a feature selection algorithm is proposed as a possible solution to find the relevant perceptions of a task.

On the other hand, learning has been analyzed and 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 perception systems may miss relevant data in other type of scenarios, like tasks implying physical contacts. Thus, learning in the force domain is a highly promising route of research to build smarter – and probably safer – robots, which is the core of this thesis.

By finding possible solutions to the aforementioned issues it would be possible to build a whole learning framework able to encode and reproduce a large set of force-based tasks, and that would also analyze the perception signals in order to extract the relevant data to learn a skill. Such a framework may be considered a step further to achieve a unified structure that endows robots with learning capabilities, which is still a challenging and open problem in the Robotics community. Also, it will be important to take into account that impedance controllers are providing robots with new control features, allowing them to learn and reproduce new kinds of tasks. Therefore, the proposed learning framework should also be able to work in these new contexts. These issues will be treated in the final chapters of this thesis.

Chapter 3

Selecting relevant perceptions

During the demonstration phase, the robot *observes* the human performing a task using its perception system to gather all the information about the current state of the environment, the teacher actions and their effects on the surroundings. Part of the collected information may be useless for the skill at hand, and some variables may provide redundant data. These conditions give rise to one of the main open questions in LfD, namely *what to imitate?* This question refers to determining which information of the demonstrations is relevant for learning the task successfully, and it takes place at the first stage of the teaching process [12, 115]. Here, this issue is addressed by analyzing the importance of every perception channel (from now on, perceptions for short) during the whole reproduction of the task.

So far, most works tackle this problem by analyzing the variability across demonstrations of the task at trajectory level. Those parts with large variances do not have to be learned precisely, whereas low variance suggests that the corresponding motion segment is significant and deserves to be encoded [26, 84]. This approach exploits variance for constructing task constraints [25] as well as for determining secure interaction zones in a robot coaching framework [99].

However, these methods are not able to uncover the relative relevance of each individual input dimension for the task to be learned. Irrelevant or redundant information may actually be present across input dimensions, which increase the computational cost of the learning stage and make the task harder to learn (e.g., in some LWR-based methods). The point here is to select the subset of the most relevant input variables. The benefits in computational cost and noise reduction during the learning stage may outperform a hypothetical and marginal loss of information. Furthermore, this approach may be compatible with the previously described variance-based analysis criterion, as this method can be applied to the remaining variables.

This chapter shows how feature selection techniques can be used in this context. Specifically, the mutual information (**MI**) criterion can be exploited to determine which perceptions are the most relevant, by analyzing their influence on the robot actions. This method is compared against other state-of-the-art algorithms used to solve the same problem in previous works. Previously, however, specific issues concerning the processing of haptic signals provided by force sensors are also addressed here, because they are of particular interest when the robot carries out manipulation skills holding tools at its wrist.

3.1 Perception processing

When a robotic arm carries out manipulation tasks, its end-effector is often in contact with the environment directly or through an object (the workpiece or a given device). Such interaction with its surroundings produces forces and torques on the robot's tool, which have to be measured in order to gather essential information about the task. The collected data should contain only patterns related to the skill at hand, but unfortunately this never happens in real scenarios. The weight and inertial forces of tools and manipulated objects, intrinsic sensor features and the dynamics of the task introduce noise in the data streams. Thus, it is necessary to remove any undesired signal so that the robot perceptions exclusively represent the needed information.

On the one hand, high-frequency noises such as vibrations of flexible loads – induced by the robot motion – or the intrisic noise of the sensor can be significantly reduced by applying classical low-pass filters [45, 134]. On the other hand, the magnitude of inertial disturbances cannot be ignored when large accelerations and fast motions are considered (i.e., in highly dynamic tasks) or when the robot holds heavy tools. This problem is tackled here using a simple dynamic model that allows to estimate the external forces/torques by taking into consideration the inertial components in the sensor readings. Considering the system shown in Figure 3.1, let $\boldsymbol{p}$ denote the

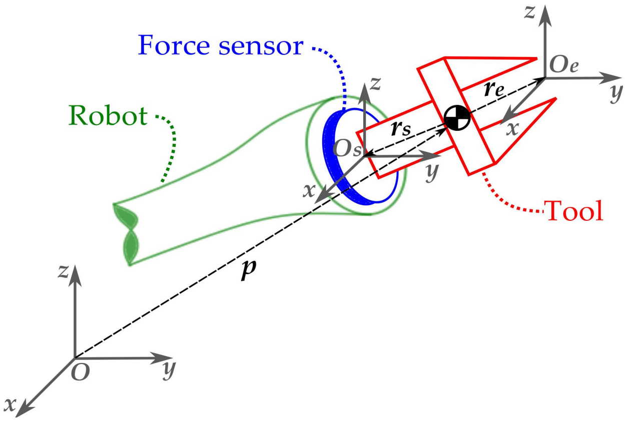


Figure 3.1: Dynamic modeling of the force sensing process.

position of the center of gravity of the robot's tool, $\boldsymbol{\omega}$ its angular velocity, m its mass, $\boldsymbol{I}$ its moment of inertia, $\boldsymbol{g}$ the gravitational force, $\{\boldsymbol{F}_s, \boldsymbol{T}_s\}$ and $\{\boldsymbol{F}_e, \boldsymbol{T}_e\}$ the sensor and external forces/torques respectively, and $\boldsymbol{r}_s$ and $\boldsymbol{r}_e$ the vectors from the center of gravity of the tool to the sensor and external forces frames. Then, using the Newton-Euler equations,

$$\sum \boldsymbol{F} = m\ddot{\boldsymbol{p}} = m\boldsymbol{g} + \boldsymbol{F}_e + \boldsymbol{F}_s, \tag{3.1}$$

$$\sum \boldsymbol{T} = \boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} = \boldsymbol{T}_s + \boldsymbol{r}_s \times \boldsymbol{F}_s + \boldsymbol{T}_e + \boldsymbol{r}_e \times \boldsymbol{F}_e. \tag{3.2}$$

It is worth mentioning that this dynamic modeling of the force sensing process has been used as the basis of more complex methods for estimating external forces/torques in high-speed robot manipulation [157], robot compliant control [102] and cooperative robots sharing a load [93]. On the other hand, the same model is greatly simplified when the task characterizes by low velocities and accelerations,

$$\boldsymbol{F}_s = -m\boldsymbol{g} - \boldsymbol{F}_e, \tag{3.3}$$

$$\boldsymbol{T}_s + \boldsymbol{r}_s \times \boldsymbol{F}_s = -\boldsymbol{T}_e - \boldsymbol{r}_e \times \boldsymbol{F}_e. \tag{3.4}$$

This model will be applied to three different experimental setups (see Section 1.4) in order to work with *clean* haptic data. These will be used as input information in the training dataset for learning processes, and also for providing the teacher with force feedback through haptic interfaces when bidirectional communication channels are implemented (Sections 1.4.1 and 1.4.2).

3.2 Selection through feature transformation

Feature transform is a well-known approach in variable selection and dimensionality reduction. It basically consists on constructing a reduced number of new variables out of the original ones. Several techniques have been proposed to achieve such transformation. One classical linear transform for dimensionality reduction is principal component analysis (**PCA**). This transform is derived from eigenvectors corresponding to the largest eigenvalues of the covariance matrix of data. The method seeks to optimally represent the data in terms of minimal mean-square-error (**MSE**) between the representation and the original data [78]. Another similar algorithm that makes use of second-order statistical information, the covariances, is the linear discriminant analysis (**LDA**). 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 find a subspace on which the data has the least Gaussian projection [73]. This criterion corresponds to finding a projection of data that looks maximally clustered.

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 [26]. This allowed to reduce 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. Such approach may be considered as a possible solution to the *what to imitate?* problem, and it will be taken into consideration in this chapter for comparison purposes against the proposed solution described in Section 3.3.

Formally speaking, let $\boldsymbol{X}$ be a matrix containing the entire training input data with variables $\{x_1, x_2, \ldots, x_{\mathcal{I}}\}$, where $\mathcal{I}$ is the dimensionality of the input dataspace. New variables $\boldsymbol{\zeta} = \{\zeta_1, \zeta_2, \ldots, \zeta_{\mathcal{D}}\}$ are computed from an orthogonal linear transformation of the original data, which is defined by the matrix $\boldsymbol{A} = \{v_1, v_2, \ldots, v_{\mathcal{D}}\}$, with v_i being the eigenvectors of the covariance matrix of $\boldsymbol{X}$ with associated eigenvalues λ_i. $\mathcal{D}$ is the minimal number of eigenvectors used to obtain a satisfying representation of the data, i.e., such that the projection of the data onto the latent space defined by $\boldsymbol{A}$ covers at least 98% of the data's spread, that is $\sum_{i=1}^{\mathcal{D}} \lambda_i > 0.98$. This technique will be applied to the *ball-in-box* task and the results will be compared against those obtained from a *feature selection* approach in Section 3.4.1, where also a discussion about its advantages and drawbacks in the LfD context will be given.

3.3 Mutual information-based selection

Feature selection methods keep only useful variables and discards others, often in the original dataspace. This approach is needed 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 used more easily than if they were projected on a different space. In this context, Mutual Information analysis is the approach proposed in this thesis as an alternative solution for choosing the relevant perceptions to learn a task.

3.3.1 Classical approach

Here the MI criterion is used, which allows to establish which input variables give more information with respect to their effects on the outputs (i.e., how perceptions affect actions). In contrast to other techniques (e.g., correlation criterion), MI detects non-linear dependencies between inputs and outputs and accounts for higher-order statistics, not only for second-order ones [63, 156]. The purpose of this method in feature selection is the reduction of the output data uncertainty, provided by each input variable [9]. In our context, depending on how the uncertainty of the output data is reduced, a robot perception gives more or less information about the desired actions. It is worth to highlight that this approach has shown satisfactory results in sensor fusion [76], and vision-based positioning of a robotic arm [163].

Formally, the MI value between two continuous variables $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as follows (more details in [146])

$$I(\boldsymbol{x};\boldsymbol{y}) = \int_x \int_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. \tag{3.5}$$

Here, both the marginal and joint probabilities may be approximated using histogram-based densities, which are computed from discrete partitions of the dataspace. The quantization error in the conversion from continuous variables to discrete ones is bounded by some constant value which depends only on the number of partitions that divide the continuous space [96]. It should be noted that other types of non-parametric density may also be used, such as Parzen windows [97].

Specifically, given a training dataset containing $\mathcal{I}$ inputs and $\mathcal{O}$ outputs, the MI value is computed for every input-output pair $(\boldsymbol{x}_i, \boldsymbol{y}_j)$, with $i = 1, \ldots, \mathcal{I}$ and $j = 1, \ldots, \mathcal{O}$. These values allow to sort the inputs based on the criterion of maximal information with respect to the output $\boldsymbol{y}_j$, this means that the first input in the ranking will be that with the largest $I(\boldsymbol{x}_i; \boldsymbol{y}_j)$, reflecting the largest dependency on the given output. Then, the problem here is to select a subset $\boldsymbol{\Omega}$ of $\mathcal{K}$ perceptions from the original set $\boldsymbol{X}$ of $\mathcal{I}$ inputs, that is "maximally informative" about the entire set of robot actions $\boldsymbol{Y}$. The simplest approach is to carry out a sequential search, where the best $\mathcal{K}$ individual input variables, i.e., the top $\mathcal{K}$ features in the descent ordering of $I(\boldsymbol{x}_i; \boldsymbol{Y})$, are often selected to create the subset $\boldsymbol{\Omega}$ for $\boldsymbol{Y}$. Nonetheless, it has been recognized that the combinations of individually good features do not necessarily lead to good classification performance [123].

3.3.2 Conditional Mutual Information

The maximal information criterion should be used for selecting only *the most relevant* input. However, to choose the remaining $\mathcal{K} - 1$ perceptions, the redundancy among inputs may be taken into consideration (i.e., a minimal redundancy criterion). To achieve this, we resort to a modified *greedy selection* algorithm known as "mutual information-based feature selection deduced from uniform distributions" (**MIFS-U**) [96], which was adapted here to fit LfD tasks characteristics as described in Algorithm 1. The core of this technique is to select the rest of variables by maximizing the conditional MI $I(\boldsymbol{x}_i; \boldsymbol{Y}|\boldsymbol{\Omega})$, this means to choose the input $\boldsymbol{x}_i$ that provides most information about

Algorithm 1 MIFS-U

1: Initialization: Set $\boldsymbol{\Omega} \leftarrow \{\}$, $\boldsymbol{X} \leftarrow \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_\mathcal{J}\}$ and $\boldsymbol{Y} \leftarrow \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_\mathcal{O}\}$
2: Compute MI: Obtain $I(\boldsymbol{x}_i; \boldsymbol{y}_j)$, $\forall \boldsymbol{x}_i \in \boldsymbol{X}$, and $\forall \boldsymbol{y}_j \in \boldsymbol{Y}$
3: Mean MI: $I(\boldsymbol{x}_i; \boldsymbol{Y}) = \frac{1}{\mathcal{O}} \sum_{j=1}^{\mathcal{O}} I(\boldsymbol{x}_i; \boldsymbol{y}_j)$, $\forall \boldsymbol{x}_i \in \boldsymbol{X}$
4: Select the most relevant input: Find the input $\boldsymbol{x}_s = \arg\max_{\boldsymbol{x}_i \in \boldsymbol{X}} I(\boldsymbol{x}_i; \boldsymbol{Y})$, and set $\boldsymbol{\Omega} \leftarrow \{\boldsymbol{x}_s\}$, $\boldsymbol{X} \leftarrow \boldsymbol{X} \setminus \{\boldsymbol{x}_s\}$
5: Greedy selection:
for $t = 1 \rightarrow \mathcal{K} - 1$ **do**
  Compute the conditional MI $I(\boldsymbol{x}_i; \boldsymbol{y}_j | \boldsymbol{\Omega})$, $\forall \boldsymbol{x}_i \in \boldsymbol{X}$, and $\forall \boldsymbol{y}_j \in \boldsymbol{Y}$
  Obtain the mean conditional MI $I(\boldsymbol{x}_i; \boldsymbol{Y} | \boldsymbol{\Omega}) = \frac{1}{\mathcal{O}} \sum_{j=1}^{\mathcal{O}} I(\boldsymbol{x}_i; \boldsymbol{y}_j | \boldsymbol{\Omega})$, $\forall \boldsymbol{x}_i \in \boldsymbol{X}$
  Find $\boldsymbol{x}_s = \arg\max_{\boldsymbol{x}_i \in \boldsymbol{X}} I(\boldsymbol{x}_i; \boldsymbol{Y} | \boldsymbol{\Omega})$, and set $\boldsymbol{\Omega} \leftarrow \{\boldsymbol{x}_s\}$, $\boldsymbol{X} \leftarrow \boldsymbol{X} \setminus \{\boldsymbol{x}_s\}$
end for
6: Output the set $\boldsymbol{\Omega}$

the whole set of outputs $\boldsymbol{Y}$ given $\boldsymbol{\Omega}$. Specifically, the conditional MI for an input-output pair is obtained by approximating $I(\boldsymbol{x}_i; \boldsymbol{y}_j | \boldsymbol{\Omega})$ as follows (more details are given in [96])

$$I(\boldsymbol{x}_i; \boldsymbol{y}_j | \boldsymbol{\Omega}) = I(\boldsymbol{x}_i; \boldsymbol{y}_j) - \frac{I(\boldsymbol{y}_j; \boldsymbol{\Omega})}{H(\boldsymbol{\Omega})} I(\boldsymbol{\Omega}; \boldsymbol{x}_i), \tag{3.6}$$

where $H(\boldsymbol{\Omega})$ represents the entropy of $\boldsymbol{\Omega}$. Note that if the problem has multiple outputs (e.g., the robot actions are represented as acceleration commands at task level), a different subset $\boldsymbol{\Omega}_j$ is defined for each $\boldsymbol{y}_j$. Here, it is assumed that every output equally influences the satisfactory accomplishment of the task, thus mean MI values are computed across all the subsets $\boldsymbol{\Omega}_j$.

3.3.3 Automatic selection of input variables

As shown previously, the number of inputs $\mathcal{K}$ to be selected was predefined in advance, however it would be desirable to have a measure to decide on the optimal number of components selected by Algorithm 1. In this direction, let us define a new variable ζ^t that computes the ratio at iteration t between the information a candidate input variable $\boldsymbol{x}_c^t$ provides and the one already given by the current subset of selected perceptions $\boldsymbol{\Omega}^t$ as follows,

$$\zeta^t = \frac{I(\boldsymbol{x}_c^t, \boldsymbol{Y} | \boldsymbol{\Omega})}{I(\boldsymbol{\Omega}^t, \boldsymbol{Y})}, \tag{3.7}$$

where $\boldsymbol{x}_c^t = \arg\max_{\boldsymbol{x}_i \in \boldsymbol{X}} I(\boldsymbol{x}_i, \boldsymbol{Y} | \boldsymbol{\Omega})$ and $I(\boldsymbol{\Omega}^t, \boldsymbol{Y}) = \sum_{j=1}^{t} I(\boldsymbol{\Omega}^j, \boldsymbol{Y} | \boldsymbol{\Omega}^{j-1})$. Such *conditional Mutual Information ratio* shows how much information the next input to be

Algorithm 2 Threshold-driven MIFS-U

1: Initialization: Set $\boldsymbol{\Omega} \leftarrow \{\}$, $\boldsymbol{X} \leftarrow \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{\mathcal{I}}\}$ and $\boldsymbol{Y} \leftarrow \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{\mathcal{O}}\}$

2: Compute MI: Obtain $I(\boldsymbol{x}_i; \boldsymbol{y}_j)$, $\forall \boldsymbol{x}_i \in \boldsymbol{X}$, and $\forall \boldsymbol{y}_j \in \boldsymbol{Y}$

3: Mean MI: $I(\boldsymbol{x}_i; \boldsymbol{Y}) = \frac{1}{\mathcal{O}} \sum_{j=1}^{\mathcal{O}} I(\boldsymbol{x}_i; \boldsymbol{y}_j)$, $\forall \boldsymbol{x}_i \in \boldsymbol{X}$

4: Select the most relevant input: Find the input $\boldsymbol{x}_s = \arg\max_{\boldsymbol{x}_i \in \boldsymbol{X}} I(\boldsymbol{x}_i; \boldsymbol{Y})$, and set $\boldsymbol{\Omega} \leftarrow \{\boldsymbol{x}_s\}$, $\boldsymbol{X} \leftarrow \boldsymbol{X} \setminus \{\boldsymbol{x}_s\}$

5: Threshold-driven selection:

  Find the candidate input $\boldsymbol{x}_c^t = \arg\max_{\boldsymbol{x}_i \in \boldsymbol{X}} I(\boldsymbol{x}_i, \boldsymbol{Y} | \boldsymbol{\Omega})$.

  Compute the conditional Mutual Information ratio ζ^t

  if $\zeta^t \geq \phi$ **then**

    Set $\boldsymbol{\Omega} \leftarrow \{\boldsymbol{x}_c^t\}$, $\boldsymbol{X} \leftarrow \boldsymbol{X} \setminus \{\boldsymbol{x}_c^t\}$

  else

    Output the set $\boldsymbol{\Omega}$

  end

selected provides taking into consideration the accumulated conditional MI given by the current selected variables. In this sense, it is desired that ζ^t is greater than a predefined threshold $0 < \phi \leq 1$, which controls what is the minimum information ratio that allows to select one more input variable (i.e., the minimum mutual information that a variable should provide). It is worth mentioning that this new selection criterion would modify step 5 in Algorithm 1, where the *greedy* selection is now controlled by ζ^t, which is evaluated at each iteration before selecting the next input (see Algorithm 2). Thus, the algorithm keeps selecting variables while the condition $\zeta^t \geq \phi$ is satisfied. Note that the higher ϕ, the more selective the algorithm. See Section 3.4.4 for an example on how this criterion is applied.

3.4 Experimental results

This section presents a comparison between PCA and MI approaches in the context of the *ball-in-box* task, highlighting their potential advantages and drawbacks when applied to LfD scenarios. After, MI results are shown for two additional experimental setups (i.e., the *pouring* and *table assembly* tasks), where more realistic skills are demonstrated to the robot in order to show how the proposed solution performs in real situations.

3.4.1 Ball-in-box task

In this task, the main goal is to take a ball out of a plastic box, based on the haptic perceptions generated by the ball motion. Here, the robot should learn a mapping between the perceived forces/torques and its actions, which are represented as angular velocity commands at joint level. As described in Section 1.4.1, force-based perceptions are fed back to the teacher in order to establish a bidirectional communication channel during the demonstration stage. A first experimental finding derived from the use of haptic feedback is the need for filtering. Since the box is not a perfectly rigid structure, it vibrates as the robot moves. These unwanted vibrations introduce noise in the teleoperation system, leading to unstable behavior. To avoid this, a low-pass digital filter is implemented to greatly reduce all vibration signals, in a similar way as done in [45] for suppressing residual vibrations in flexible payloads carried by robot manipulators. The signals' fundamental frequency is computed by subjecting the container – with the ball inside – to vibrations through a force applied perpendicularly to the container's base, at the front edge of it. Then, the frequency spectrum of the generated data is analyzed, from which the fundamental frequency (7.5Hz) is set as the cutoff frequency of the low-pass filter. Using $MATLAB^{®}$'s FDAtool, the filter is designed by implementing the *Constrained Least Squares technique* of order 75 [42].

After this, several people tested the experimental setting, by teleoperating the robotic arm through the haptic interface while receiving force-torque feedback from the sensor mounted on the robotic wrist. Initially, they teleoperated the robot while feeling both the container's mass and the ball's dynamics. Then, they carried out the same task just feeling the ball's dynamics. All the participants argued that the presence of the container's mass was a very distracting factor making the task more difficult to teach. Thus, the filtering and dynamic compensation are necessary to obtain better demonstrations and to improve the bidirectional communication channel.

3.4.1.1 PCA results

PCA was applied to this task in order to develop a variable selection through feature transform. In this case, only the input variables set (i.e., the perceived forces/torques) is taken into account to find the latent space where the selection process takes place.

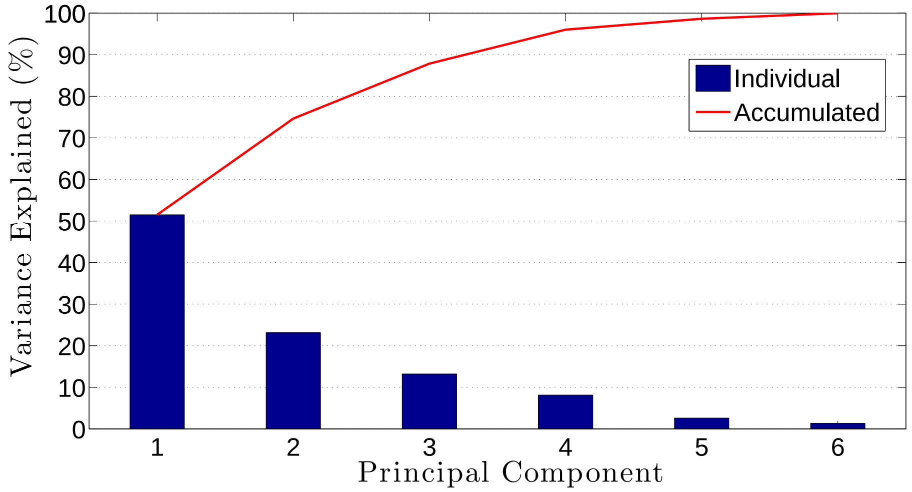


Figure 3.2: Percentage of the variance explained by each principal component in the ball-in-box task.

It should be noted that the forces and torques have different ranges of values which may considerably affect the PCA results. Thus, a variance-based normalization was implemented for every input. After this, the process described at the end of Section 3.2 was carried out.

Figure 3.2 displays the variance explained by each principal component obtained from PCA. It is possible to observe that only one component (i.e., the sixth one) may be removed because the remaining ones are needed to cover at least the 98% of the data's spread, according to the constraint previously explained. This yields a transformation matrix $\boldsymbol{A}$ that projects the datapoints to a latent space whose dimensionality is only one component lower than the original input dataset. Such variable selection is not easy to analyze even for this apparently simple task. It is worth noting that the resulting components are linear combinations of the original input variables and determine the directions along which the variability of the data is maximal, without taking into consideration the outputs of the problem at hand. A discussion about the possible advantages and drawbacks of this technique in the context of the *what to imitate?* problem is given in Section 3.4.1.3.

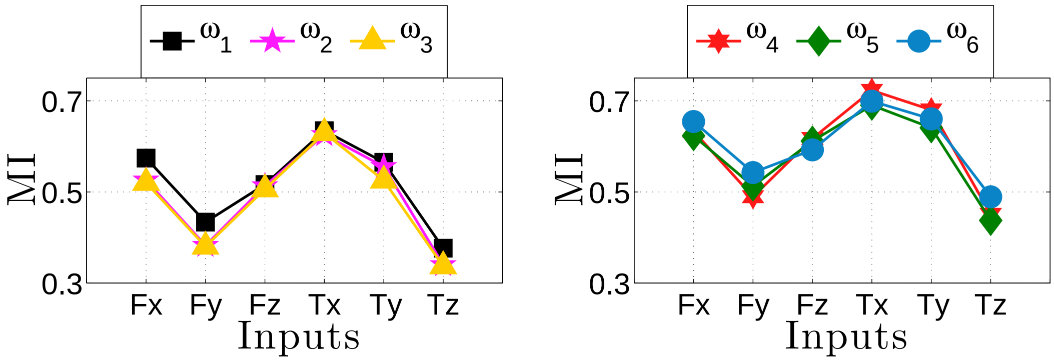
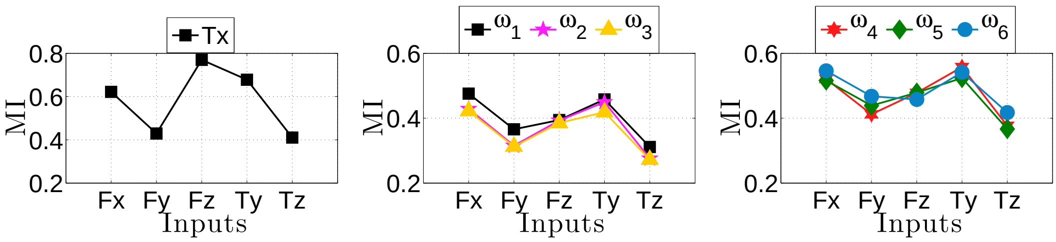


Figure 3.3: MI values for all the input-output pairs of the *ball-in-box* task.

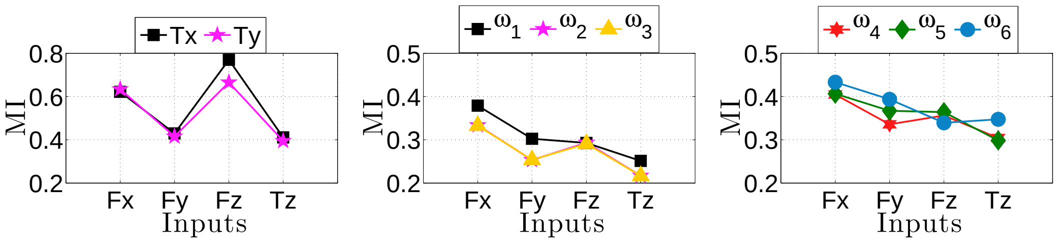
3.4.1.2 MI results

Initially, the most relevant input was found according to the maximal information criterion (steps 1 to 4 in Algorithm 1). Figure 3.3 shows the different MI values for all the input-output pairs in the task. In general terms, the input variables F_y and T_z show less relevance whereas T_x and T_y are the most correlated variables with the outputs. This does make sense as they are the variables that give the most useful information for knowing where the ball is inside the box (see Figure 3.5). These results confirm what is intuitively expected about which input variables were the most relevant for this task. Thus, T_x was chosen as the perception that provides most information with respect to the robot commands executed during the demonstration stage.

Then, the remaining $\mathcal{K} - 1$ input variables are selected according to step 5 of Algorithm 1 (with $\mathcal{K} = 3$). After having chosen T_x, the "conditional relevance" of F_x and T_y keeps high, while the F_z's one is slightly reduced (see Figure 3.4(a)). Then, the algorithm selects T_y, which drastically weakens the importance of F_z, making F_x definitively the best third variable to be selected. Note that initially the MI values for F_x and F_z are very similar for most of outputs if only the maximal relevance criterion is taken into consideration (see Figure 3.3). However, once $\{T_x\ T_y\}$ have been chosen, the information provided by F_z given these two inputs is significantly reduced, due to the high correlation between them (as shown in Figure 3.4(b)). This is in accordance to intuition, since F_z is the force along the vertical axis in the robot frame, which represents the gravitational force of the ball. Such force generates the torques about the axes x and y, and thus F_z and $\{T_x\ T_y\}$ are highly correlated. Finally, the selected



(a) MI between the most relevant input and the remaining perceptions (*left*), and conditional MI for all the input-output pairs given the first selected variable.



(b) MI between the two most informative inputs and the remaining perceptions (*left*), and conditional MI for all the input-output pairs given the two selected variables.

Figure 3.4: Resulting MI values across iterations of MIFS-U for the *ball-in-box* task data.

perceptions for this task corresponded to the subset $\mathbf{\Omega} = \{T_x \ T_y \ F_x\}$.

It should be noted that by plotting the first samples of the resulting "most informative" variables, namely T_x and T_y, it is possible to observe that they do describe where the ball is inside the container. Figure 3.5 shows how these two variables make unequivocal clusters for each starting location of the ball, which confirms that these perceptions provide enough information about its position so that the robot performs accordingly with success in learning and execution phases as shown in Section 4.3.3.

3.4.1.3 Discussion about PCA and MI results

It is clear that PCA and MI are really different approaches, mostly because their feature selection is carried out in different dataspaces using distinct criteria. Here some possible advantages and drawbacks for both techniques are contrasted, and some arguments are provided to justify why one of them may be preferred in LfD tasks.

- Both PCA and MI provide a ranking that is used to carry out the selection process. In this context, the main difference is that PCA indirectly gives the

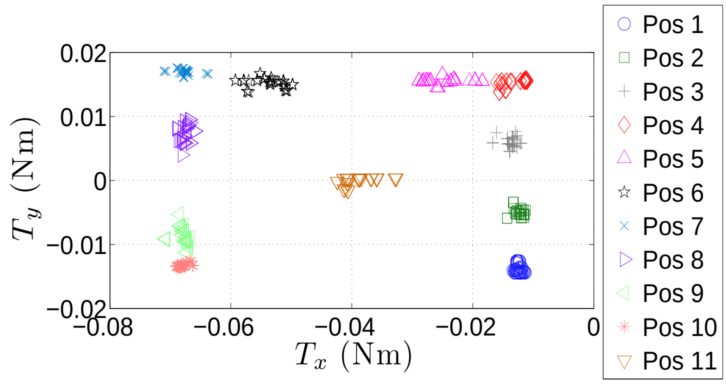


Figure 3.5: Torques map representing clusters for each initial position of the ball inside the container.

number of features to be chosen based on the data's spread criterion (see Section 3.2). Thus, the selection is completely automatic and data-driven. In contrast, MI does need to know the number of inputs to be selected in advance.

- As PCA intrinsically consists in a feature transform, the resulting latent space variables are not easily understandable. In contrast, MI works in the original dataspace, providing the possibility of a straightforward analysis of the performed selection process (as done previously).

- MI carries out the feature selection taking into account how the inputs affect the outputs. This aspect is in the line of control policies in imitation learning, where a mapping function from perceptions to actions of the robot is learned (see Section 2.1.3). Opposed to MI, PCA only studies the correlation among the data variables, no inputs-outputs distinction is taken into consideration.

- In terms of computational cost, the computational complexity of Algorithm 1 is higher than PCA when the data are subjected to the feature selection process before encoding the demonstrations. In contrast, PCA implies that each datapoint is online projected to the latent space during the reproduction of the task. Although this computation is simply a linear combination of the original inputs, it is more time-consuming than MI, which only needs to discard the irrelevant – non-selected – input variables.

- PCA strongly depends on the range of values (or magnitude) of the variables, which of course can be solved by normalizing the data. On the other hand, MI is invariant under space transformations. Such property is based on the fact that the argument of the logarithm in Equation 3.5 is nondimensional, thus the integral value does not depend on the chosen coordinates [48].

Beyond the aforementioned aspects of both algorithms, feature selection approaches may be preferable to feature transforms in LfD when the original inputs contain intuitive information that can be used to convey cues about the task. For instance, in active learning, the robot may let the teacher know which perceptions it has selected, in order to get feedback about how well or how convenient its selection was according to the human knowledge of the task. Such human assistance will not be available if the robot carries out the selection in a transformed dataspace. This fact may occur in [26], where the authors propose to project the human samples onto a latent space obtained from PCA to diminish redundancies, where the transformed variables do not have a straightforward interpretation for a human teacher anymore.

The characteristics previously discussed and the obtained results show that MI better fits the LfD paradigm, and its use raises several interesting challenges to be solved. For instance, in this thesis MI was applied to low-level learning, where the sensory inputs relevance was evaluated from their influence on the robot commands. However, in learning at higher levels, it might be necessary to evaluate how low-level instructions help to accomplish a high-level task goal. It should be noted that MI is totally compatible with other different state-of-the-art approaches that solve the *what to imitate?* issue, as those based on variance information (mentioned at the beginning of this chapter).

The other two tasks described in Sections 1.4.2 and 1.4.3 will be analyzed below using MI exclusively, with the aim of showing how this technique and the proposed algorithm perform on more realistic scenarios.

3.4.2 Pouring task

In the task described in Section 1.4.2, a robotic manipulator learns to pour drinks using its force perceptions exclusively. Similarly to the *ball-in-box* task, the demonstrations

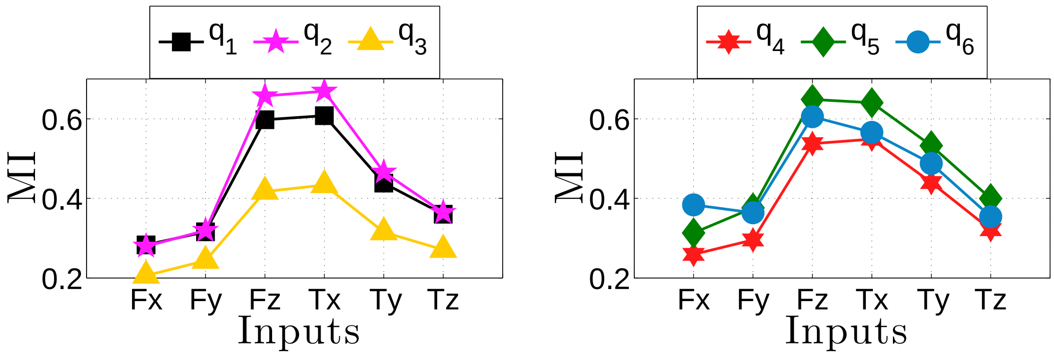
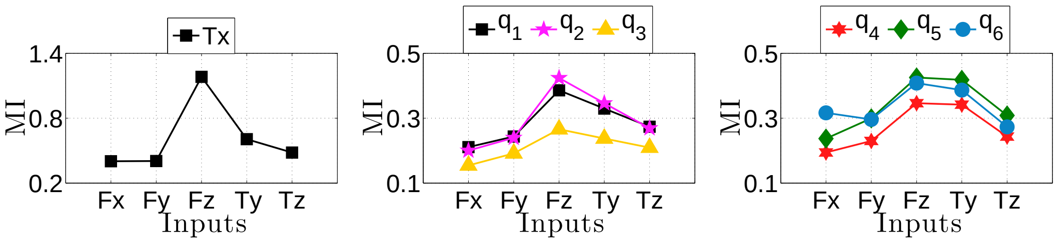


Figure 3.6: MI values for all the input-output pairs of the *pouring* task.

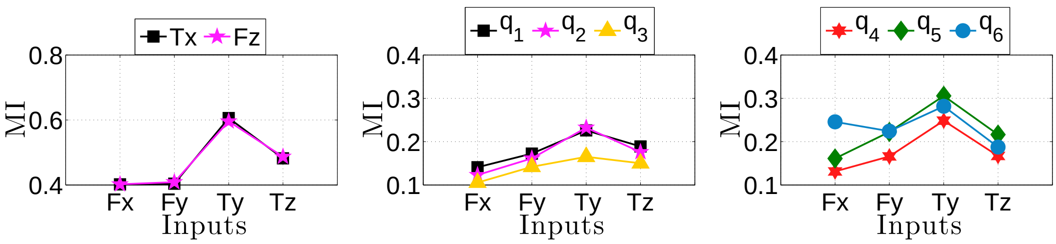
are carried out by teleoperating the robot through a force reflecting device. Regarding the signal processing, a smoothing filter was implemented to reduce the noise from the sensor readings, mainly generated by the tiny metallic spheres. Moreover, the dynamic compensation model previously presented was used here for removing the bottle mass effects from the sensor readings, in order to feed back the teacher with only the external forces-torques generated by the "fluid" at the demonstration phase.

Here, inputs are the forces and torques conveying information about the fluid in the bottle, while outputs are the desired robot joint position to be achieved. Again, the MI value was computed for all the input-output pairs in order to select the most important input variable, T_x in this case (see Figure 3.6). Note that T_x and F_z display nearly the same MI value for all the robot joints. This is an expected result because F_z is the vertical force in the robot frame representing the gravitational component of the load (i.e., the bottle and fluid masses), while T_x is approximately the torque generated by such a load. This means that both variables are providing similar information with respect to the robot movements, because they significantly vary as the fluid comes out of the bottle.

Subsequently, Algorithm 1 was applied to select the remaining $\mathcal{K}-1$ variables (with $\mathcal{K}=3$), from which the resulting "most informative" set of inputs was $\boldsymbol{\Omega} = \{T_x\ F_z\ T_y\}$. The selection process and the values of the conditional mutual information are shown in Figures 3.7(a) and 3.7(b). There is an interesting aspect to highlight from these results: the third selected variable T_y shows slight variations when the robot rotates the bottle to pour a drink, which are likely produced by the location change of the



(a) MI between the most relevant input and the remaining perceptions (*left*), and conditional MI for all the input/output pairs given the first selected variable.



(b) MI between the two most informative inputs and the remaining perceptions (*left*), and conditional MI for all the input/output pairs given the two selected variables.

Figure 3.7: Resulting MI values across iterations of MIFS-U for the *pouring* task data.

center of mass of the load due to the "fluid dynamics" into the bottle. Note that such dynamics may be hardly modeled as reported in [154], but the algorithm was able to detect that T_y was non-linearly correlated to the robot motion encapsulating part of the fluid dynamics (also confirmed after a detailed analysis of the data streams).

Another point to mention is that, in some cases, MIFS-U still gives more preference to redundant variables over irrelevant ones during the selection process, despite it is aimed at reducing the redundancy among the selected inputs [48]. In this case the variables T_x and F_z provide nearly the same information about the task, but both are chosen even being redundant, because their relevance with respect to the outputs keeps higher than that of irrelevant variables, despite one of them has been selected previously.

3.4.3 Collaborative table assembly task

In contrast to the above experiments, MI is applied here to a collaborative task where the robot should learn distinct reactive behaviors through the physical interaction with a human partner as described in Section 1.4.3. Here, the data streams consist of

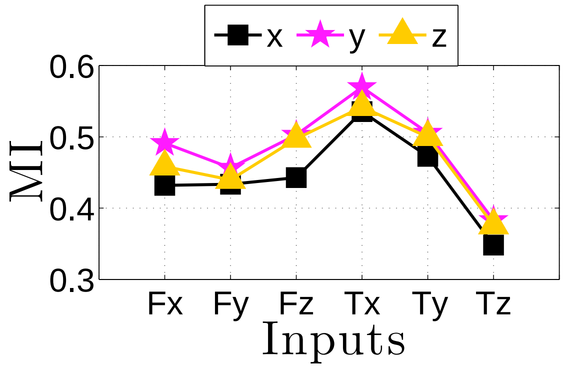


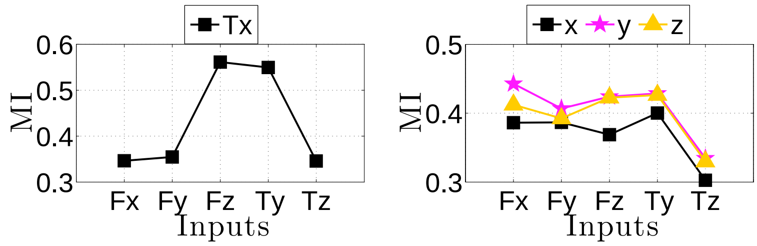
Figure 3.8: MI values for all the input-output pairs of the *table assembly* task.

the sensed haptic inputs and the robot motion in its operational space during the demonstrations of the task. The robot is controlled at the execution phase through force commands that partially depend on stiffness matrix estimates that are obtained once the model has successfully encoded the different behaviors (details are given in Section 5.2).

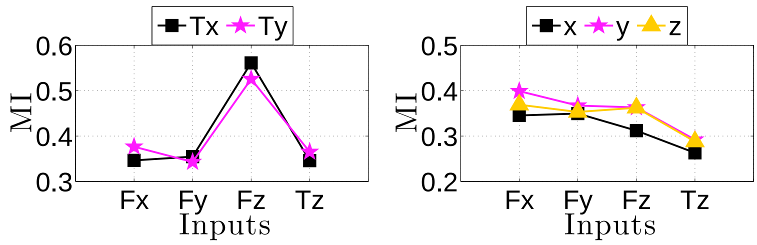
In order to work only with those forces generated by the interaction between the partners through the table, it is necessary to remove any other kind of force-based signals from the sensor readings (e.g., screwing the table to the sensor produces an offset in the frame of reference of the sensor). Again, a smoothing filter was implemented to reduce the intrinsic noise of the sensor and the small vibration effects generated by the table. Subsequently, a dynamic compensation of the forces/torques generated by the table mass was implemented using the model shown in Figure 3.1.

The MI analysis is implemented using the positional information of the robot end-effector (i.e., the output data) along with the perceived forces/torques (i.e., the inputs). The selected input variables should contain enough information so that the learning model can correctly encode the demonstrations, and so that the subsequent estimations properly encapsulate the different compliance levels that the robot should adopt.

Figure 3.8 displays the MI values for all the haptic inputs with respect to the three positional axes describing the end-effector location. It is possible to observe that the most relevant input is T_x, and also that F_x, F_z and T_y nearly provide the same quantity of information individually, thus it is not clear a priori which of these three should be selected according to the maximal information criterion if $\mathcal{K} = 3$. Again, the remaining



(a) MI between the most relevant input and the remaining perceptions (*left*), and conditional MI for all the input/output pairs given the first selected variable.



(b) MI between the two most informative inputs and the remaining perceptions (*left*), and conditional MI for all the input/output pairs given the two selected variables.

Figure 3.9: Resulting MI values across iterations of MIFS-U for the *table assembly* task data.

$\mathcal{K} - 1$ perceptions are chosen through the conditional MI, from which the resulting subset $\mathbf{\Omega} = \{T_x \; T_y \; F_x\}$ is found. Figure 3.9 displays the MI values obtained using the proposed algorithm.

It is worth highlighting the prominent – and nearly the same – correlation between T_x and the subset $\{F_z \; T_y\}$. On the one hand, the interdependence between the torque about the axis x and the force along z is clear, as this force is orthogonal to the plane defined by the axes x and y. On the other hand, T_x and T_y also show to be highly correlated (see Figure 3.9(a)). This can be explained by the fact that both variables become noticeable higher/lower when the human starts/finishes to screw a leg, indicating that a robot behavior switching is needed (which is also the role of F_z).[1]

Lastly, note that the force along the axis x is the third selected input, due to the high correlation between the subset $\{T_x \; T_y\}$ and F_z significantly reduced the information provided by the force along z, as shown in Figure 3.9(b).

[1]Note that the haptic pattern described by $\{T_x \; T_y\}$ unambiguously specifies the table thread locations.

3.4.4 Testing the automatic selection criterion

Again every training dataset of manipulation tasks previously presented was subjected to MI analysis, but this time using the Algorithm 2 with $\phi = 0.4$. This value was chosen among a conservative range experimentally defined by $0.3 \leq \phi \leq 0.7$, in order to avoid the selection process to be too much selective by removing still relevant variables for the problem. Regarding the *ball-in-box* task, the resulting subset of selected inputs was again $\mathbf{\Omega} = \{T_x \ T_y \ F_x\}$, supporting the analysis explained in Section 3.4.1. In contrast, for the *pouring* task, the resulting selected perceptions were $\mathbf{\Omega} = \{T_x \ F_z\}$. This tells that T_y might not provide enough information about the robot actions when T_x and F_z have been already selected. In fact, T_y shows the highest correlation when T_x and F_z were selected previously (see Figure 3.7), therefore the proposed automatic selection criterion does allow to add this input variable to the set $\mathbf{\Omega}$ (this also depends on the value given for ϕ). Similarly, the perceptions selected for the *collaborative assembly* task this time correspond to the set $\mathbf{\Omega} = \{T_x \ T_y\}$. Here, the proposed criterion showed that F_x does not provide enough additional information to be selected. In this case, due to the way the task is performed, the chosen variables already convey the data needed to know the relative position of the threads with respect to the end-effector frame.

3.5 Chapter highlights

The *what to imitate?* problem was solved from a new perspective, by using MI-based inputs selection. Results showed that this technique is appropriate to find which input variables are the most relevant to learn a task. This presents several advantages in LfD settings: reduction of input space dimensionality, less computational cost and probably faster training and execution stages. This method can also be applied before finding the demonstration segments with low variability that indicate those sections that must be learned, jointly working with approaches such as the one proposed in [25]. It should be noted that this variance-based approach is more suitable for trajectory learning (i.e., low-level encoding), whereas the solution proposed in this thesis is more generic and may be applied to a larger set of tasks.

Every learning framework proposed in this thesis (see Section 1.3) uses this input selection technique, and the selected force-based input variables are further used in the learning and reproduction phases. As shown in the next chapters, each task is successfully encoded and reproduced using the input subsets obtained in this chapter, demonstrating that the chosen haptic perceptions contain enough information for the robot to perform as expected. Therefore, Mutual information-based selection is shown to be a suitable tool in LfD frameworks dealing with force-based tasks.

Chapter 4

Learning manipulation tasks from haptic inputs

When robots interact with their surroundings and/or with humans, force-based perceptions arise as a rich and valuable source of information about what is going on over the course of such processes. Assembly and grasping tasks are classical examples where force data are often necessary to represent their different states [72, 148], which condition the robot actions. Likewise, haptic cues provide relevant information about intentions and roles in pHRI scenarios. However, in spite of its remarkable importance, this type of sensory input has been seldom used and exploited in LfD.

This chapter aims at filling this gap by analyzing how some state-of-the-art algorithms work on the force domain, and by proposing a compact – and possibly extendable – framework able to learn skills from haptic inputs. The learning methods are studied using as test bed the *ball-in-box* task described in Section 1.4.1. The performance of the proposed framework is also evaluated in depth through this task. Furthermore, an extension of the learning framework is analyzed, that takes *task parameters* into account (see Figure 1.2). The *pouring task* described in Section 1.4.2 constitutes a simple and realistic scenario to evaluate this extended framework.

4.1 From positional information to haptic cues

In general terms, the robot skills are encoded by a policy representing a mapping $\boldsymbol{u}(t) = \boldsymbol{\Pi}(\boldsymbol{z}(t), t, \boldsymbol{\zeta})$, where $\boldsymbol{u}(t)$ represents the robot actions (e.g., motor commands),

$\boldsymbol{z}$ is the internal state of the robot and the state of the environment, t is time and $\boldsymbol{\zeta}$ is the set of parameters to be learned to shape the mapping function $\boldsymbol{\Pi}$ [5]. This policy may be considered as a non-autonomous system, because it is explicitly time dependent. Obviously, such representation is very restricting and little useful for reproducing complex and dynamic tasks, because such explicit time-based dependence limits the generalization capabilities of the model and it is not very robust in coping with unforeseen perturbations of the task [143]. Thus, autonomous systems are preferred, i.e., the ones driven by output commands $\boldsymbol{u}(t) = \boldsymbol{\Pi}(\boldsymbol{z}(t), \boldsymbol{\zeta})$.

Therefore, the problem is to learn the parameters $\boldsymbol{\zeta}$ of the policy $\boldsymbol{\Pi}$ from imitation, as explained in Section 2.4. The learned policy governs the robot behavior through output commands expressing a "desired time-derivative", that is, implying a desired change of the state information,

$$\dot{\boldsymbol{z}}(t) = \boldsymbol{\Pi}(\boldsymbol{z}(t), \boldsymbol{\zeta}). \tag{4.1}$$

It should be noted that a skill is frequently defined either in the internal coordinates of the robot (i.e., joint angular coordinates $\boldsymbol{q}$) $\dot{\boldsymbol{q}}(t) = \boldsymbol{\Pi}(\boldsymbol{z}(t), \boldsymbol{\zeta})$, or using a task-level representation (i.e., the position/orientation of the robot's end-effector $\boldsymbol{x}$) $\dot{\boldsymbol{x}}(t) = \boldsymbol{\Pi}(\boldsymbol{z}(t), \boldsymbol{\zeta})$. Both approaches use all the possible state information about the robot motion and the environment as input, but only output a variable that is the desired change of state of the robot in the selected coordinate system.

Research on LfD has typically focused on learning and reproducing tasks at trajectory level. In this context, time and space constraints govern the skill to be learned by the robot. Gesture reproduction [22], pick-and-place skills [72], writing [92] and full body imitation [91], among others, are common examples of tasks that can be represented onto spatio-temporal dataspaces. However, these works do not fully exploit the robustness and generic properties of the representation given in Equation 4.1, where the $\boldsymbol{z}$ variable may contain much more information than only positional data provided by proprioceptive sensors. In contrast, in this chapter we use a richer representation, first by using exclusively force data as variables of the task (see Section 4.3) and then by adding to it the current state of the robot (see Section 4.4). Generative models are used to learn the policies $\boldsymbol{\Pi}$ in each case, as shown in next sections.

4.2 Encoding and reproduction of a force-based task

In this chapter, the learning of the policy $\boldsymbol{\Pi}$ is considered as a problem of finding a mapping function between states $\boldsymbol{z}(t)$ (i.e., force perceptions) and output commands $\boldsymbol{u}(t)$ (i.e., position changes). Here, regression techniques are used to map demonstration states to continuous action spaces, controlling low-level robot motions. Two different approaches are considered: *(i)* locally weighted projection regression [159], and *(ii)* Gaussian mixture regression [26]. Hence, next sections are aimed at analyzing the possible advantages and drawbacks of both algorithms in the context of LfD of force-based manipulation skills, where they have been scarcely applied in the past.

4.2.1 Regression-based learning

Locally-weighted projection regression is an incremental regression algorithm that performs piecewise linear function approximation and may be currently considered as the standard real-time learning method in robot control applications [142]. The algorithm does not require storage of the training data and has been proved to be efficient in a variety of robot learning tasks including high dimensional data [160]. By detecting locally redundant or irrelevant input dimensions, the method locally reduces the dimensionality of the input data by finding local projections through Partial Least Squares regression [54].

Specifically, LWPR predicts the target values (i.e., the robot actions) for a given perception value $\boldsymbol{x}$ through a combination of K individually weighted locally linear models. The weighted prediction $\hat{y}$ is given by

$$\hat{y}(\boldsymbol{x}) = \frac{\sum_{k=1}^{K} w_k \bar{y}_k(\boldsymbol{x})}{\sum_{k=1}^{K} w_k}, \tag{4.2}$$

with $\bar{y}_k(\boldsymbol{x}) = \bar{\boldsymbol{x}}_k^\top \boldsymbol{\theta}_k$ and $\bar{\boldsymbol{x}}_k = [(\boldsymbol{x} - \boldsymbol{c}_k)^\top, 1]^\top$, where w_k is the weight or attributed responsability of the model, $\boldsymbol{\theta}_k$ contains the estimated parameters of the model and $\boldsymbol{c}_k$ is the center of the k-th linear model.[1] The weight w_k determines whether a datapoint falls into the region of validity of the model k, similar to a receptive field, and is usually characterized with a Gaussian kernel (more details in [159]). During the learning

[1] The number of local linear models is automatically adapted by the incremental learning process.

process, both the shape of the receptive fields and the parameters of the local models are adjusted such that the error between the predicted values and the observed targets is minimal. At this point, it is worth mentioning the most known drawback of this technique, namely that it requires *skillful tuning of the meta parameters for the learning process in order to achieve competitive performance* [117]. This aspect prevents LWPR from being applied to friendly LfD frameworks, because the teacher may likely take too long to find the correct parameter values. Nevertheless, once these values have been found, LWPR has shown good computational results – in terms of mean squared error – when learning a simple force-based skill, namely the ball-in-box task [135].

On the other hand, assuming the aforementioned issue was solved, there are still more aspects to be considered. Force-based manipulation skills might be characterized by perceptual aliases, this means that during the task the robot may carry out different actions for the same perception pattern [19]. In other words, the policy $\mathbf{\Pi}$ would correspond to a *multivalued* function, and in such case LWPR is not applicable anymore. Several solutions may be proposed to cope with this problem, for instance, the robot may disambiguate the context by keeping some state information, taking into account the time-series nature of data. This approach would considerably increase the dimensionality of the dataspace, making the tuning process even harder. A different solution may be to include time as an additional input variable, at the cost of limiting the generalization capabilities of the learning framework, as discussed previously. Despite several works have benefited from LWPR capabilities, its drawbacks entail to consider other techniques able to perform successfully in this kind of situations.

4.2.2 Probabilistic learning

GMM is an algorithm belonging to the family of generative stochastic models, and it has been extensively used to encode low-level action primitives in LfD (as explained in Section 2.4.1). Broadly speaking, the main idea is to represent the set of provided demonstrations $\boldsymbol{O}$ as a mixture of N_c multivariate Gaussian distributions,

$$p(\boldsymbol{O}) = \sum_{i=1}^{N_c} p(i)p(\boldsymbol{O}|i), \tag{4.3}$$

where $p(i)$ is a prior and $p(\boldsymbol{O}|i)$ is a conditional probability density function. The parameters in Equation 4.3 are defined by $p(i) = \pi_i$ and $p(\boldsymbol{O}|i) = \mathcal{N}(\boldsymbol{O}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, with $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ as the mean and covariance matrix of the Gaussian distribution i. These parameters are estimated through the Expectation-Maximization algorithm (**EM**) [40]. This method carries out a local search that guarantees a monotone increase of the likelihood during optimization. Moreover, EM is a simple iterative method which alternates between inferring the missing values given the parameters (E step), and then optimizing the parameters given the data (M step). The iteration stops when the increase in the log-likelihood at each iteration becomes too small, i.e., when

$$\frac{\mathcal{L}^{t+1}}{\mathcal{L}^{t}} < \phi, \quad \text{where} \quad \mathcal{L} = \sum_{i=1}^{N_c} \log\left[p(i)p(\boldsymbol{O}|i)\right]. \tag{4.4}$$

Once data have been encoded, Gaussian mixture regression is used to retrieve smooth robot actions from a set of input perceptions. In other words, the GMM/GMR framework carries out the state-action mapping, representing the policiy $\boldsymbol{\Pi}$. Specifically, the aim of GMR is to estimate the conditional expectation of an output $\boldsymbol{y}$ given $\boldsymbol{x}$ on the basis of a set of demonstrations $\boldsymbol{O}$,

$$p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{i=1}^{N_c} \beta_i \left[\boldsymbol{\mu}_i^{\boldsymbol{y}} + \boldsymbol{\Sigma}_i^{\boldsymbol{yx}}(\boldsymbol{\Sigma}_i^{\boldsymbol{xx}})^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i^{\boldsymbol{x}})\right], \tag{4.5}$$

where $\beta_i = \frac{p(i)p(\boldsymbol{x}|i)}{\sum_{j=1}^{N_c} p(j)p(\boldsymbol{x}|j)}$, and considering that the parameters of each Gaussian i can be expressed as follows

$$\boldsymbol{\mu}_i = \begin{bmatrix} \boldsymbol{\mu}_i^{\boldsymbol{x}} \\ \boldsymbol{\mu}_i^{\boldsymbol{y}} \end{bmatrix}, \quad \boldsymbol{\Sigma}_i = \begin{bmatrix} \boldsymbol{\Sigma}_i^{\boldsymbol{xx}} & \boldsymbol{\Sigma}_i^{\boldsymbol{xy}} \\ \boldsymbol{\Sigma}_i^{\boldsymbol{yx}} & \boldsymbol{\Sigma}_i^{\boldsymbol{yy}} \end{bmatrix}. \tag{4.6}$$

The input $\boldsymbol{x}$ and the output $\boldsymbol{y}$ represent the state $\boldsymbol{z}(t)$ and the robot actions $\boldsymbol{u}(t)$ in the context of robot LfD. Note that in this framework the only open parameter to be tuned is the number of GMM components, which may be predefined by the user, or better obtained by methods like the Bayesian Information Criterion (**BIC**) [26], or through more recent approaches like the ones based on Dirichlet processes [87]. Here the regression function is not approximated directly, in contrast to other regression approaches (e.g., LWPR). Instead, the joint density of the set of demonstrations is first estimated by a model from which the regression function is derived. Unlike LWPR, the statistical nature of GMM allows to capture the data variability in the demonstration phase, and thus there is not a unique trajectory which can be used for action reproduction through GMR. This is one of the main advantages of GMM/GMR, since it generates at the same time a mean response estimate and a covariance response estimate, which may be used for reproduction and handling of constraints [25], or for estimating the robot joint compliance [99]. It should be noted that incremental versions of this approach have been also proposed [24, 31]. Again, the computational results based on the mean squared error (MSE) for the ball-in-box task were satisfactory [134, 135]. At this point, it is worth mentioning that the MSE-based performance criterion may provide a very low value even if there are high prediction errors for datapoints belonging to multivalued cases. This is possible because MSE averages all error values along the query datastream. This fact was more evident when these algorithms were tested on the real setting for practical experimentation, due to both of them failed when the robot faced a multivalued point. Thus, the MSE should be used along with another criteria (e.g., a task goal-based score) in order to evaluate the performance of the robot [136, 138].

In spite of the aforementioned advantages, the GMM/GMR framework also fails when facing tasks whose underlying behavior corresponds to a multivalued function. Again, the previously proposed solutions to overcome this problem are also applicable here, but sharing similar drawbacks. For instance, if time is included, the generalization capabilities are considerably degraded and a previous signal processing is needed (e.g., dynamic time warping [114]) in order to carry out a temporal alignment of the different demonstrations. On the other hand, note that when GMM deals with high dimensionality dataspaces, a large number of local minima may exist, which makes converging to a "good" local minimum much more difficult. In such case, the GMM/GMR may benefit from the MI analysis presented in the previous chapter, which allows to reduce the dimensionality of the dataspace by selecting the most relevant input variables of the problem at hand (see Section 3.3).

The previous analysis and the obtained results initially suggest that a suitable and generic algorithm for learning force-based skills would be one that *(i)* encodes the temporal information of a task without including time explicitly, *(ii)* encapsulates the data variability and *(iii)* does not include too many open parameters to be tuned. Such requirements are completely fulfilled by a hidden Markov model with Gaussian components. This algorithm is introduced in the next section along with a modified version of GMR, composing a learning framework (see Section 1.3) able to statistically encode force-based tasks while exploiting the time-series nature of the demonstrations. A detailed analysis of the learning and reproduction phases is also given.

4.3 Exploiting implicit sequential information

Previous research in LfD has proposed to use GMM for encoding manipulation tasks. However, as stated before, this algorithm does not extract temporal information from data, and time must explicitly be considered as an input variable if required by the type of task to be learned (as in Calinon *et al.* [26]). Force-torque signals tend to show very large time discrepancies, for instance, intrinsic noise, highly dynamic movements and external forces may generate different haptic signal profiles across demonstrations. This makes a specific force perceived at a given time step be different for several human examples. Also, note that a task exclusively based on force data may not depend solely on time as trajectory learning does, where velocity and/or accelerations constraints appear. Such problem may be tackled using techniques as dynamic time warping [114], at the price of increasing the complexity of the learning framework. Instead, HMM is used here to avoid including such explicit temporal dependency in the model: it exploits the sequential patterns in the data and it is therefore more appropriate to encode the features of force-based tasks without using time as an additional input variable, which would significantly constrain the generalization capabilities. In other words, HMM is able to encode implicit time constraints of the task, which may depend on a specific sequence of actions to be carried out by the robot irrespective of their duration.

HMM can be interpreted as an extension of GMM in which the choice of the mixture component for each observation depends also on the choice of the component for the previous observation. Hence, this algorithm is also able to encapsulate the data variability as GMM does. The HMM has been widely used in several computer science areas as speech recognition [128], human motion patterns encoding [11] and LfD applications [89, 99], among others. Most of LfD works use HMM to learn trajectories from human demonstrations [50] or to encode a task with predefined states as in assembly processes that can be represented at a symbolic level [43]. However, the tasks addressed in this chapter differ from these and other works in the following points:

(i) the task goal may be achieved by executing different trajectories depending on the initial conditions of the task (e.g., initial position of the ball inside the container for the *ball-in-box* task or the fluid quantity inside the bottle for the *pouring* skill), *(ii)* sequential information is exploited using the observed force-torque patterns generated over the course of the task (time is not explicitly used as an additional input variable).

4.3.1 Encoding using hidden Markov models

Formally, each demonstration $m \in \{1, \ldots, M\}$ contains T_m datapoints forming a training dataset $\boldsymbol{O} = \{\boldsymbol{\delta}_t\}_{t=1}^{T}$, with $T = \sum_m^M T_m$. Each datapoint (or observation) $\boldsymbol{\delta}_t \in \mathbb{R}^D$, with D as the total number of input and output variables, i.e., the state $\boldsymbol{z}(t)$ and the robot actions $\boldsymbol{u}(t)$, respectively. An ergodic HMM of N_c components is defined as $\lambda = (\{a_{ij}\}, \{b_j(\boldsymbol{\delta}_t)\}, \boldsymbol{\pi})$ where:

- $\{a_{ij}\}$ is the state transition probability matrix, with $1 \leq i, j \leq N_c$.
- $\{b_j(\boldsymbol{\delta}_t)\}$ is a continuous observation probability density defined as a normal distribution $\mathcal{N}(\boldsymbol{\delta}_t; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, where $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ are respectively its center and covariance matrix.
- $\boldsymbol{\pi} = \{\pi_i\}$ is the initial state probability vector, with $1 \leq i \leq N_c$.

The main idea is to adjust the model to maximize $P(\boldsymbol{O}|\lambda)$. To achieve this, an EM process is applied to HMM, which is also know as the *Baum-Welch* method (more details in [127]). In order to describe the procedure for re-estimation of HMM parameters, it is necessary to define the following variables:

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(\boldsymbol{\delta}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N_c} \sum_{j=1}^{N_c} \alpha_t(i) a_{ij} b_j(\boldsymbol{\delta}_{t+1}) \beta_{t+1}(j)}, \tag{4.7}$$

$$\gamma_t(i) = \sum_{j=1}^{N_c} \xi_t(i,j), \tag{4.8}$$

where α and β are called *forward* and *backward* variables, respectively, and are defined as:

$$\alpha_1(i) = \pi_i b_i(\boldsymbol{\delta}_1), \quad \alpha_{t+1}(j) = \left[\sum_{i=1}^{N_c} \alpha_t(i) a_{ij}\right] b_j(\boldsymbol{\delta}_{t+1}) \quad \text{and} \tag{4.9}$$

$$\beta_T(i) = 1, \quad \beta_t(i) = \sum_{j=1}^{N_c} a_{ij} b_j(\boldsymbol{\delta}_{t+1}) \beta_{t+1}(j). \tag{4.10}$$

From equations 4.7 and 4.8, the HMM parameters are iteratively estimated as follows

$$\bar{\pi}_i = \gamma_1(i), \quad \bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)},$$

$$\bar{\boldsymbol{\mu}}_i = \frac{\sum_{t=1}^{T} \gamma_t(i) \boldsymbol{\delta}_t}{\sum_{t=1}^{T} \gamma_t(i)}, \quad \bar{\boldsymbol{\Sigma}}_i = \frac{\sum_{t=1}^{T} \gamma_t(i)(\boldsymbol{\delta}_t - \boldsymbol{\mu}_i)(\boldsymbol{\delta}_t - \boldsymbol{\mu}_i)^\top}{\sum_{t=1}^{T} \gamma_t(i)}.$$

Again, the stopping criterion previously introduced (see Eq. (4.4)) is used to determine convergence of the algorithm. These equations permit obtaining a suitable trained HMM that represents the teacher demonstrations statistically through a model capturing the robot motion for given force-based perceptions and taking temporal coherence into account from the resulting transitions $\{a_{ij}\}$. The only open parameter is again the number of components of the model. In this chapter this parameter was found by using the BIC as explained later on.

It is worth noting that the *Baum-Welch* algorithm, as an EM process, shares one of the properties that all EM iterative methods have, namely, it cannot guarantee that a global minimum is found. The initial estimate is thus important. In this case, starting from a rough initialization of the Gaussian components parameters by *k-means* has shown to provide convergence to "good" local minima in terms of the log-likelihood [112]. The resulting models satisfactorily encapsulate the data and provide good regression properties.

4.3.2 Reproduction using Gaussian mixture regression

Since the tasks are neither strictly learned as a sequence of discrete actions nor as simple trajectories, it is necessary to find a suitable way to reconstruct the output commands, given a perception, the resulting trained HMM and by taking into account the temporal coherence of the data. To achieve this goal, a modified version of GMR (here named GMRa) is used for computing the robot actions to be sent to the controller as the desired robot state to be achieved, as described next.

The standard GMR averages the different observations, even if they have been observed at different parts of the skill. This method does not take advantage of the temporal information encapsulated in the HMM. To overcome this drawback, the weighting technique proposed in [27] is adopted here, where the robot's actions are computed from a modified version of the well-known GMR. This version computes the predictions from a mixture of Gaussians (in this case the HMM components) taking the encapsulated temporal information by the HMM (i.e., the variable α) into account along with the given inputs (i.e., the robot perceptions). In this way, the proposed learning framework is able to handle perceptual aliases by itself, this means that the robot may be able to carry out the correct action if more than one output exist for the same perception pattern, by taking advantage of the sequential information of the task. This makes the proposed structure more generic and versatile, thus being useful for a wider set of manipulation skills.

In GMRa, the weights are estimated using the actual values of the inputs (mainly force-torque data in the tasks implemented in this chapter), and also implicitly their previous values, through the transition probabilities related to the forward variable α. Formally, the definition of the new regression based on temporal information is given by:

$$\hat{\boldsymbol{y}} = \sum_{i=1}^{N_c} \alpha_t^{\boldsymbol{x}}(i) \left[\boldsymbol{\mu}_i^{\boldsymbol{y}} + \boldsymbol{\Sigma}_i^{\boldsymbol{yx}} (\boldsymbol{\Sigma}_i^{\boldsymbol{xx}})^{-1} (\boldsymbol{x}_t - \boldsymbol{\mu}_i^{\boldsymbol{x}}) \right] , \tag{4.11}$$

where $\alpha_t^{\boldsymbol{x}}(i)$ is the forward variable for the i-th Gaussian in the HMM (see Equation 4.9), and is computed with the input parts $\boldsymbol{x}$ of the observation vector. This variable expresses the probability of observing the partial sequence, $\boldsymbol{O} = \{\boldsymbol{\delta}_1\ \boldsymbol{\delta}_2\ \ldots\ \boldsymbol{\delta}_t\}$ and of being in HMM component S_i at time t. Now, for a given force-torque perception, the predicted command is based on current and past observations, which makes sense for those tasks where more than one output exists for a given input pattern.

Note that Lee and Ott's work [99] proposes a similar framework that encodes the demonstrations through an HMM and retrieves the robot actions using a time-driven version of the classical GMR. In contrast to our *forward variable*-based weights, the weighting mechanism used by GMR exclusively depends on time, and neither previous observations nor sequential information are taken into account in this approach. Nevertheless, their approach allows to handle data sequences of different duration and speed, avoiding to use a preprocessing of the observation data such as dynamic time warping.

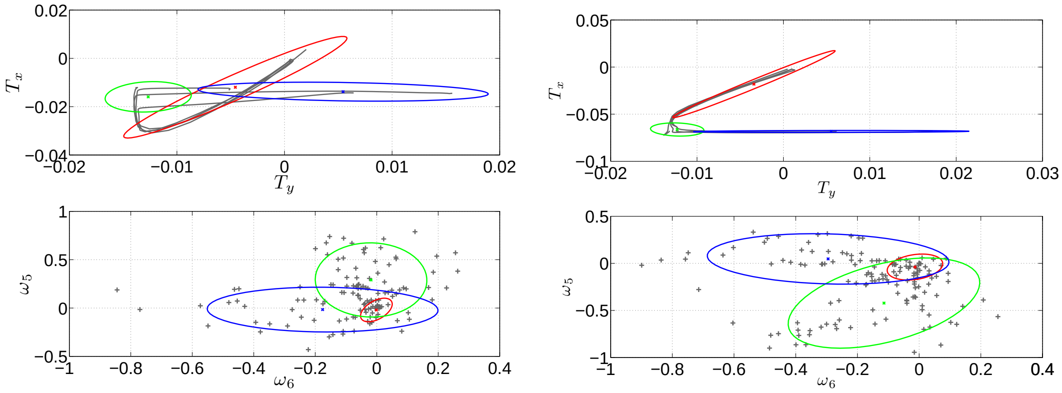
4.3.3 Ball-in-box task

In this section the encoding and reproduction results of the proposed framework are analyzed by means of a simple force-based task, previously described in Section 1.4.1.

4.3.3.1 HMM encoding

For encoding this task, inputs are the force-torque sensed at the robotic wrist and outputs are the velocity commands ω_l at each robot joint q_l with $l = 1, \ldots, N_q$. Therefore, in general terms the goal is to approximate the policy $\mathbf{\Pi}$ that maps force perceptions to robot joint velocities. Note that joint velocities were chosen as outputs because they do represent the robot actions to be performed according to the force-torque perceptions. As explained in Section 3.4.1, the original set of inputs was subjected to a MI-based analysis, from which the subset $\mathbf{\Omega}$ of input variables containing the most relevant information about the task outputs was found. Thus, in this task each training datapoint is defined as $\boldsymbol{\delta}_t = \{T_x \; T_y \; F_x \; \omega_1 \; \ldots \; \omega_{N_q}\}$.

In other words, λ is encoding the joint distribution $P(\mathbf{\Omega}, \boldsymbol{\omega})$. To understand better this idea and how the model works, observe Figure 4.1 showing the HMM convergence for two different datasets: Figure 4.1(a) displays a three-components HMM trained with similar demonstrations starting from positions {1, 2, 3, 4}, while Figure 4.1(b) shows another model trained with samples starting from positions {7, 8, 9, 10}. Note how the hidden *left-to-right* structure is obtained after convergence (having an ergodic HMM at the beginning), which is the appropriate topology for learning these datasets separately. For both cases, the resulting vector $\boldsymbol{\pi}$ gives as initial state the blue Gaussian, that corresponds to the first movement carried out by the teacher (i.e., when the user orients the robot in such a way that the ball rolls towards the wall adjacent to the hole). In Figure 4.1(a), blue and red states intersect each other in input space, covering the same segments of trajectories. In this case, the temporal information is essential to



(a) Three-components HMM trained with positions {1,2,3,4}

(b) Three-components HMM trained with positions {7,8,9,10}

Figure 4.1: Resulting HMM for two different training datasets of the *ball-in-box* task.

determine what velocity command has to be provided, i.e., to disambiguate the states corresponding to inputs lying in the intersection area, which cannot be achieved using a GMM-based approach. This is an example of a multivalued case for which the proposed framework works properly.

To obtain a complete model, the teacher carried out four demonstrations for ten different initial ball positions placed along the box edges. Every demonstration was executed by teleoperating the robotic arm through the 6-DoF haptic device (as shown in Figure 1.4) and following the motion strategy explained in Section 1.4.1. The resulting training dataset consisted of all datapoints $\{\boldsymbol{\delta}_t\}_{t=1}^{T}$, which were used to train several HMMs by applying the *Baum-Welch* method until convergence. To find the "best" model, the BIC was used, which allows to find a trade-off between optimizing the model's fitting and the number of states [26]. Note that the selected HMM will be a model that can fit the data well, with no overfitting in BIC sense. Figure 4.2 displays the different BIC values for the set of models tested, and Figure 4.3 shows the selected five-components HMM along with a graphical representation of the resulting transition probabilities matrix.

The execution and generalization capabilities were tested for some of these models using query data extracted from the demonstrations and real experiments. The two-components HMM showed the worst performance, this model was not able to carry out the task starting at any place, even if it did it from a pre-trained initial position. The

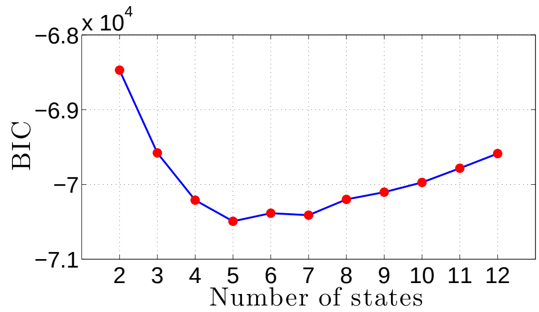


Figure 4.2: BIC values for models encoding the *ball-in-box* task with different number of states.

HMMs with four, eight and nine states could achieve the goal from every pre-trained positions but sometimes failed starting at non-trained initial configurations, showing poor generalization capabilities. Finally, the models with five, six and seven states showed very similar performances with no clear differences, and all of them performed the task successfully.

Observing the selected HMM with five components, it is interesting to highlight how the proposed framework is able to learn a multiple solution task by taking advantage of the HMM properties. The model is shown in Figure 4.3, where the blue state in the input space covers the beginning of all demonstrations whose initial positions are placed on the wall opposite to where the hole is. At these starting positions, a larger velocity command is required to draw the ball out of its resting configuration by moving the robot joint q_6 (Figure 4.3, output space projection). After, the green and light-blue states represent the movements to force the ball to role to the hole, through q_5 and depending on whether the ball is up or down with respect to the hole (i.e., positive or negative velocity commands, respectively). The yellow Gaussian can be considered as an intermediate state the system goes through to reach the final state (red ellipse) at which the velocity commands are zero (i.e., when the ball is getting out of the box) in input space.

4.3.3.2 GMRa reproduction

As for the reproduction phase, one teacher's demonstration for each initial position was removed from the training examples and used as "query data" for evaluating

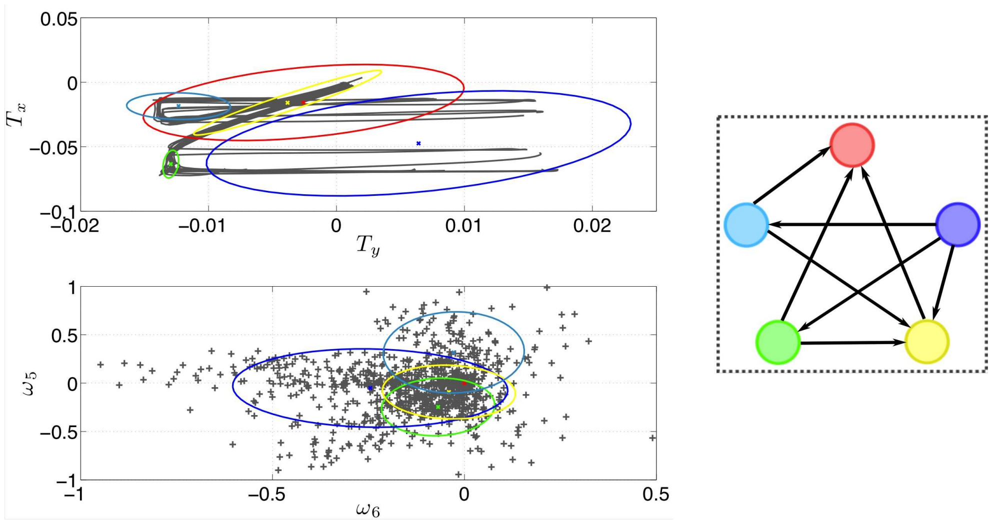


Figure 4.3: Resulting 5-states HMM trained with demonstrations starting at every position inside the box.

the learning framework performance by comparing its results with the teacher executions. All robot joint trajectories obtained from velocity commands synthesized by the HMM/GMRa approach are smoother than the teacher's demonstrations (as shown in Figures 4.4 and 4.5). Figure 4.4 shows the robot joint trajectories and velocities obtained while the teacher demonstrates how to take the ball out of the box (solid blue line), when starting at position 7. The trajectories and velocity profiles of the robot in the execution phase are also displayed (dashed red line). These predictions have been computed via GMRa for the inputs displayed in the first row of the figure and using the HMM displayed in Figure 4.3. It can be observed that the learning framework is able to compute the correct velocity commands to follow the teacher's strategy as well as to accomplish the task's goal. In addition, every joint trajectory is very similar to the *desired* one, even for those robot joints that are not playing a relevant role in the task (e.g., q_1 or q_3).

By observing the obtained velocity profiles for each robot joint, one sees that they are also smoother than the ones performed by the teacher, because human user executions show several abrupt changes, which are not over-fitted by the learning framework. This can be attributed to the selection process of the number of components of the HMM – avoiding overfitting – and also to the fact of using GMRa to retrieve the velocity command, because this type of regression takes the covariance information

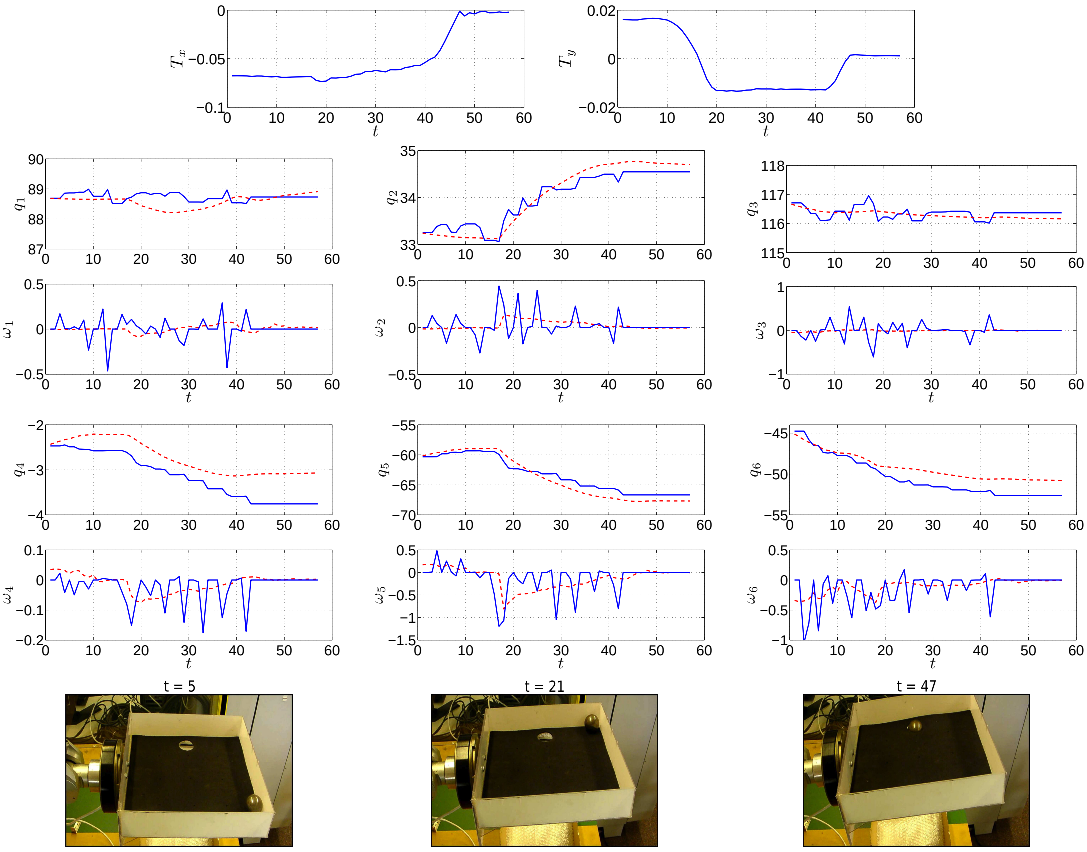


Figure 4.4: Input and output data streams over the course of a reproduction of the *ball-in-box* task starting from the position number 7. The human demonstration is displayed by the blue solid line, while the robot execution corresponds to the red dashed line.

into account for computing the estimation of the output, outperforming techniques that only use the mean of the Gaussians. From these results, it is possible to conclude that the robot performs better than the teacher. In addition, all synthesized trajectories follow the same motion pattern as that of the teacher's executions, which indicates that the strategy applied by the human user was learned successfully.

Once computational results were satisfactory, the framework was validated on the experimental setup. First, the robot had to perform the task with the ball starting at the already trained initial positions. In all experiments, the robot was able to carry out the task effectively. After this, a second set of tests was executed, where the ball was located at random positions inside the container. For these tests, the robot was also able to achieve the task's goal, executing the motions learned for the closest initial position,

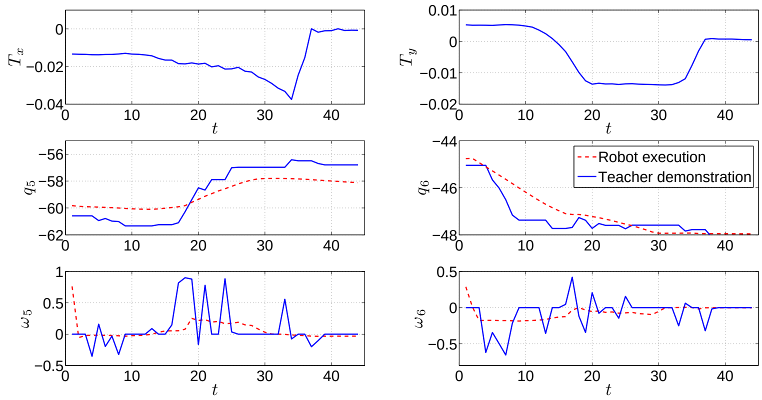


Figure 4.5: Input and output data streams over the course of a reproduction of the *ball-in-box* task starting from the position number 3.

by identifying the corresponding HMM state. It was observed that in some executions the ball reached and surpassed the hole, without falling through it, because the hole is only slightly larger than the ball. This behavior may be justified by the fact that the task is assumed to be "quasi-static".[1] However, the robot was always able to take the ball out of the box after some more executions, as it correctly identified the HMM state corresponding to the current and past input patterns (taking into account the temporal information). This means that the robot generates its actions as a function of its current and past perceptions, following the taught motion strategy. If the robot fails to reach the goal, the ball goes to another position inside the box, providing new perceptions from which the robot can compute new movements.

4.3.3.3 Evaluating the robot performance

As the robot was able to accomplish the desired goal in every test, even when the ball reached and surpassed the hole, the performance of the robot executions was evaluated by using a time-based criterion [151]. Here, the idea is to determine how much time the robot takes to complete the task successfully by executing the commands obtained from the proposed framework compared with the three following cases: *(i)* the robot executes hand-coded actions according to pre-programmed *if-then* rules, *(ii)* the teacher

[1]Note that the model variables are force-torque and joint velocities at the given time step, thus no information about the past is explicitly provided. Moreover, the robot controller only allows position-based control, thus it is not possible to send the desired velocity commands directly.

carries out the task by teleoperation following the mentioned strategy, *(iii)* the robot performs random movements that may take the ball towards the hole. Figure 4.6 shows execution times for the aforementioned cases. As expected, the teacher's executions show the lowest times, except for position number 2 where the robot was faster than the human. A relevant aspect to discuss is the fact that the robot execution times are much larger than the teacher's ones for positions 3 to 8. Regarding positions 3 to 5, higher times are due to the fact that the robot starts the task by moving the joint q_6 as expected, however it also moves q_5 slightly which sometimes causes the ball to go to the bottom of the box, justifying higher standard deviations for positions 3 and 4. This is a normal effect because the first state of the learned HMM covers non-zero angular velocities for the variable ω_5. Thus, in these cases, the robot identifies the new state where the ball is and changes its motion strategy according to the given input data for reaching the target.

In the case of positions 6 to 8, the robot does also move the joint q_5, however it is because the teacher demonstrations showed that the human tries to guarantee "a stable motion" by taking the ball towards the wall adjacent to the hole along the wall at the bottom of the box. This causes that, when the ball reaches the wall adjacent to the hole, the robot has to carry out more movements in order to take the metalic sphere towards the hole, since the robot must compensate the initial inclination of the box given by the wrong motion of q_5. Thus, the high robot execution times are mainly a consequence of two factors: first, there is a delay between the sensing and execution phases that increases the time measures as the ball is farther from the target, and second, the joint velocity profiles of the robot execution show lesser magnitudes than the teacher ones (as observed in Figures 4.4 and 4.5), implying that when these velocity commands are translated into desired positional configurations of the robot, the joints rotation is lower and more velocity commands are needed to orient the box.

Regarding the times shown for the hand-coded actions, several robot learned executions outperformed the hand-coded ones (e.g., starting at positions 1, 2, 4, 7, 9 and 10). This mainly happened because the hand-coded actions also suffered the "surpassing" effect, that is, the ball did not go out through the hole at the first attempt. Moreover, it is important to emphasize that the *if-then* rules programming was tedious and time-consuming, even for this simple task. On the one hand, it was essential to

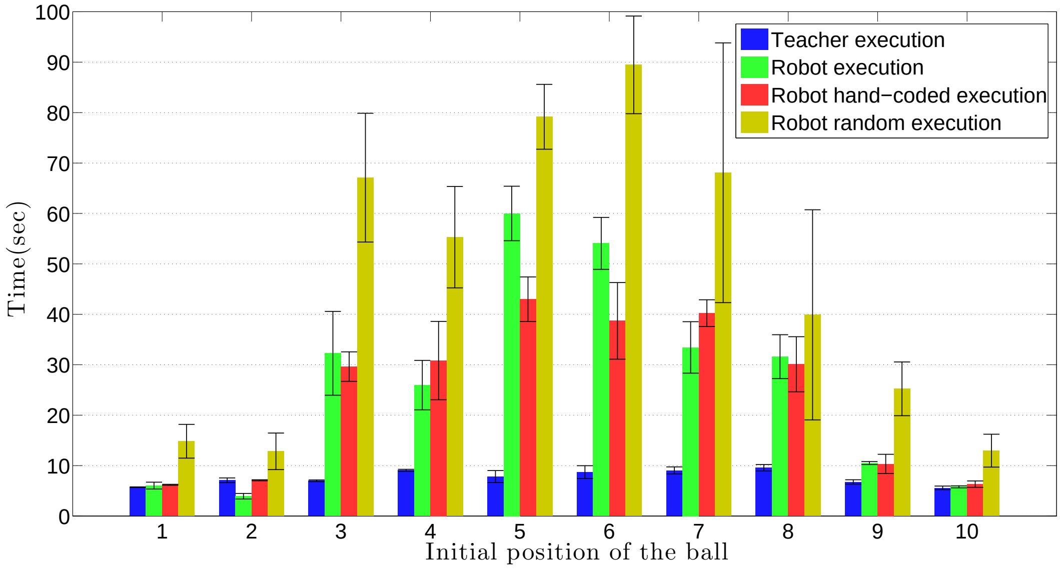


Figure 4.6: Time-based evaluation of the robot performance in the *ball-in-box* task.

determine how the input space could be transformed to discrete regions to set the *if* conditions. On the other hand, a tuning process was needed to specify the velocity commands that the robot executed. One may think that the higher the velocity, the less time the robot might take to accomplish the task, however the "surpassing" effect may occur more often, increasing the time execution significantly. Thus, the learning-based approach is preferred because being similarly efficient, it is friendlier and can be applied by non-expert users.

Finally, execution times for a "random" strategy show that trying to accomplish the goal by chance is possible, nevertheless this implies much higher times and variances in comparison with when the robot carried out the task by using the taught strategy. These high values occur because the random strategy does not impose movement constraints to the robot and, therefore, a huge set of available motions can be executed, leading to very varied and long trials. This constitutes a reference (lower bound) for comparison purposes, against which the improvement attained by different learning techniques and teaching strategies can be evaluated.

4.4 Exploiting force-based parameters

As previously shown, HMM is a suitable probabilistic tool for representing force-based tasks while implicitly encapsulating temporal information that can be exploited in order to obtain robust reproductions, even for tasks with a underlying multivalued function behavior. However, there is an open issue in the former experiment that is also a common feature of several force-based skills. One may picture a situation where the teacher wants the robot to carry out the same task with a different ball – bigger or smaller – whose weight significantly differs from the ball's one used in the demonstration phase. In such case, the skill remains unchanged (i.e., the robot actions) while the perceptions do not because the sensed force-torque would vary in magnitude. Here, the mass of the sphere may be considered as a task parameter, and such information may be exploited to deal with this kind of circumstances, instead of repeating the whole teaching process for each new different ball.

Note that the classic HMM does not handle task parameters explicitly, and if a parameter exists for every demonstration, a possible solution would be to add it as an additional (constant) input variable into the observation vector with the cost of increasing the dataspace dimensionality. Instead, it is possible to resort to a parametric version of the HMM, namely PHMM [164]. This technique models the dependence on the parameter of interest in an explicit way through the output densities b_j. Such model allows to obtain a more generic learning framework (see Section 1.3 and Figure 1.2), providing the possibility of encoding and reproducing a larger range of tasks.

4.4.1 Parametric hidden Markov models

Formally, the observation probability distributions of the classical HMM are now a function of the training datapoint and an associated parameter $\boldsymbol{\theta}_m$ of the demonstration m: $b_j(\boldsymbol{\delta}_t; \boldsymbol{\theta}_m)$. The dimension of the parameter depends on its degrees of freedom, for instance $\boldsymbol{\theta}$ would be a three dimensional vector if representing the location of an object in the space. Here, the linear dependence of the mean of the Gaussian distributions on $\boldsymbol{\theta}$ is adopted, wherein the center of each component j of the model is expressed as follows:

$$\hat{\boldsymbol{\mu}}_j(\boldsymbol{\theta}_m) = \boldsymbol{W}_j \boldsymbol{\theta}_m + \boldsymbol{\mu}_j, \tag{4.12}$$

where $\boldsymbol{W}_j$ describes the linear variation. Equation 4.12 can be expressed in a matrix fashion as $\hat{\boldsymbol{\mu}}_j(\boldsymbol{\theta}_m) = \boldsymbol{Z}_j \boldsymbol{\Psi}_m$, where

$$\boldsymbol{Z}_j \equiv [\boldsymbol{W}_j \quad \boldsymbol{\mu}_j], \tag{4.13}$$

$$\boldsymbol{\Psi}_m \equiv [\boldsymbol{\theta}_m \quad 1]^\top. \tag{4.14}$$

The former linear formulation allows to estimate only one additional model parameter through EM, namely $\boldsymbol{Z}_j$, from which the means of the model are computed for a specific value of $\boldsymbol{\theta}_m$ as follows (the readers are referred to [164] for details):

$$\boldsymbol{Z}_j = \left[\sum_{m,t} \gamma_{mt}(j) \boldsymbol{\delta}_{mt} \boldsymbol{\Psi}_m^\top\right] \left[\sum_{m,t} \gamma_{mt}(j) \boldsymbol{\Psi}_m \boldsymbol{\Psi}_m^\top\right]^{-1}, \tag{4.15}$$

where the sub-index m refers to the demonstration with an associated parameter $\boldsymbol{\theta}_m$. Once the means are estimated, the covariance matrices $\boldsymbol{\Sigma}_j$ are updated as explained in Section 4.3.1. Again, Eq. (4.4) is used as stopping criterion for this EM process. Note that this parametric model provides a compact probabilistic encoding of the demonstrations, handling the task parameters through a linear dependence that modifies the location of the output densities in the dataspace, without influencing their shape (i.e., the covariance information).

4.4.2 Pouring task

In this section, the performance of the classical and parametric versions of the HMM jointly working with GMRa is analyzed in a more realistic task, where the teacher demonstrates to a robot how to pour drinks using force-based perceptions (details are given in Section 1.4.2). In this context, some drawbacks of the learning framework based on the HMM are evident. Nonetheless, it is also shown how these disadvantages can be overcome by using PHMMs [139]. This new task also allows to show the flexibility and versatility of the proposed learning framework, that can be therefore used for encoding and reproducing many everyday manipulation tasks.

4.4.2.1 HMM vs. PHMM encoding

In this experimental setting, the robot perceptions are the the sensed forces and torques in the robot's frame. However, as explained in Section 3.4.2, several of these variables do not play a relevant role for learning the skill at all, thus Mutual Information analysis was applied to the input variables in order to select the most relevant perceptions. The resulting "most informative" set of inputs was $\boldsymbol{\Omega} = \{T_x \;\; F_z \;\; T_y\}$, of which T_x and F_z may also be directly extracted from an analysis of the task at hand (more details in Section 3.4.2).[1] In contrast to the previous experiment, the input space is composed of the selected subset of variables $\boldsymbol{\Omega}$ and the current joint value $\boldsymbol{q}$ at time step t, while outputs are the desired robot state to be achieved at $t+1$. Thus, each training datapoint is defined as $\boldsymbol{\delta}_t = \{T_x^t \; F_z^t \; T_y^t \; q_1^t \; \ldots \; q_{N_q}^t \; q_1^{t+1} \; \ldots \; q_{N_q}^{t+1}\}$, where N_q is the number of joints of the robot.

This means that the state information $\boldsymbol{z}(t)$ of the policy $\boldsymbol{\Pi}$ now includes information about the internal proprioceptive state of the robot, exploiting the generic formulation of Equation 4.1. The fact of having included the robot state into the input vector is aimed at encapsulating the task dynamics, so that the next robot state depends not only on the force-based perceptions but also on the current robot joint values, which is in the line of a consistent representation of the skill [121, 143]. This differs from the representation used in the ball-in-box task, where the state of the task was expressed only by force information. Now, this extended state allows to disambiguate perceptual aliases more easily, making the proposed structure more robust (at cost of increasing the dataspace dimensionality), considering that this representation encapsulates low-level sequential information while the learning model extracting higher-level sequences (as explained in Section 4.3.2). Moreover, this shows the flexibility of the framework for dealing with different input/output mappings.

In order to teach the robot to pour drinks, three "complete executions" of the task (i.e., serving four drinks consecutively) are provided to the robot by teleoperation as described in Section 1.4.2. Each demonstration consists in starting with the bottle full of fluid and pouring four 100 ml drinks. Note that after each drink is poured, the initial

[1] A simple analysis should convince the reader that the bottle with the fluid inside produces a load on the sensor because of the gravitational force, generating a force along the axis z and a torque about x.

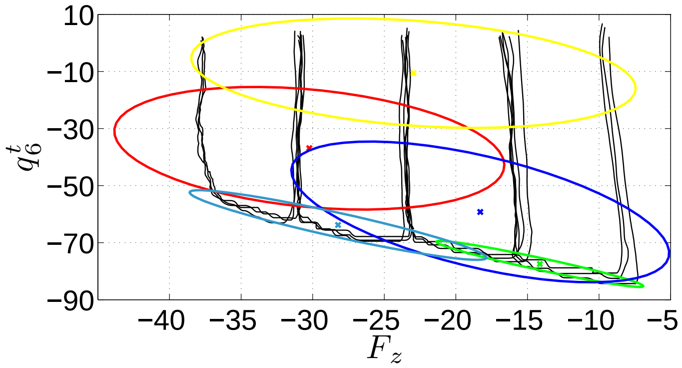


Figure 4.7: Resulting five-components HMM trained with three demonstrations of the pouring task.

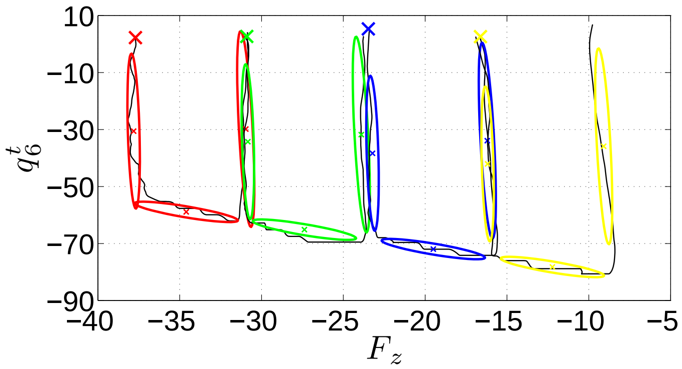
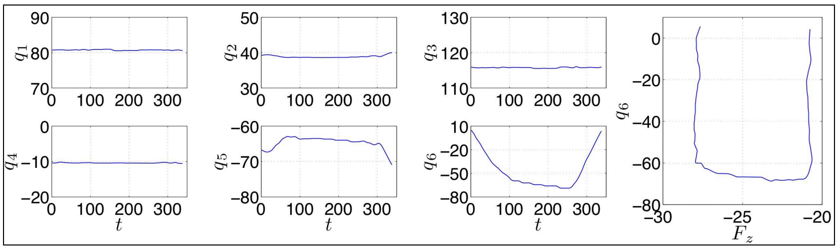


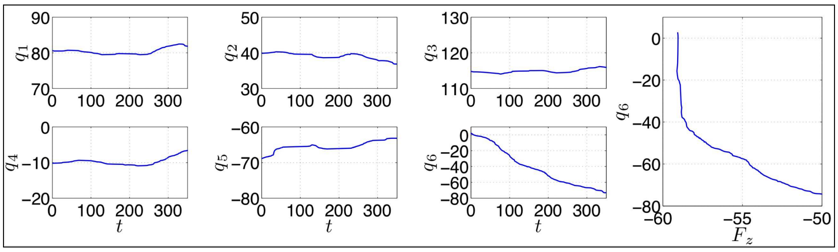
Figure 4.8: Resulting 3-components PHMM trained with one demonstration of the pouring task (four 100 ml drinks poured).

force-torque value changes for the next one, which conditions the robot movements as shown in Figure 4.7 where the black lines represent the teacher's demonstrations. Observe that the less quantity of fluid, the more the robot rotates the bottle. Using this training dataset, two different models were trained for comparison purposes, one of them corresponds to the classic HMM and the other one to the PHMM.

The resulting training dataset was used to train a five-components HMM by applying the *Baum-Welch* method until convergence. The number of Gaussians was chosen according to the BIC. Figure 4.7 shows the model encoding the pouring skill, where the yellow component covers the beginning and the end of all the executions, whereas the light blue and green ellipses are encapsulating the phases when the fluid is coming out of the bottle. The other two Gaussians can be considered as intermediate phases



(a)



(b)

Figure 4.9: Reproduction of the pouring task for two different quantities of fluid using the HMM.

of the task.

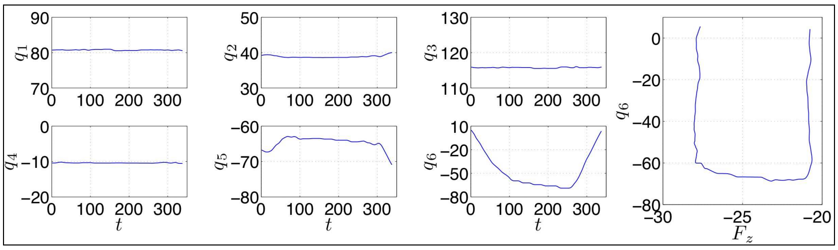
On the other hand, a PHMM of three Gaussians was trained using only one demonstration of the pouring skill, i.e., four examples of pouring 100 ml drinks. The parameter was set to be the initial force-torque values directly representing the quantity of fluid at the beginning of the demonstration m, that is $\boldsymbol{\theta}_m = [F_z^t \;\; T_x^t]^\top$ with $t = 1$. Figure 4.8 shows the resulting model, where it is possible to see how the PHMM is able to encode the task through a simple *left-to-right* topology. Note how the provided parameter translates the model components to cover the corresponding demonstration data due to the linear relationship between the task parameter and the Gaussian means (see Equation 4.12). It should be noted that neither the Gaussians shape nor their orientation are modified. In this task – and those also depending on parameters – it may be useful to shape the Gaussians to improve the reproductions. For instance, the first component of the model may be explicitly shaped according to how much the robot rotates the bottle for a given force parameter. This issue will be discussed at the end of this chapter.

4.4.2.2 GMR reproduction

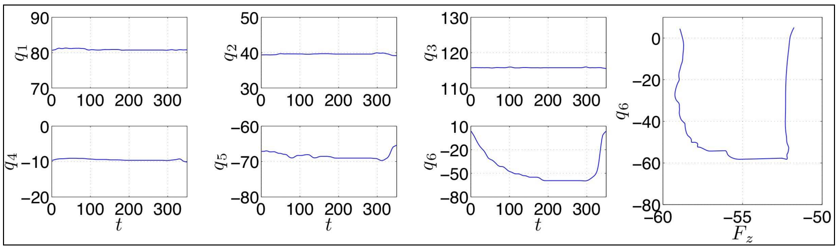
In order to test the reproduction performance of the models, one demonstration (serving four 100 ml drinks) was removed from the training data to be used as query datapoints. All the joint trajectories executed by the robot were quite similar to the ones obtained from the teacher examples as well as the input-output pattern, using both HMM and PHMM. After this, the following tests were aimed at evaluating the generalization capabilities of both models. In this case, the bottle contained quantities of fluid different from the ones used at the demonstration phase. Regarding the HMM, the robot performed successfully for all the tests where the starting force-torque perception was covered by the initial component of the model (i.e., the yellow ellipse in Figure 4.7). The robot joint trajectories and the input-output pattern for one of these tests is shown in Figure 4.9(a). Nevertheless, as the starting perception significantly differs from the values encapsulated by the initial component, the robot performance deteriorates considerably (see Figure 4.9(b)). In other words, the actual model shows good interpolation competences but a poor extrapolation performance, which constrains the range of situations where the robot would perform successfully without retraining the HMM.

Regarding the PHMM performance, again different quantities of fluid from the ones used in the demonstrations were given to the robot, using the same bottle and a larger one. For both cases, the robot successfully poured a drink of 100 ml approximately (see Figure 4.10). Figure 4.10(b) displays a reproduction of the robot when a larger bottle full of fluid was used, where it is possible to observe how the robot joint q_6 follows a trajectory quite similar to the demonstrated ones, performing the skill as expected. This result shows that the robot is able to pour drinks using bottles of similar shapes but different sizes without retraining the model, because the force-based parameter allows to displace the model so that it can cover the subspace where the data are expected to be, which is not possible using the classic HMM (see Figure 4.9(b)). Thus, the PHMM provides better generalization capabilities than those observed using the HMM, with fewer components and a simpler topology.

Figure 4.11 displays the Gaussian distribution and the influence of the model components for the reproduction shown in Figure 4.10(b). On the one hand, one can observe how the model translates to cover a different data subspace given the new task



(a)



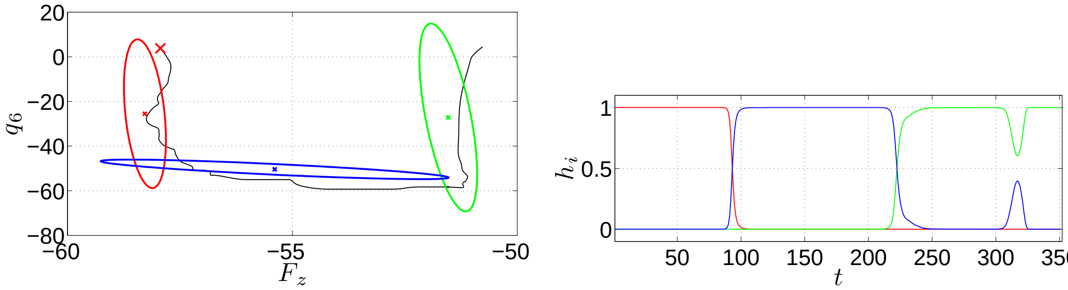
(b)

Figure 4.10: Reproduction of the pouring task for two different quantities of fluid using the PHMM.

parameter. On the other hand, it is also possible to see how the GMRa weights $\alpha(i)$ (Equation 4.11) evolve over time, showing how the model components influence the reproduction.

4.5 Chapter highlights

Several learning techniques were analyzed in the context of force-based tasks, where the limitations of methods like LWPR and the framework GMM/GMR were evidenced when representing skills characterized by underlying multivalued functions. Thus, the proposed solution considered a learning framework based on HMM and GMRa that is able to encode and reproduce force-based skills successfully. The framework performs efficiently when the teacher's demonstrations exhibit a multivalued function behavior, which was achieved by means of the GMRa using temporal information encapsulated by the HMM without explicitly considering time as another input variable and avoiding to deal with very large time discrepancies. Time, or rather sequential information, is already implicitly present along the teacher's demonstrations. Afterwards, an extension of this work was proposed to handle scenarios where the robot actions also depend on task parameters. In this case, a structure based on PHMM and GMRa allows



(a) Resulting PHMM for the given unobserved force perception

(b) PHMM component activation over the course of the skill

Figure 4.11: PHMM for unobserved data and the corresponding component influence profiles during the reproduction phase.

to compactly represent such kind of behaviors without merely including these data into the observation vector. Computational and experimental results of two different manipulation tasks validated the appropriateness of the proposed approaches.

Nonetheless, several issues need still to be addressed. One of them is related to the fact that the covariance of the PHMM components is not affected at all by the task parameter, which may be problematic when the local variability of data depends on it. On the other hand, the experimental scenarios studied in this chapter did not consider any additional perception, and the way in which the teacher carried out the demonstrations clearly suggests that manipulation tasks are not exclusively based on force data, and therefore vision information should be included. These two issues are addressed in the next chapter in the context of a human-robot collaborative assembly task, where a probabilistic model that parameterizes both the mean and covariance of the Gaussian components is introduced.

Chapter 5

Merging visual and haptic information in collaborative manipulation tasks

While executing manipulation tasks, humans commonly look first at the scenario, select an object, approach their hands, grasp the object and displace it by lifting, pushing or pulling it. Further manipulative actions may include insertion, if the object is a component of an assembly, filling or emptying it if it is a container, exerting forces on other objects if it is a tool, etc. In this process, vision, tactile and force data are generated, which provide the human with information for estimating the state and properties of the object and the surroundings. Humans resort simultaneously to various perception sources in a natural way while performing their jobs. However, research on LfD has mostly focused on using only one input channel to gather information about the task, which shows to be enough for very specific skills and environments. It should be noted that vision-based systems are the most frequently used hardware in LfD settings, which despite its richness and advantages might also miss specific valuable information when the robot manipulates objects or interacts with humans (as discussed in Section 2.2). Thus, if required by the task, or if it means a significant enhancement of the task learning process, other types of information sources like touch, force and sound sensing should be considered as well.

A decade ago, it was shown how humans integrate visual and haptic information in a statistically optimal fashion when manipulating objects [47]. This concept was further studied in a visual-haptic virtual reality setup for estimating the compliance of a given material [95]. It was observed that an integration process takes place through a weighted summation of two random variables, which are defined by single modality estimates (i.e., Gaussian distributions of the vision-haptic data). Such results make evident that force-based data enrich vision systems and provide additional useful information. Just a few works in progress have recently shown how vision information can be merged with force data in robot learning scenarios. Prats *et al.* proposed a position-vision-tactile hybrid control modified by an impedance force controller to carry out a door opening task [126]. Falco *et al.* used force measurements to improve the observation of the human hand motion [51], which may be of particular interest in LfD applications.

As observed, vision and force data merging is a promising idea to enhance the robot perception system when interacting with physical entities (e.g., objects or human partners), and it is fundamental to take into consideration the type of data that each input channel may provide. On the one hand, vision-based systems often take care of capturing the pose of an object of particular interest in the task. On the other hand, force-based data generally convey information about contact with an object, and also static and dynamic characteristics (e.g., the weight of an object or the inertial components of force during dynamic manipulation). Both perceptions are evidently needed in robotic manipulation tasks. This chapter deals with merging vision and force in LfD, where instead of simply augmenting the observation vector, the inputs provided by the vision system are considered as **task variables** while the force data compose the input space as usual, taking advantage of the task features (see Section 4.3 and 4.4). This approach avoids to increase the dataspace dimensionality (and probably the computational complexity), and it also provides a compact structure for dealing with tasks where several objects are manipulated.

It is worth mentioning that in contrast to PHMM, the key difference here lies on the task-parameterized encoding, where both the means and covariances of a GMM are modified over time as a function of the vision-based parameters, namely the pose of an object of interest (see Section 5.1). The approach is applied to learning reactive impedance-based behaviors of a robot collaborating with a human in an assembly task (described in Section 1.4.3). This scenario is suitable to see how vision and force perceptions can be jointly exploited in LfD, and how task variables are used in a probabilistic model to obtain a compact encoding of the demonstrations provided by the teacher. Moreover, from this model, a set of virtual springs is proposed to govern the robot behavior at the execution stage, as explained in Section 5.2. The robot impedance is shaped by estimating stiffness matrices from the set of demonstrations.

5.1 Vision-based task parametrization

When robots manipulate objects, their actions may largely depend on the given goals and object poses, which can be defined through reference frames. Here, the robot's behavior is conditioned by a set of task variables representing the coordinate systems of relevant frames of reference. For generalization purposes, it is desirable to have a model enclosing different actions as a function of these variables instead of representing each action with a different model. An example of such approach is the parametric hidden Markov model (**PHMM**), previously used to encode a task with force-based variables conditioning the robot actions (see Section 4.4). However, this approach does not allow to parameterize the covariance term of the Gaussians, which is crucial because covariance encodes the local relationships among the variables that are of interest for the task, as well as the expected variations during its execution. With standard PHMM, a common practice to circumvent this problem is to deliberately increase the number of Gaussian components when encoding a continuous trajectory (typically, by considering more components for a synthesis problem [104] than for a recognition problem [100]). The resulting effect is to reduce the importance of the covariance term in the modeling, at the expense of increasing the required number of states and of losing information about the local spread of data.

Other works have exploited the use of several candidate frames relevant to the task for learning robot actions. Cederborg *et al.* [31] used the notion of *framings* to incrementally learn various tasks that can be defined in different frames, where the system infers from demonstrations which particular frame should be used for each task. Calinon *et al.* [28] proposed to learn an HMM in several position-varying landmark frames, and then to estimate a resulting model in the form of products of Gaussians. By taking the aforementioned strategies into consideration, a frames-based approach that considers task parameterization on both the centers and the covariance matrices of the Gaussian distributions over time is used, that is known as parametric Gaussian mixture model (**PGMM**). The advantages of this approach compared to other task-parameterized models in the context of trajectory learning are discussed in [30].

Previous work proposed to tackle the problem of parameterizing both the centers and covariances by encoding each demonstration in a separate model [65]. After proper realignment, a resulting model is estimated as a weighted sum of the centers and covariances of the different models. One drawback of this approach is that it only allows scalar transformations of the covariance instead of generic linear transformations. It does not allow, for example, to re-orient a normal distribution with respect to objects in the robot's environment. Thus, this approach may not be applied to local stiffness estimation processes, where the main axes of the model components may change over time yielding stiffness matrices whose highest values change accordingly.

Brand and Hertzmann [15] dealt also with the problem of center and covariance parameterization through a stylistic HMM, in which the model parameters depend on a style variable. Their solution – applied to motion data analysis for computer graphics – extracts motion patterns from a highly varied set of motion capture sequences, which are then related to stylistic parameters. These style variables along with variation matrices modulate the model Gaussians. A notable difference is that the model parameters learning is carried out by entropy minimization.

5.1.1 Parametric Gaussian mixture model

As an alternative approach, PGMM is proposed here to learn a skill where the probabilistic model is locally influenced by a set of variables describing the task situation. Instead of defining the centers of the Gaussians directly as a linear relationship with respect to the task variables, like in the standard PHMM [164], products of Gaussians are used to combine a set of linear relationships. This approach takes inspiration from the *product of experts*, where each expert (or probabilistic component such as a Gaussian) represents a soft constraint of the problem. For an event to be likely under a product model, all constraints must be (approximately) satisfied, in contrast to how a *mixture of experts* works [67], where all constraints are non-linearly combined. In other words, the PGMM trains a different sub-model for each candidate frame (i.e., the task variables) and obtains a single model from the resulting Gaussian product of all the sub-models encoding the problem constraints locally in each frame.

The result is a simple model that can parameterize not only the centers of the Gaussians but also the covariance matrices (this computation is helpful to consider simultaneously center and covariance parameterization). In this thesis, PGMM is exploited to encode impedance-based behaviors in a LfD scenario where haptic inputs along with vision-based task variables determine how the robot should behave to accomplish the task successfully [137].

Formally, each demonstration $m \in \{1, \ldots, M\}$ contains T_m datapoints forming a dataset of N datapoints $\{\boldsymbol{\xi}_n\}_{n=1}^{N}$ with $N = \sum_{m=1}^{M} T_m$. Each $\boldsymbol{\xi}_n \in \mathbb{R}^D$ is associated with task variables $\{\boldsymbol{A}_{n,j}, \boldsymbol{b}_{n,j}\}_{j=1}^{N_P}$ representing N_P candidate frames of reference, with transformation matrices $\boldsymbol{A}_{n,j}$, and offset position vectors $\boldsymbol{b}_{n,j}$. D is the training data dimensionality, and the indexes n and j represent the time step and the task variable, respectively.

The **parameters of the model** are $\{\pi_i, \boldsymbol{Z}^{\mu}_{i,j}, \boldsymbol{Z}^{\Sigma}_{i,j}\}$, representing respectively the mixing coefficients, centers and covariances matrices for each frame j and mixture component i. With this model, for an observation of frames at iteration n, the resulting center $\boldsymbol{\mu}_{n,i}$ and covariance matrix $\boldsymbol{\Sigma}_{n,i}$ of each component i are computed as products of linearly transformed Gaussians

$$\mathcal{N}(\boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i}) = \prod_{j=1}^{N_P} \mathcal{N}\Big(\boldsymbol{A}_{n,j} \boldsymbol{Z}^{\mu}_{i,j} + \boldsymbol{b}_{n,j},\ \boldsymbol{A}_{n,j} \boldsymbol{Z}^{\Sigma}_{i,j} \boldsymbol{A}^{\top}_{n,j}\Big).$$

By using the product property of normal distributions, the above equation is computed as

$$\begin{aligned} \boldsymbol{\mu}_{n,i} &= \boldsymbol{\Sigma}_{n,i} \sum_{j=1}^{N_P} (\boldsymbol{A}_{n,j} \boldsymbol{Z}^{\Sigma}_{i,j} \boldsymbol{A}^{\top}_{n,j})^{-1} (\boldsymbol{A}_{n,j} \boldsymbol{Z}^{\mu}_{i,j} + \boldsymbol{b}_{n,j}), \\ \boldsymbol{\Sigma}_{n,i} &= \Big(\sum_{j=1}^{N_P} (\boldsymbol{A}_{n,j} \boldsymbol{Z}^{\Sigma}_{i,j} \boldsymbol{A}^{\top}_{n,j})^{-1} \Big)^{-1}. \end{aligned} \tag{5.1}$$

The parameters of the model are iteratively estimated with the following EM procedure. In the E-step, (5.1) are used as temporary Gaussian parameters to compute the likelihood.

E-step:

$$h_{n,i} = \frac{\pi_i \mathcal{N}(\boldsymbol{\xi}_n | \boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i})}{\sum_k^{N_K} \pi_k \mathcal{N}(\boldsymbol{\xi}_n | \boldsymbol{\mu}_{n,k}, \boldsymbol{\Sigma}_{n,k})}. \tag{5.2}$$

M-step:

$$\pi_i = \frac{\sum\limits_n h_{n,i}}{N}, \quad \boldsymbol{Z}^{\mu}_{i,j} = \frac{\sum\limits_n h_{n,i}\ \boldsymbol{A}^{-1}_{n,j}[\boldsymbol{\xi}_n - \boldsymbol{b}_{n,j}]}{\sum\limits_n h_{n,i}},$$

$$\boldsymbol{Z}^{\Sigma}_{i,j} = \frac{\sum\limits_n h_{n,i}\ \boldsymbol{A}^{-1}_{n,j}[\boldsymbol{\xi}_n - \tilde{\boldsymbol{\mu}}_{n,i,j}][\boldsymbol{\xi}_n - \tilde{\boldsymbol{\mu}}_{n,i,j}]^{\top} \boldsymbol{A}^{-\top}_{n,j}}{\sum\limits_n h_{n,i}},$$

$$\text{with} \quad \tilde{\boldsymbol{\mu}}_{n,i,j} = \boldsymbol{A}_{n,j} \boldsymbol{Z}^{\mu}_{i,j} + \boldsymbol{b}_{n,j}. \tag{5.3}$$

5.1.2 Vision-based parametrization of force-based tasks

Note that for force-based tasks, the datapoints, means and covariance matrices can be decomposed into their position and force components

$$\boldsymbol{\xi}_n = \begin{bmatrix} \boldsymbol{\xi}^{\boldsymbol{x}}_n \\ \boldsymbol{\xi}^{\boldsymbol{F}}_n \end{bmatrix}, \quad \boldsymbol{\mu}_{n,i} = \begin{bmatrix} \boldsymbol{\mu}^{\boldsymbol{x}}_{n,i} \\ \boldsymbol{\mu}^{\boldsymbol{F}}_{n,i} \end{bmatrix}, \quad \boldsymbol{\Sigma}_{n,i} = \begin{bmatrix} \boldsymbol{\Sigma}^{\boldsymbol{x}}_{n,i} & \boldsymbol{\Sigma}^{\boldsymbol{xF}}_{n,i} \\ \boldsymbol{\Sigma}^{\boldsymbol{Fx}}_{n,i} & \boldsymbol{\Sigma}^{\boldsymbol{F}}_{n,i} \end{bmatrix}.$$

The model parameters may be initialized with a *k-means* procedure, modified by following a similar task-parametrized structure. Model selection is compatible with the techniques employed in standard GMM (Bayesian information criterion, Dirichlet process, etc.).

Fig. 5.1 illustrates the approach with a simple example of an impedance-based behavior learning. *(a)* shows three demonstrations where the robot behaves compliantly while another object (i.e., the green triangle with its corresponding candidate frame) is far away from its end-effector, and becomes stiff when the object approaches it with a specific orientation (the black line is the robot's trajectory). *(b)* displays the two phases of the task, where the robot motion is driven by a set of virtual springs connected to the center of the model's Gaussians (two components in this case). The mean and covariance vary according to the task variables (i.e., the object and robot frames), and the influence of each model component (see Eq. (5.2)) determines how compliant the robot behaves. Dark ellipses and thick-line springs represent an activated Gaussian. The candidate frames are displayed in red color.

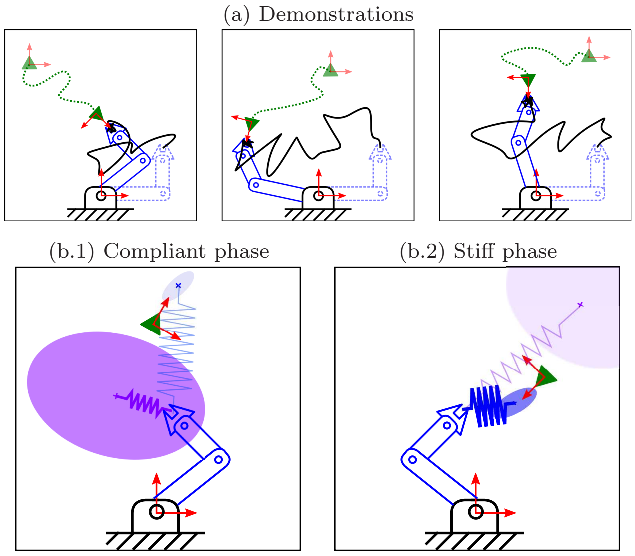


Figure 5.1: Simplified impedance behavior learning.

One novelty with respect to [30] is that here the model is augmented with virtual stiffness matrices $\boldsymbol{K}_i^P$ associated to each component i, which will be estimated as explained in Section 5.2. Thus, the complete set of parameters of the model is $\{\pi_i, \{\boldsymbol{Z}_{i,j}^{\Sigma}, \boldsymbol{Z}_{i,j}^{\mu}\}_{j=1}^{N_p}, \boldsymbol{K}_i^P\}_{i=1}^{N_K}$. Note that the variables of the task are obtained from the position and orientation of a set of candidate frames of reference to learn the task[1]; for instance in the collaborative task (Section 1.4.3), the table legs and robot frames define these variables.

5.2 Learning impedance-based behaviors

Impedance-based behaviors in Robotics can be understood as flexible regulations of stiffness and damping variables governing the robot actions in impedance control (see Section 2.6.2). This allows LfD to be applied in a new learning paradigm where the task is not merely to follow a given trajectory. Instead, it is possible to learn a new set of tasks from demonstrations by shaping the variables of an impedance-based model. Several approaches have been proposed to estimate from collected data the stiffness and damping variables to be used to control robots. Erickson *et al.* [46] compared

[1]Here, such frames are selected by the experimenter and consist of coordinate systems associated to a set of candidate objects or landmarks that might play a role in the task. However, we do not rule out that they could be chosen automatically.

four different methods to estimate the robot impedance based on signal processing, adaptive control and recursive least squares. Krislock *et al.* [85] proposed to use linear least squares to estimate a positive semi-definite matrix applied to the estimation of local compliance matrices during deformable object modeling. Flacco and De Luca [52] estimated the nonlinear stiffness of robot joints with flexible transmissions by using dynamic residual signals along with least-squares and regressor-based techniques.

From a different perspective, a LfD approach was proposed in [28] to find a stiffness matrix using variability information extracted from training data in the form of a GMM, where the stiffness matrix is estimated from the inverse of the observed covariance in the position space. Similarly, Lee and Ott [99] used variability encoded in the components of an HMM to define a motion refinement tube that permits a deviation from nominal trajectories for kinesthetic corrections by controlling the stiffness value at the robot joint level. In contrast, the approach proposed in this thesis considers that the robot end-effector behaves as a virtual mass connected through springs to a set of attractors (as shown in Figure 5.1),

$$\boldsymbol{F}_n = \sum_{i=1}^{N_K} h_{n,i} \left[\boldsymbol{K}_i^P \left(\boldsymbol{\mu}_{n,i}^{\boldsymbol{x}} - \boldsymbol{x}_n \right) \right], \tag{5.4}$$

where $\boldsymbol{F}_n$, $\boldsymbol{\mu}_{n,i}^{\boldsymbol{x}}$ and $\boldsymbol{x}_n$ are respectively the sensed force, the positional part of the Gaussians' centers in the model and the robot's end-effector position at time step n.

For obtaining an estimate of the stiffness values for this set of springs, an approximation based on an algebraic closed-form solution is computed to find the closest symmetric positive semi-definite stiffness matrix of a weighted least-squares estimation. Specifically, a weighted least squares (**WLS**) regression is used to compute a first estimate $\tilde{\boldsymbol{K}}_i^P = \left[(\boldsymbol{X}_i^\top \boldsymbol{W}_i \boldsymbol{X}_i)^{-1} \boldsymbol{X}_i^\top \boldsymbol{W}_i \boldsymbol{F} \right]$ of the stiffness matrices by concatenating all the N datapoints in matrices $\boldsymbol{X}_i = \left[(\boldsymbol{\mu}_{1,i}^{\boldsymbol{x}} - \boldsymbol{x}_1), \, \ldots, \, (\boldsymbol{\mu}_{N,i}^{\boldsymbol{x}} - \boldsymbol{x}_N) \right]^\top$ and $\boldsymbol{F}$, with a weighting matrix $\boldsymbol{W}_i = diag([h_{1,i}, h_{2,i}, \ldots, h_{N,i}])$ (see Eq. (5.2)). Such estimate does not necessarily comply with the constraints of a stiffness matrix, namely to be symmetric positive semi-definite. Therefore, the formulation presented in [66], and used in [33] to estimate the robot's dynamics variables for robot control, is implemented to to compute $\boldsymbol{K}_i^P$ as the nearest symmetric positive semi-definite (SPSD) matrix to $\tilde{\boldsymbol{K}}_i^P$ according to the Frobenius norm. From the estimated stiffness matrix $\tilde{\boldsymbol{K}}_i^P$, the nearest

Table 5.1: Learning and reproduction of impedance-based behaviors.

1. Task demonstrations
- Determine N_P (number of frames or task variables)
- $\forall n \in \{1, \ldots, N\}$
  - Collect $\boldsymbol{\xi}_n$ and task variables $\{\boldsymbol{A}_{n,j}, \boldsymbol{b}_{n,j}\}_{j=1}^{N_P}$

2. Model fitting
- Determine N_K (number of components of the model)
- Initialize $\{\pi_i, \{\boldsymbol{Z}^{\mu}_{i,j}, \boldsymbol{Z}^{\Sigma}_{i,j}\}_{j=1}^{N_P}\}_{i=1}^{N_K}$
- Use Eq. (5.3) to refine the model parameters with EM

3. Stiffness estimation
- Create matrices $\boldsymbol{X}$ and $\boldsymbol{W}_i$
- Compute the first estimate $\tilde{\boldsymbol{K}}_i^P$ through WLS
- Find $\boldsymbol{K}_i^P$ for each virtual spring by using Eq. (5.5).

4. Reproduction
- Collect $\boldsymbol{\xi}_n$ and $\{\boldsymbol{A}_{n,j}, \boldsymbol{b}_{n,j}\}_{j=1}^{N_P}$
- Estimate $\{\pi_i, \boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i}\}_{i=1}^{N_K}$ through Eq. (5.1)
- Compute weights (influence) $h_{n,i}$ using Eq. (5.2)
- Apply the force command computed from Eq. (5.4)

SPSD matrix $\boldsymbol{K}_i^P$ is computed as

$$\boldsymbol{K}_i^P = \frac{\boldsymbol{B} + \boldsymbol{H}}{2}, \tag{5.5}$$

$$\boldsymbol{B} = \frac{\tilde{\boldsymbol{K}}_i^P + (\tilde{\boldsymbol{K}}_i^P)^\top}{2} \quad \text{and} \quad \boldsymbol{H} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^\top.$$

$\boldsymbol{H}$ is the symmetric polar factor which can be found from the singular value decomposition of $\boldsymbol{B}$, namely, $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$. Table 5.1 summarizes the learning and estimation processes. In the next section, results on the model encoding and the estimation process are shown for a collaborative assembly where vision and haptic data are relevant to carry out the task successfully.

5.3 Experimental scenario: collaborative table assembly task

A collaborative task where a human and a robot carry out a table assembly is proposed to illustrate the functioning of the previously presented learning framework. As described in Section 1.4.3, the core idea is to teach the robot reactive impedance-based

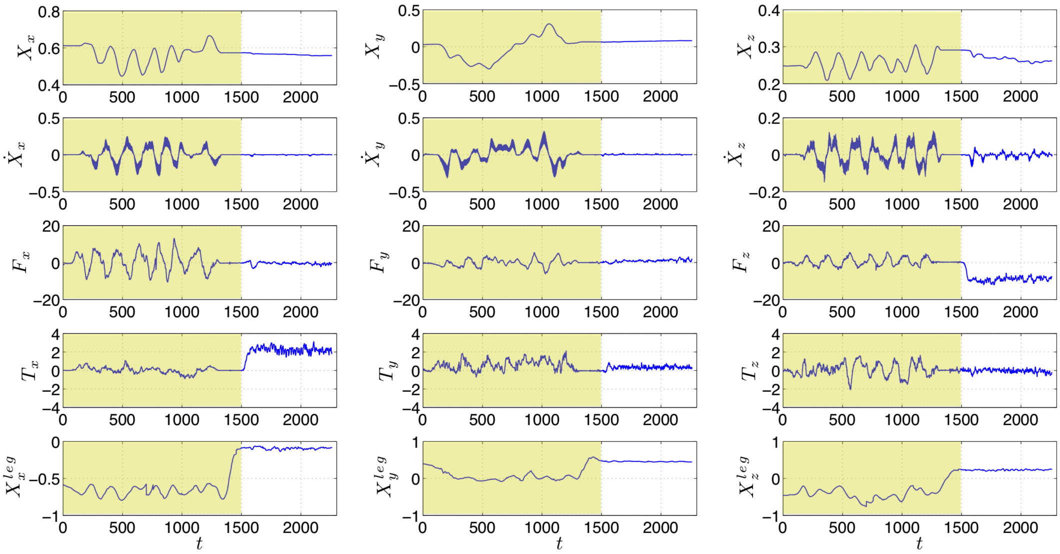


Figure 5.2: Data streams displaying the demonstration of the assembly of one leg.

behaviors. In this setting, vision-force merging is exploited. Vision captures the object configuration (i.e., the leg pose in the space), representing the task variables. Specifically, a transformation matrix is computed to represent the leg configuration in the fixed robot frame $\mathcal{O}_R$, from which $\boldsymbol{b}_n^{\text{leg}}$ and $\boldsymbol{A}_n^{\text{leg}}$ define the Cartesian position and the orientation of the leg as a rotation matrix, respectively. During both demonstration and reproduction phases, $\{\boldsymbol{A}_n^{\text{leg}}, \boldsymbol{b}_n^{\text{leg}}\}$ are recorded at each time step n to determine the task variables. Lastly, the other candidate frame $\{\boldsymbol{A}_n^{\text{R}}, \boldsymbol{b}_n^{\text{R}}\}$ defines the robot's fixed frame of reference.[1]

Forces and torques compose the observation vector of the problem, which provide information about when and where the leg is being assembled, as evidenced in Figure 5.2. The PGMM encapsulates both perceptions and the demonstrated robot actions in a compact model. Stiffness matrices are estimated using the provided demonstrations and the proposed set of virtual springs. The whole framework is then used at the reproduction stage, where force commands are computed using Eq. 5.4, and later sent to the Cartesian impedance controller defined by

$$\boldsymbol{\tau}_d = \boldsymbol{J}^{\top}\boldsymbol{F}_d + V(\kappa_d^V) + f(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}),$$

[1] A 3D coordinate frame is replicated for the variables $\boldsymbol{x}$ and $\boldsymbol{F}$, the offset is only set to $\boldsymbol{x}$.

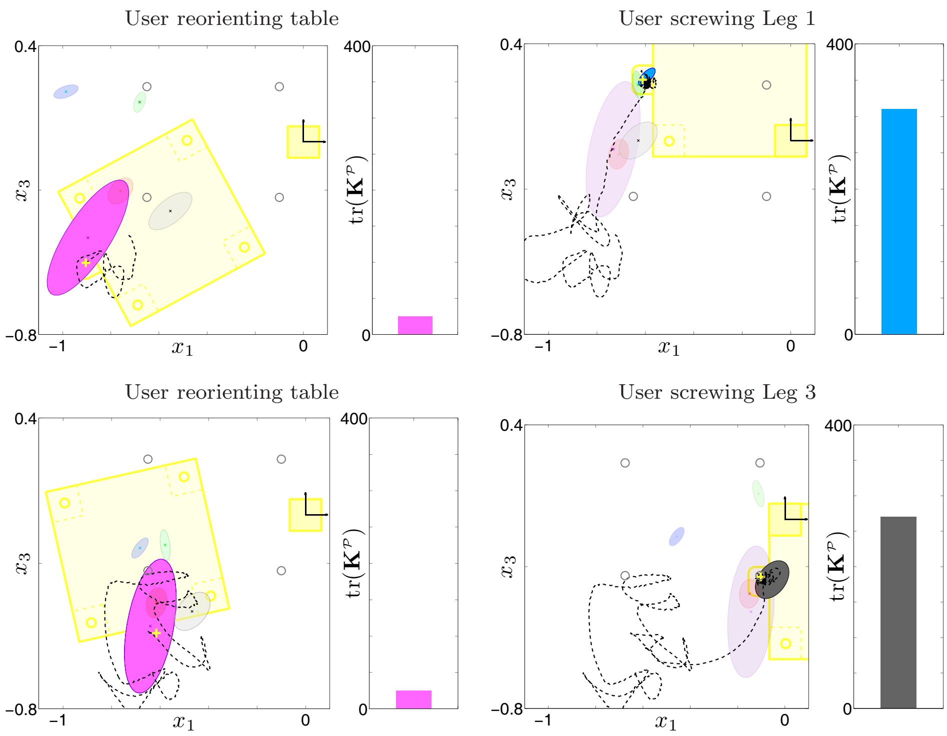


Figure 5.3: Reproduction results at different phases of the interaction

where $\boldsymbol{\tau}_d$ is the desired joint torques vector, $\boldsymbol{F}_d$ the desired force computed from the resulting set of virtual springs (Eq. 5.4), V is the damping function with desired damping values κ_d^V and $f(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ the dynamic model of the robot. [1]

5.3.1 Parametric encoding results

Two candidate frames of reference are considered in this task ($N_P = 2$): the fixed robot frame of reference $\mathcal{O}_R$ and the leg frame of reference $\mathcal{O}_L$. A model of five components ($N_K = 5$) was trained with sixteen demonstrations (i.e., four samples of the leg screwing process were provided for each of the legs). The model was initialized by a modified k-means that follows a similar strategy as the EM mechanism in Eq. (5.3). The resulting model encodes four stiff components corresponding to the four screwing phases while the remaining component represents the compliant behavior, automatically estimated by the model. Each "stiff component" is characterized by the force-torque pattern and

[1]Note that $\boldsymbol{F}_d$ is computed at each time step n given the current position of the robot's end-effector, the learned centers $\boldsymbol{\mu}_{n,i}^{\boldsymbol{x}}$ and the estimated $\boldsymbol{K}_i^P$.

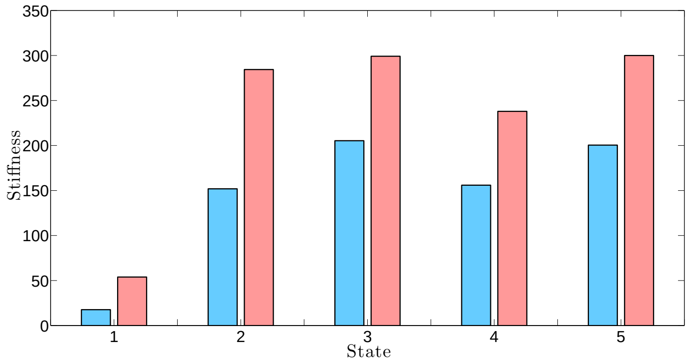
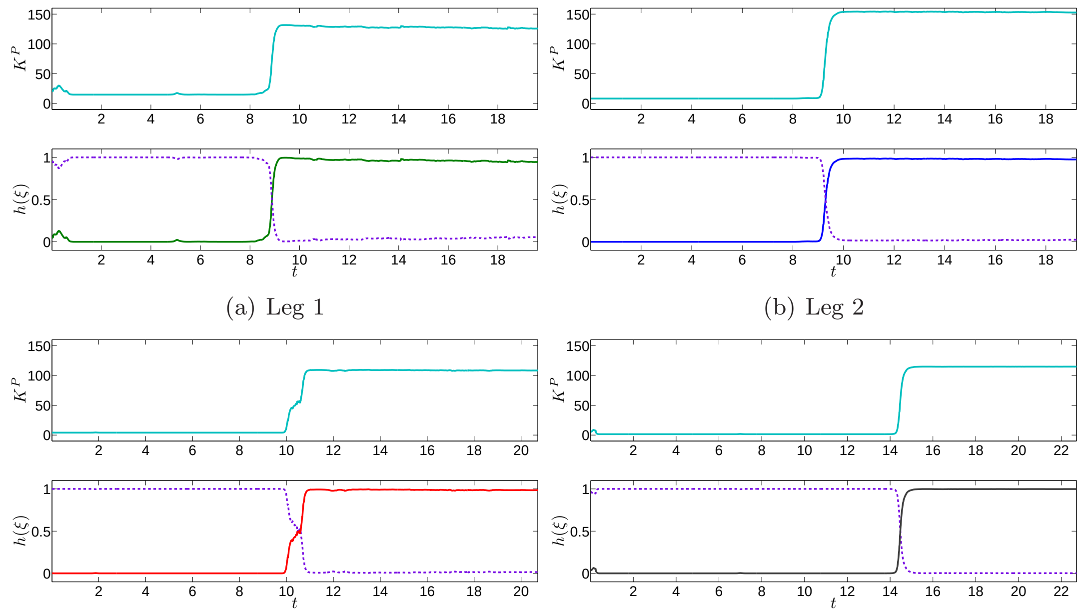


Figure 5.4: Compliance level estimation for the five-states PGMM using all the demonstrations from the training dataset.

the relative position of the leg with respect to the robot tool frame, which are different for each leg. The "compliant component" encodes the remaining points in the data space, i.e., the interaction forces-torques as well as the varying robot end-effector and leg positions.

Figure 5.2 displays the collected data during the demonstration of the assembly of one leg. The position and velocity of the robot's end-effector are shown in the first two rows, while forces and torques in the robot frame of reference are displayed in the next two rows. The last row shows the leg position with respect to the robot tool's frame. Here, it is possible to observe when the screwing process begins (just after the yellow area representing the compliant phase) and how the leg position remains nearly constant afterwards. Also, the force/torque pattern shows how the force applied along the z axis F_z as well as the torques generated by this force (T_x and T_y) change significantly when the leg is placed on the table.

Figure 5.3 shows that the Gaussian corresponding to the compliant phase is already spatially distinguishable from the Gaussians encoding the stiff behaviors during the screwing processes (four in this case). It should be noted that the Gaussian components in the PGMM representing the stiff phases show an elongated shape – in the spatial subspace – and change its orientation over time. Such type of time-varying information encapsulated in the covariance matrices cannot be encoded properly by using the classic PHMM (no covariance parameterization) or the modified version proposed in [65] based on scalar transformation.



(a) Leg 1 (b) Leg 2

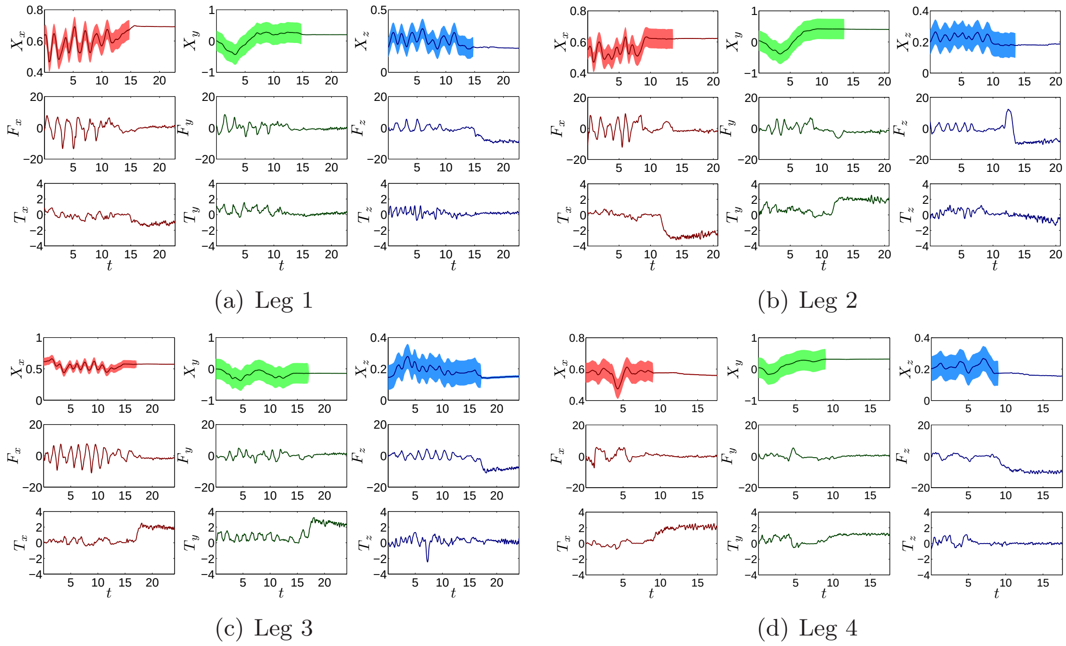
(c) Leg 3 (d) Leg 4

Figure 5.5: Resulting stiffness matrix trace described by $K^P = \sum_i^{N_s} h_i K_i^P$

5.3.2 Stiffness estimation results

Once the PGMM is learned, it is possible to carry out the stiffness matrix estimation as described in Section 5.2. It should be emphasized that a stiffness matrix is locally associated with each component in the PGMM, and it represents the stiffness of the virtual spring connected to center of the component. During reproduction, a force command is estimated as a combination of the virtual springs (see Figure 5.1 and Eq. 5.4).

The stiffness estimation based on the inverse of the observed covariance [28] is compared to the proposed approach in Figure 5.4. The bars display the trace of the estimated mean stiffness matrices for all the PGMM states. *Light blue* bars indicate the stiffness values $\boldsymbol{K}^P_{\text{Frb}}$ (using our approach) while *Light red* ones show the values $\boldsymbol{K}^P_{\text{Inv}}$ (using the position variability) for all the states (the first state corresponds to the compliant behavior). The inverse of the covariance sub-matrix $\boldsymbol{\Sigma}_i^{\boldsymbol{x}}$ in the Cartesian position space is computed for each Gaussian i. Such estimate is obtained for each datapoint of the training dataset. An average stiffness estimation $\boldsymbol{K}^P_{\text{Inv}}$ is then calculated. Note that the core idea of this method shares similarities with the stiffness estimation



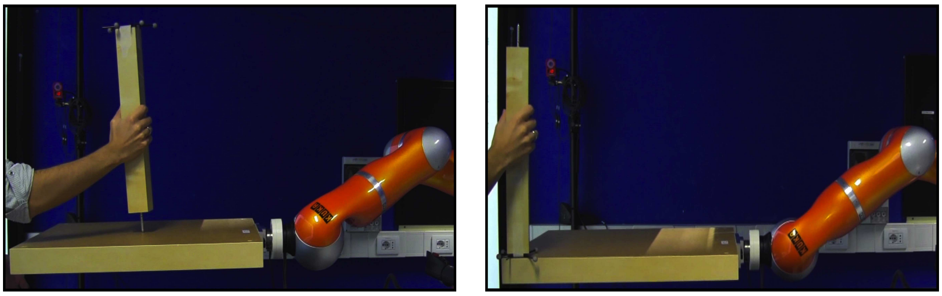
(a) Leg 1 (b) Leg 2

(c) Leg 3 (d) Leg 4

Figure 5.6: Estimated stiffness along the three Cartesian axes of the robot and the corresponding force/torque profiles.

proposed in [99], when time dependency is removed from the learning model. In such case, the estimates exclusively depend on the observed covariance of the spatial variables. On the other hand, $\boldsymbol{K}^{P}_{\text{Frb}}$ is the stiffness obtained using the approach proposed in this thesis, where only one initial stiffness estimation for each component is computed through a weighted least squares regression using all the demonstrations at a time, as described in Section 5.2. With the given training set, both approaches estimate the different stiffness levels appropriately.

However, the estimate based on the inverse of the observed covariance has the disadvantage that it takes only the positional information from the data into account, whose variability can sometimes be too weak if only a few number of demonstrations are considered. For example, the user may provide very similar examples without covering all of the possible variations that the task allows. In contrast, the approach that we adopt in this paper does consider the haptic inputs in the estimation process, which help to overcome the aforementioned drawback by providing additional and distinct sensory information to the learning framework. Figure 5.4 displays the trace of the estimated stiffness matrices for each Gaussian, comparing the results obtained by both approaches.



(a) Wrong leg position (away from the threads)
(b) Wrong orientation (table leg turned upside-down)

Figure 5.7: Tests in situations that have not been shown in the demonstration phase

The ratio between the stiff and compliant values (computed from the matrix traces) is higher using our approach (10.12 as average) than those obtained from the approach based on position variability (5.19 as average), which allows a better clamping of the robot stiffness considering the obtained maximum and minimum values. This indicates that the difference between the compliant and stiff levels is more pronounced when the estimation process is based on the proposed approach.

The reproduction and generalization capabilities of the system were evaluated by carrying out the assembly process for all the legs. In Figure 5.5, it is possible to observe how the compliant component (purple dotted line) is influential during the first half of the reproduction, dominating the rest of PGMM components. After this compliant phase, the robot becomes stiff, with specific patterns depending on which leg is being screwed. This means that not all the PGMM components influence the robot impedance at the stiff phase, but only the Gaussian encoding the stiff behavior for the corresponding leg is governing the robot behavior (as observed from the different colors representing the different stiff components), while the remaining component weights stay close to zero. Each plot displays how the weight belonging to each component changes over time – between 0 and 1 – showing its influence on the weighted least squares based stiffness estimation.

Figure 5.6 shows the resulting stiffness matrix for the demonstration corresponding to leg 1, obtained by the model displayed in Figure 5.3, where the Cartesian robot position is shown along with the corresponding stiffness value for each Cartesian axis. Here, it is possible to observe how different stiffness values for the different Cartesian axes can be learned from the demonstrations, which is useful when the robot behavior demands to be stiff along a specific direction while being compliant along the remaining ones. The light color zone at the first row represents the robot arm compliance (the wider the zone is, the more compliant the robot behaves). When the envelope is wide the robot can be moved easily, while the narrow envelope represents the table holding phase constrained by high stiffness values along each coordinate axis.

In order to show the relevance of merging visual and haptic data, two additional situations that did not appear in the demonstration phase were presented to the robot. In the first case, the human holding the leg tried to screw it at the center of the table, which means that the leg was placed at an incorrect position (see Figure 5.7(a)). In the second situation, the leg is positioned in one of the table threads but this time it was tilted with an orientation at which the screwing process was unfeasible (see Figure 5.7(b)). In both cases, the robot behaved compliantly as expected, because neither of the human actions should make the robot behave stiffly.

5.4 Chapter highlights

This chapter presented a novel learning framework to encode and reproduce impedance-based robot behaviors using a parametric probabilistic model. The proposed approach allows to encode reactive impedance behaviors that rely on task variables, yielding only one model to encode the whole task. Unlike PHMM, the learning algorithm considers varying variables along the task and allows to modify the center and covariance of the components over time accordingly. In contrast to other previous approaches where the robot impedance is learned from demonstrated trajectories (i.e., using position information), the robot instead extracts the impedance-based behavior of the teacher recording both force patterns and visual information. Force-based perceptions are used not only to encode the task, but also to estimate the stiffness of virtual springs governing the collaborative behavior that is used to control the robot, thus emphasizing the fact that interaction force-torque profiles are different during different phases of the task.

It is worth mentioning that the ideas presented in this chapter constitute one step forward to achieve a multimodal LfD framework, in which a robot may benefit from a set of different sources of information. Vision and force data may also be merged with auditory and tactile perceptions, which may be exploited in robot coaching scenarios where the robot interacts with the teacher through refinements and corrections, or also in collaborative tasks enhancing the communication channel between partners. In this context, PGMM would encapsulate the task variables and the input variables in a compact and more generic model. Time constraints may also be taken into account in the encoding of the task by introducing temporal relationships among the model components as those presented in the HMM (see Section 4.3). Furthermore, the proposed approach also allows to learn impedance-based behaviors, a promising field of research for creating safer and user-friendly robots.

Chapter 6

Conclusions

Throughout this thesis, several challenging issues in the field of force-based learning from demonstration were tackled. Three main concerns were satisfactorily addressed, namely: *(i)* what to imitate from human demonstrations? *(ii)* how to learn and reproduce a force-based task? and *(iii)* how to exploit the combination of vision and force information? This chapter presents the conclusions obtained from the approaches proposed to solve the aforementioned questions.

6.1 Selecting relevant perceptions

The *what to imitate?* problem was solved from a new perspective based on the Mutual Information analysis. The proposed approach allows to select the relevant inputs of a task by analyzing their influence on the robot actions (see Section 3.2). This methodology suits satisfactorily our frameworks, sharing concepts from control policy learning where functions map perceptions to actions. In this context, this dissertation addressed two shortcomings of the approach:

- Classic Mutual Information analysis selects the input variables without considering the information that other inputs provide. This limitation was successfully solved by conditioning the selection of a given perception on the information provided by the previously chosen inputs. This allows not only to select the most relevant perceptions, but also to reduce the redundancy in the data.

- Common selection processes are not fully automatic and assume that the number of inputs to be chosen is given by the user. This thesis proposed a threshold-driven automatic selection where the number of perceptions is automatically determined by evaluating a conditional mutual information ratio (see Algorithm 2).

The obtained results show that the proposed methodology successfully performs in different kinds of LfD scenarios, where the selected perceptions provided enough information to encode and reproduce the tasks satisfactorily. Nevertheless, several challenging topics are still open. For instance, the proposed stopping criterion depends on a threshold whose value is determined by the user according to how selective the method is supposed to be. Also, in the case of multiple outputs, the mutual information is averaged assuming that every output has the same relevance. These issues are of high interest for future research as discussed in Chapter 7.

6.2 Learning manipulation tasks from haptic inputs

A novel whole learning framework has been developed in this thesis for encoding and reproducing force-based manipulation skills, where probabilistic methods have been exploited for solving the following problems:

- Force-based tasks often show large time discrepancies in the data, and may also suffer from perceptual aliasing. The first proposed learning framework addressed these issues by taking advantage of the sequential information embedded in the states representation of the hidden Markov model. This method avoids to use time as an explicit input, and also allows the robot to look back at the evolution of the task (by using the transition probabilities), so that it can distinguish the correct action when the current perception does not have a unique output command to be performed.

- The robot actions may be conditioned on a set of force-based parameters. This issue was successfully tackled by exploiting the parametric formulation of the classic hidden Markov model. Such an approach allows to encode force parametrized tasks by linearly modulating the robot actions according to a force variable, in contrast to former works where the task parameters were represented by the location – and possibly the orientation – of objects to be manipulated.

The resulting learning structure performed well in two different scenarios, showing good generalization capabilities when facing unseen conditions. Therefore, this new framework can be considered as a step forward to achieve a more generic and versatile structure to learn tasks in the force domain from human demonstrations. Nonetheless, its two main limitations are: *(i)* the parameters do not affect the covariance information of the model states, and *(ii)* it is not possible to define how long the system stays at a specific state of the model, which is represented by the self-transition probabilities. The first drawback was addressed in Chapter 5, while the second one is considered as future work of this thesis (see Section 7.2).

6.3 Merging visual and haptic information in collaborative manipulation

In collaborative manipulation tasks, the partners actions often rely not only on the information provided by their vision sensors, but also on the force data. Moreover, in such kind of complex scenarios, the task performed by a robot is not merely to follow a given trajectory. In this context, this dissertation contributed to solve the following aspects:

- Vision and force data need to be combined in a compact way. The proposed solution was to parametrize the force-based task by using the information coming from the vision system. This was achieved by taking advantage of a complete parametric Gaussian mixture model, which also allowed to modulate the covariance of its components, outperforming the encoding capabilities of the parametric learning structure proposed in Chapter 4 in the case of parametrized tasks.
- The robot role in the task was based on a reactive behavior. In this context, this thesis was aimed at extending LfD to impedance-based behaviors. The core idea was to assume that the robot motion was driven by a set of virtual springs. Impedance learning took place when both the position of the springs and their stiffness are computed from the demonstrations provided by a human.

The proposed impedance-based learning framework permitted to learn reactive impedance behaviors in a collaborative manipulation task, where the robot was able to shape its compliance according to the needs of its partner. Such an approach does exploit the new control schemes of torque-controlled robots and the combination of vision and force information, and hence allows to learn safer and more compliant robot actions. At the current stage of this research, only reactive roles can be learned. In order the robot to carry out a larger variety of collaborative tasks, it should be endowed with a more active role. Such an aspect is part of the future work of this thesis and is discussed in Section 7.5.

Chapter 7

Future work

Although the objectives of this dissertation were successfully achieved, there are still some issues of importance to be tackled in the future. This chapter is aimed at discussing new possible routes of research arising from the work presented in the previous chapters, taking into consideration the current and future requirements to be fulfilled in order to build smarter and friendlier robot companions.

7.1 Extensions of the Mutual Information analysis

The mutual information-based selection algorithm proposed in Section 3.3 does not consider the case in which the relevant perceptions change over the course of the task. Such situation may happen, for instance, when the robot is endowed with a multimodal perception system, because it may use different sets of sensors during the execution of the skill. Here, it would be desirable that the solution to the *what to imitate?* problem selects online the relevant perceptions for every phase of the task according to its constraints and the given demonstrations. This is a very challenging problem because the selected perceptions would change over time and the learning framework should be able to handle this situation, possibly through adaptive and incremental models [91].

Another open problem concerning how mutual information works is related to multiple-input/multiple-output cases. In Section 3.3, Algorithm 1 selects the most relevant variables considering that all the outputs have the same influence on the selection process (note that all the mutual information values are averaged). This might be a strong assumption when the subset of relevant inputs is different for each output, which may result in very similar MI scores for all the inputs after averaging. Thus, it would be desirable to know how much a specific output variable contributes to accomplish the task successfully. In this sense, a goal-driven selection process based on mutual information may be implemented in two phases, namely, *(i)* a weight w_j is computed for each output j according to its contribution to achieve the task goal, then *(ii)* a weighted average of the mutual information values is computed to select the most relevant inputs.

Finally, note that mutual information analysis applied to the *pouring task* showed that T_x, F_z and T_y were the most relevant variables (see Section 3.4.2). Then, the initial values of two of them composed the task parameter $\boldsymbol{\theta}_m$ for encoding the task through a PHMM (details in Section 4.4.2). In this particular case, mutual information results provided some cues about the variables modulating the centers of the Gaussians in the model. Therefore, this may be further explored in order to take advantage of the mutual information estimates for automatically determining the vector $\boldsymbol{\theta}_m$, instead of setting it manually.

7.2 Hidden semi-Markov models as a more generic tool for encoding temporal information

One of the main limitations of the classic hidden Markov model is the fact that it does not allow to define how long the system stays at a specific state of the model, which is restrictively represented by the self-transition probabilities $\{a_{ii}\}$. This drawback is more evident when the HMM is desired to encode time constraints of the task (which should not be confused with the encoding of the sequential information, that is how the system progresses through the model states) as retrieving duration information based on the fixed set of self-transition probabilities of conventional HMM does not provide a good model of state duration. It is however possible to modify the structure of the HMM to replace the self-transition of each state i with a parametric model $p_i(d)$ of the state duration $d \in \{1, 2, \ldots, d_{max}\}$, where d_{max} determines a maximum limit for the number of iterations that the system can stay in a state. This approach is known as the hidden semi-Markov model [167].

This new model would allow to encode manipulation tasks in a more robust and generic way, dealing with time constraints and sequential information as well as encoding the observations through a probabilistic representation. Moreover, if models handling task parameters (e.g., PHMM or PGMM) were integrated in this approach, the resulting learning framework might encode and reproduce a larger set of skills. Thus, the main idea is to improve the framework proposed in Section 1.3 (see Figure 1.2) by including the parameterization of the state duration $p_i(d)$ in the parametric hidden Markov model. After this, covariance parameterization may also be added by taking inspiration from the PGMM.

7.3 Extending the PGMM capabilities to deal with parameters coming from different perception systems

The parametric Gaussian mixture model was introduced in Section 5.1.1 as a possible extension of the parametric version of the HMM (see Section 4.4.1), where the mean and covariance of the model components are modified by the task parameter. However, one of the drawbacks of the PGMM is the fact that the parameters are defined as frames of reference, mainly expressed as the location and orientation of objects of interest for the problem at hand. This significantly limits the range of tasks where PGMM may be used, because this model, for instance, may not be able to encode skills with force-based parameters, as the one described in Section 1.4.2. In contrast, PHMM does provide this feature and thus parameters coming from different sources can be encapsulated by this method. Therefore, the challenge regarding this issue is to analyze how the PGMM parameters (i.e., the transformation matrix $\boldsymbol{A}$ and the vector $\boldsymbol{b}$) may be used to encode several kinds of task parameters.

At first sight, the main problem is how to set the matrix $\boldsymbol{A}$ when not representing a frame of reference. Thus, it is necessary to study how the task parameters condition the robot actions and how the model parameters should be defined to express such dependence. A first step to solve this issue might be to consider learning a task where the provided demonstrations were represented at the operational and joint spaces of the robot, which would demand the PGMM to deal with frames in different spaces. Hence, this would involve to take into account an operator mapping the robot actions between both spaces, i.e., the Jacobian of the robot. In such a way, it would be possible to see how the model behaves when the parameters $\{\boldsymbol{A}, \boldsymbol{b}\}$ do not represent exclusively locations and orientations of specific Cartesian frames.

7.4 Impedance-based behavior learning

Most of the machine learning tools developed so far are decomposed into an offline model estimation phase and a retrieval/regression phase (see Section 2.4). Instead, learning in compliant robots should view demonstration and reproduction as an interlaced process that can combine both imitation and reinforcement learning strategies to incrementally refine the task. The development of compliant robots brings up new challenges in machine learning and physical human-robot interaction, by extending the skill transfer problem towards tasks involving force information, and towards systems capable of learning how to cope with various sources of perturbation introduced by the user and the task. In this thesis a model based on a set of virtual springs was proposed to control the motion of a robotic arm by using force commands (see Section 5.2). The complete learning framework is able to learn reactive impedance behaviors using position and haptic information. Nevertheless, the approach does not take into consideration the velocity data which may enhance the robot performance. This can be achieved by including virtual dampers to the model.

Having a virtual set of spring-damper systems makes the estimation of the impedance parameters (i.e., the stiffness and damping) from the given demonstrations a challenging problem. The approach presented in Section 5.2 should be reformulated in order to include the estimation of the damping parameter, by taking inspiration from the work developed by Erickson *et al.* [46] where the contact stiffness and damping are identified during robot constrained motion. The idea is to keep using position and haptic information in the estimation process but this time for a more robust and stable model.

7.5 Haptic inputs in role determination for physical HRI

One of the main contributions of this thesis is based on exploiting haptic data in LfD scenarios. Transferring collaborative manipulation skills to robots in a user-friendly manner was the core of the experimental setup and the proposed learning framework briefly explained in Sections 1.4.3 and 1.3, respectively. Here, the robot was able to learn reactive impedance-based behaviors in order to assist the human partner to develop the task satisfactorily. However, the robot should also be able to play a more active role, in other words, the robot should be a proactive partner. In order to achieve this, the robot needs to recognize the human intention through action anticipation. Haptic inputs have shown to be a rich and complex communication channel to communicate intent (see Section 2.6.1).

It would be possible to endow the robotic partner with reactive and proactive behaviors by taking advantage of the properties of the hidden semi-Markov model, which can encode position and haptic information in a probabilistic model while encapsulating time constraints of the collaborative task. This approach may be used for both learning the skill and recognizing the human intention. Thus, the improvement of the learning framework proposed in Section 7.2 may also be exploited in this type of human-robot collaborative tasks.

Appendix A

Publications by the author

This appendix lists the publications of the author with a brief comment on how they are connected to this thesis.

- **Rozo, L.**, Jimenez, P. and Torras, C. (2010). Learning Force-Based Robot Skills from Haptic Demonstration. *13th International Conference of the Catalan Association for Artificial Intelligence (CCIA)*, Tarragona-Spain, pp. 331-340.
  This paper presents a performance comparison in terms of the mean squared error of a robot manipulator carrying out the ball-in-box task (described in Section 1.4.1). Approaches based on the Locally Weighted Learning and Gaussian Mixture Model are analyzed in the context of encoding and reproduction of a force-based task, from which the analysis described in Section 4.1 has been derived. Computational results showed very similar error values, but both failed during the experimental execution of the task when facing multivalued cases. This permitted devising new research stages where the sequential information of the task was taken into account (see Section 4.3).

- **Rozo, L.**, Jimenez, P. and Torras, C. (2010). Sharpening Haptic Inputs for Teaching a Manipulation Skill to a Robot. *1st International Conference on Applied Bionics and Biomechanics (ICABB)*, Venice-Italy, pp. 370-377.
  The solution to the *what to imitate?* problem based on mutual information is introduced in this paper. Here, the input variables are exclusively selected according to the ranking obtained from the computation of the mutual information among every input/output pair, which is explained at the beginning of Section

3.3. Also, issues like signal pre-processing and dynamic compensation when using force signals are addressed in this work (see Section 3.1). The results allowed to observe the possible limitations of the input selection algorithm for more complex tasks, which demanded to look for more sophisticated approaches to robustly choose the relevant perceptions in LfD scenarios.

- **Rozo, L.**, Jimenez, P. and Torras, C. (2011). Robot Learning from Demonstration of Force-based Tasks with Multiple Solution Trajectories. *15th International Conference on Advanced Robotics (ICAR)*, Tallin-Estonia, pp. 124-129.
  Here, the advantages of working with hidden Markov models and Gaussian mixture regression in the context of force-based learning were shown through the *ball-in-box* task (see Section 1.4.1). The paper presented the analysis and results regarding how to exploit sequential information for learning tasks with multiple solution trajectories (Section 1.1). The promising results encouraged the use of these techniques in more realistic scenarios.

- **Rozo, L.**, Jimenez, P. and Torras, C. (2011). Robot Learning from Demonstration in the Force Domain. *22th International Joint Conference on Artificial Intelligence (IJCAI), Workshop on Agents Learning Interactively from Human Teachers*, Barcelona-Spain, pp. 1-6.
  The first whole learning framework proposed in this thesis (Figure 1.1) was introduced in this paper. Results concerning mutual information analysis (Section 3.4.1), encoding through hidden Markov models and reproduction using Gaussian mixture regression (Section 4.3.3) are presented and analyzed in the context of the *ball-in-box* task (see Section 1.4.1). Also, performance criteria based on success/failure and time-based measures were used to assess a degree of accomplishment of the task.

- Pardo, D., **Rozo, L.**, Alenya, G. and Torras, C. (2012). Dynamically Consistent Probabilistic Model for Robot Motion Learning. *International Conference on Intelligent Robot Systems (IROS), Workshop on Learning and Interaction in Haptic Robots*, Vilamoura-Portugal, pp. 1-2.
  This work presented how an extended representation of the state of the robot (i.e., not only the current kinematic position but also on its first time derivative)

can be used to obtain a consistent representation of the dynamics of the task. The concepts related to autonomous systems introduced in Section 4.1 were applied here. The learning framework proposed previously showed to benefit from this complete representation mostly when the robot carries out tasks characterized by an underlying multivalued function.

- **Rozo, L.**, Jimenez, P. and Torras, C. (2013). A Robot Learning from Demonstration Framework to Perform Force-based Manipulation Tasks. *Journal of Intelligent Service Robotics, Special Issue on Artificial Intelligence Techniques for Robotics: Sensing, Representation and Action, Part 2*, 6(1):33-51.
  This paper presented an improvement in the input variable selection process by incorporating the conditional mutual information criterion just after the most relevant input has been selected (see Section 3.3 and Algorithm 1). On the other hand, a more realistic task was designed to evaluate the generality of the proposed framework by learning to pour drinks based on force data exclusively (Section 1.4.2). The obtained results showed that the new perception selection algorithm, the encoding and reproduction methods also performed satisfactorily in other kind of scenarios.

- **Rozo, L.**, Jimenez, P. and Torras, C. (2013). Force-based Robot Learning of Pouring Skills using Parametric Hidden Markov Models. *9th International Workshop on Robot Motion and Control*, *Accepted*.
  The extension for learning force-based parametrized tasks explained in Section 4.4 was introduced in this work. Here force parameters are exploited to obtain a compact encoding of skills that are exclusively based on haptic perceptions. The proposed learning framework (Figure 1.2) is based on a parametric version of the classic HMM and Gaussian mixture regression to encapsulate the demonstrations and reproduce the task, respectively. Tests were carried out on the same pouring task (Section 1.4.2) used for testing the first proposed framework, where the advantages of this extension were evident (as explained in Section 4.4.2).

- **Rozo, L.**, Calinon, S., Caldwell, D., Jimenez, P. and Torras, C. (2013). Learning Parametrized Impedance-based Robot Behaviors. *27th AAAI Conference on Artificial Intelligence*, *Accepted*.

This paper summarizes the approach and results presented in Chapter 5. Here, a novel learning framework (Figure 1.3) was proposed to transfer impedance-based behaviors to a torque-controlled robot, with demonstrations provided by kinesthetic teaching. The proposed model encodes the examples as a task-parameterized statistical dynamical system, where the robot impedance is shaped by estimating virtual stiffness matrices from the set of demonstrations, as explained in Sections 5.1 and 5.2, respectively. The collaborative assembly task described in Section 1.4.3 was used as testbed. The results showed that the model can be used to modify the robot impedance along task execution to facilitate the collaboration, by triggering stiff and compliant behaviors in an on-line manner to adapt to the user's actions (see Section 5.3).

Bibliography

[1] A. Ajoudani, N. Tsagarakis, and A. Bicchi. Tele-impedance: Towards transferring human impedance regulation skills to robots. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 382–388, 2012.

[2] B. Akgun, M. Cakmak, J. Yoo, and A. Thomaz. Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In *ACM/IEEE Intl. Conf. on Human-Robot Interaction (HRI)*, pages 391–398, 2012.

[3] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger. The DLR lightweight robot - design and control concepts for robots in human environments. *Industrial Robot: An Intl. Journal*, 34(5):376–385, 2007.

[4] A. Albu-Schäffer, O. Eiberger, M. Grebenstein, S. Haddadin, C. Ott, T. Wimböck, S. Wolf, and G. Hirzinger. Soft robotics. *IEEE Robotics and Automation Magazine*, 15(3):2–30, 2008.

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

[6] C. Atkeson and S. Schaal. Learning tasks from a single demonstration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1706–1712, 1997.

[7] C. Atkeson, A. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.

[8] P. Bakker and Y. Kuniyoshi. Robot see, robot do: An overview of robot imitation. In *AISB Workshop on Learning in Robots and Animals*, pages 3–11, 1996.

[9] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994.

[10] D. Bentivegna, A. Ude, C. Atkeson, and G. Cheng. Humanoid robot learning and game playing using pc-based vision. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2449–2454, 2002.

[11] A. Billard, S. Calinon, and F. Guenter. Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics and Autonomous Systems*, 54:370–384, 2006.

[12] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot programming by demonstration. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 1371–1394. Springer, Secaucus, NJ, USA, 2008.

[13] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[14] E. Bizzi and F. Mussa-Ivaldi. The acquisition of motor behavior. *Daedalus*, 127: 217–232, 1998.

[15] M. Brand and A. Hertzmann. Style machines. In *Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 183–192, 2000.

[16] M. Brass and C. Heyes. Imitation: Is cognitive neuroscience solving the correspondence problem? *Trends in Cognitive Sciences*, 9(10):489–495, 2005.

[17] C. Breazeal and B. Scassellati. Robots that imitate humans. *Trends in Cognitive Science*, pages 481–487, 2002.

[18] E. Burdet, R. Osu, D. Franklin, T. Milner, and M. Kawato. The central nervous system stabilizes unstable dynamics by learning optimal impedance. *Nature*, 414 (6862):446–449, 2001.

[19] J. Butterfield, S. Osentosky, G. Jay, and O. C. Jenkins. Learning from demonstration using a multi-valued function regressor for time-series data. In *IEEE/RAS Intl. Conf. on Humanoid Robots (Humanoids)*, pages 328–333, 2010.

[20] M. Cakmak and A. Thomaz. Designing robot learners that ask good questions. In *ACM/IEEE Intl. Conf. on Human-Robot Interaction (HRI)*, pages 17–24, 2012.

[21] S. Calinon. *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press, 2009.

[22] S. Calinon and A. Billard. Stochastic gesture reproduction and recognition model for a humanoid robot. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2769–2774, 2004.

[23] S. Calinon and A. Billard. Learning of gestures by imitation in a humanoid robot. In *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*, pages 153–177. 2007. K. Dautenhahn and C.L. Nehaniv (Eds).

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

[25] S. Calinon and A. Billard. A probabilistic programming by demonstration framework handling constraints in joint space and task space. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 367–372, 2008.

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

[27] S. Calinon, F. D'halluin, E. Sauser, D. Caldwell, and A. Billard. Learning and reproduction of gestures by imitation - an approach based on hidden Markov model and Gaussian mixture regression. *IEEE Robotics and Automation Magazine*, 17 (2):44–54, 2010.

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

[29] S. Calinon, A. Pistillo, and D. Caldwell. Encoding the time and space constraints of a task in explicit-duration hidden Markov model. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3413–3418, 2011.

[30] S. Calinon, Z. Li, T. Alizadeh, N. Tsagarakis, and D. Caldwell. Statistical dynamical systems for skills acquisition in humanoids. In *IEEE/RAS Intl. Conf. on Humanoid Robots (Humanoids)*, pages 323–329, 2012.

[31] T. Cederborg, L. Ming, A. Baranes, and P-Y. Oudeyer. Incremental local online Gaussian mixture regression for imitation learning of multiple tasks. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 267–274, 2010.

[32] S. Chatzis, D. Korkinof, and Y. Demiris. A quantum-statistical approach towards robot learning by demonstration. *Transactions on Robotics*, 28(6):1371–1381, 2012.

[33] Y. Chen and J. McInroy. Estimating symmetric, positive definite matrices in robotic control. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4269–4274, 2002.

[34] A. Clark and R. Grush. Towards a cognitive robotics. *Adaptive Behavior*, 7(1): 5–16, 1999.

[35] F. Dallalibera, F. Basoeki, T. Minato, H. Ishiguro, and E. Menegatti. Teaching by touching : Interpretation of tactile instructions for motion development. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3480–3487, 2011.

[36] K. Dautenhahn. Socially intelligent robots: dimensions of human-robot interaction. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 352(1480):679–704, 2007.

[37] K. Dautenhahn and C. Nehaniv. *Imitation in animals and artifacts*. MIT Press, 2002.

[38] P. Dayan. Unsupervised learning. In *The MIT Encyclopedia of the Cognitive Sciences*, pages 1–7. MIT press, 1999.

[39] Y. Demiris. Mirror neurons, imitation and the learning of movement sequences. In *Intl. Conf. on Neural Information Processing (ICONIP)*, pages 111–115, 2002.

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

[41] R. Dillmann, M. Kaiser, and A. Ude. Acquisition of elementary robot skills from human demonstration. In *Intl. Symposium on Intelligent Robotics Systems*, pages 185–192, 1995.

[42] K. Dines. Constrained least squares filtering. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 25(4):346–350, 1977.

[43] S. Dong and F. Naghdy. Application of hidden Markov model to acquisition of manipulation skills from haptic rendered virtual environment. *Robotics and Computer-Integrated Manufacturing*, 23(3):351–360, 2007.

[44] S. Dong and B. Williams. Motion learning in variable environments using probabilistic flow tubes. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1976–1981, 2011.

[45] D. Economou, C. Lee, C. Mavroidis, and I. Antoniadis. Robust vibration suppression in flexible payloads carried by robot manipulators using digital filtering of joint trajectories. In *Intl. Symposium on Robotics and Automation*, pages 244–249, 2000.

[46] D. Erickson, M. Weber, and I. Sharf. Contact stiffness and damping estimation for robotic systems. *Intl. Journal of Robotics Research*, 22(1):41–57, 2003.

[47] M. Ernst and M. Banks. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415(6870):429–433, 2002.

[48] P. Estévez, M. Tesmer, C. Perez, and J. Zurada. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201, 2009.

[49] P. Evrard and A. Kheddar. Homotopy switching model for dyad haptic interaction in physical collaborative tasks. In *Joint EuroHaptics Conf. and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 45–50, 2009.

[50] P. Evrard, E. Gribovskaya, S. Calinon, A. Billard, and A. Kheddar. Teaching physical collaborative tasks: Object-lifting case study with a humanoid. In *IEEE/RAS Intl. Conf. on Humanoid Robots (Humanoids)*, pages 399–404, 2009.

[51] P. Falco, R. Jäkel, C. Natale, and R. Dillmann. Improvement of human hand motion observation by exploiting contact force measurements. In *IEEE-RAS Intl. Conf. on Humanoid Robots (Humanoids)*, pages 141–146, 2011.

[52] F. Flacco and A. De Luca. Residual-based stiffness estimation in robots with flexible transmissions. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 5541–5547, 2011.

[53] G. Ganesh, A. Albu-Schäffer, M. Haruno, M. Kawato, and E. Burdet. Biomimetic motor behavior for simultaneous adaptation of force, impedance and trajectory in interaction tasks. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2705–2711, 2010.

[54] P. Garthwaite. An interpretation of partial least squares. *Journal of the American Statistical Association*, 89(425):122–127, 1994.

[55] Z. Ghahramani and M. Jordan. Supervised learning from incomplete data via EM approach. In *Neural Information Processing Systems (NIPS)*, pages 120–127, 1994.

[56] Z. Ghahramani and M. Jordan. Factorial hidden Markov models. *Machine Learning*, 29(2):245–273, 1997.

[57] B. Goldstein. *Sensation and perception.* Cengage Learning, 2009.

[58] H. Gomi and M. Kawato. Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement. *Science*, 272(5258):117–120, 1996.

[59] S. Grange, F. Conti, P. Helmer, P. Rouiller, and C. Baur. Overview of the delta haptic device. In *Eurohaptics*, pages 1–6, 2001.

[60] E. Gribovskaya, A. Kheddar, and A. Billard. Motion learning and adaptive impedance for robot control during physical interaction with humans. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4326–4332, 2011.

[61] D. Grimes, R. Chalodhorn, and R. Rao. Dynamic imitation in a humanoid robot through nonparametric probabilistic inference. In *Robotics: Science and Systems (R:SS)*, pages 1–8, 2006.

[62] D. Grollman and O. Jenkins. Sparse incremental learning for interactive robot control policy estimation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3315–3320, 2008.

[63] I. Guyon. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[64] S. Haddadin, S. Haddadin, A. Khoury, T. Rokahr, S. Parusel, R. Burgkart, A. Bicchi, and A. Albu-Schäffer. On making robots understand safety: Embedding injury knowledge into control. *Intl. Journal of Robotics Research*, 31(13):1578–1602, 2012.

[65] D. Herzog, V. Krüger, and D. Grest. Parametric hidden Markov models for recognition and synthesis of movements. In *British Machine Vision Conf.*, pages 163–172, 2008.

[66] N. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications*, 103:103–118, 1988.

[67] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[68] H. Hoffmann, P. Pastor, D. Park, and S. Schaal. Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2587–2592, 2009.

[69] N. Hogan. Impedance control: An approach to manipulation: Part i, ii, iii. *Journal of Dynamic Systems, Measurement, and Control*, 107(1):1–24, 1985.

[70] A. Howard and C. Park. Haptically guided teleoperation for learning manipulation tasks. In *Robotics: Science and Systems R:SS - Workshop on Robot Manipulation*, pages 1–6, 2007.

[71] M. Howard, D. Mitrovic, and S. Vijayakumar. Transferring impedance control strategies between heterogeneous systems via apprenticeship learning. In *IEEE/RAS Intl. Conf. on Humanoid Robots (Humanoids)*, pages 98 – 105, 2010.

[72] K. Hsiao and T. Lozano-Perez. Imitation learning of whole-body grasps. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 5657–5662, 2006.

[73] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4–5):411–430, 2000.

[74] A. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1398–1403, 2002.

[75] A. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In *Neural Information Processing Systems (NIPS)*, pages 1523–1530, 2003.

[76] T. Ikeda, H. Ishiguro, and M. Asada. Adaptive fusion of sensor signals based on mutual information maximization. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4398–4402, 2003.

[77] O. Jenkins, R. Peters, and R. Bodenheimer. Uncovering success in manipulation. In *Robotics: Science and Systems (R:SS) - Workshop on Manipulation in Human Environments*, pages 1–6, 2006.

[78] I. Jolliffe. *Principal Component Analysis*. Springer, 2002.

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

[80] M. Kaiser and R. Dillmann. Building elementary robot skills from human demonstration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2700–2705, 1996.

[81] C. Kemp, A. Edsinger, and E. Torres-Jara. Challenges for robot manipulation in human environments. *IEEE Robotics and Automation Magazine*, 14(1):20–29, 2007.

[82] M. Khansari-Zadeh and A. Billard. Learning stable nonlinear dynamical systems with Gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011.

[83] J. Kober, B. Mohler, and J. Peters. Learning perceptual coupling for motor primitives. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 22–26, 2008.

[84] P. Kormushev, S. Calinon, and D. Caldwell. Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input. *Advanced Robotics*, 25(5):581–603, 2011.

[85] N. Krislock, J. Lang, J. Varah, D. Pai, and H. Seidel. Local compliance estimation via positive semidefinite constrained least squares. *IEEE Transactions on Robotics*, 20(6):1007–1011, 2004.

[86] V. Krüger, D. Herzog, S. Baby, A. Ude, and D. Kragic. Learning actions from observations. *IEEE Robotics and Automation Magazine*, 17(2):30–43, 2010.

[87] V. Krüger, V. Tikhanoff, L. Natale, and G. Sandini. Imitation learning of non-linear point-to-point robot motions using Dirichlet processes. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2029–2034, 2012.

[88] D. Kulić and Y. Nakamura. Incremental learning of full body motions primitives. In *IEEE/RAS Intl. Conf. on Humanoid Robots (Humanoids)*, pages 326–332, 2008.

[89] D. Kulić and Y. Nakamura. Incremental learning of human behaviors using hierarchical hidden Markov models. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 4649–4655, 2010.

[90] D. Kulić, W. Takano, and Y. Nakamura. Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden Markov chains. *Intl. Journal of Robotics Research*, 27(7):761–784, 2008.

[91] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura. Incremental learning of full body motion primitives and their sequencing through human motion observation. *Intl. Journal of Robotics Research*, 31(3):330–345, 2012.

[92] T. Kulvicius, K. Ning, M. Tamosiunaite, and F. Wörgötter. Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on hadwriting. *IEEE Transactions on Robotics*, 28(1):145–157, 2012.

[93] M. Kumar and D. Garg. Sensor-based estimation and control of forces and moments in multiple cooperative robots. *ASME Journal of Dynamic Systems, Measurement, and Control*, 126(2):276–283, 2004.

[94] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822, 1994.

[95] M. Kuschel, M. Di Luca, M. Buss, and R. Klatzky. Combination and integration in the perception of visual-haptic compliance information. *IEEE Transactions on Haptics*, 3(4):234–244, 2010.

[96] N. Kwak and C. Choi. Input feature selection for classification problems. *IEEE Transactions on Neural Networks*, 13(1):143–159, 2002.

[97] N. Kwak and C. Choi. Input feature selection by mutual information based on Parzen window. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1667–1671, 2002.

[98] D. Lee and Y. Nakamura. Stochastic model of imitating a new observed motion based on the acquired motion primitives. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 4994–5000, 2006.

[99] D. Lee and C. Ott. Incremental kinesthetic teaching of motion primitves using the motion refinement tube. *Autonomous Robots*, 31:115–131, 2011.

[100] H. Lee and J. Kim. An HMM-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10): 961–973, 1999.

[101] A. Levas and M. Selfridge. A user-friendly high-level robot teaching system. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, volume 1, pages 413–416, 1984.

[102] S. Lin. Force sensing using Kalman filtering techniques for robot compliant motion control. *Journal of Intelligent and Robotic Systems*, 18:1–16, 1997.

[103] T. Lozano-Perez. Robot programming. *Proceedings of the IEEE*, 71(7):821–841, 1983.

[104] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai. Speech synthesis using HMMs with dynamic features. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 389–392, 1996.

[105] M. Mataric. Getting humanoids to move and imitate. *IEEE Intelligent Systems*, 15(4):18–24, 2000.

[106] H. Mayer, D. Burschka, A. Knoll, E. Braun, R. Bauernschmitt, and R. Lange. Human-machine skill transfer extended by a scaffolding framework. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2866–2871, 2008.

[107] S. McAmis and K. Reed. Simultaneous perception of forces and motions using bimanual interactions. *IEEE Transactions on Haptics*, 5(3):220–230, 2012.

[108] B. McCarragher. Force sensing from human demonstration: stiffness, impedance and kinesthetic sensibility. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1226–1233, 1994.

[109] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[110] H. Miyamoto and M. Kawato. A tennis serve and upswing learning robot based on bi-directional theory. *Neural Networks*, 11:1331–1344, 1998.

[111] S. Münch, J. Kreuziger, M. Kaiser, and R. Dillmann. Robot programming by demonstration (rpd) - using machine learning and user interaction methods for the development of easy and comfortable robot programming systems. In *Proc. Intl. Symposium on Industrial Robots*, pages 685–693, 1994.

[112] K. Murphy. *Machine Learning - A Probabilistic Perspective*. MIT Press, 2012.

[113] B. Mutlu, A. Terrell, and C. Huang. Coordination mechanisms in human-robot collaboration. In *ACM/IEEE Intl. Conf. on Human-Robot Interaction (HRI) - Workshop on Collaborative Manipulation*, pages 1–6, 2013.

[114] C. Myers and L. Rabiner. A comparative study of several dynamic time-warping algorithms for connected-word recognition. *The Bell System Technical Journal*, 60(7):1389–1409, 1981.

[115] C. Nehaniv and K. Dautenhahn. Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications. *Interdisciplinary Approaches to Robot Learning, World Scientific Series in Robotics and Intelligent Systems*, 24:136–161, 2000.

[116] D. Nguyen-Tuong, M. Seeger, and J. Peters. Model learning with local Gaussian process regression. *Advanced Robotics*, 23(15):2015–2034, 2009.

[117] Duy Nguyen-Tuong. *Model Learning in Robot Control*. PhD thesis, University of Freiburg, 2011.

[118] G. niemeyer and J. Slotine. Telemanipulation with time delays. *Intl. Journal of Robotics Research*, 23(9):873–890, 2004.

[119] J. Qui nonero Candela and C. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

[120] D. Pardo, C. Angulo, S. del Moral, and A. Català. Emerging motor behaviors: Learning joint coordination in articulated mobile robots. *Neurocomputing*, 72 (16–18):3624–3630, 2009.

[121] D. Pardo, L. Rozo, G. Alenyà, and C. Torras. Dynamically consistent probabilistic model for robot motion learning. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS) - Workshop on Learning and Interaction in Haptic Robots*, pages 1–2, 2012.

[122] M. Pardowitz, S. Knoop, R. Dillmann, and R. Zollner. Incremental learning of tasks from user demonstrations, past experiences, and vocal comments. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(2):322–332, 2007.

[123] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[124] R. Peters, C. Campbell, W. Bluethmann, and E. Huber. Robonaut task learning through teleoperation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2806–2811, 2003.

[125] J. Piaget. *Equilibration of Cognitive Structures: The Central Problem of Intellectual Development*. University of Chicago Press, 1985.

[126] M. Prats, P. Sanz, and A. del Pobil. Vision-tactile-force integration and robot physical interaction. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3975–3980, 2009.

[127] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[128] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[129] M. Rahman, R. Ikeura, and K. Mizutani. Investigation of the impedance characteristic of human arm for development of robots to cooperate with humans. *JSME Intl. Journal Series C. Mechanical Systems, Machine Elements and Manufacturing*, 45(2):510–518, 2002.

[130] K. Reed and M. Peshkin. Physical collaboration of human-human and human-robot teams. *IEEE Transactions on Haptics*, 1(2):108–120, 2008.

[131] K. Reed, M. Peshkin, M. Hartmann, J. Patton, P. Vishton, and M. Grabowecky. Haptic cooperation between people, and between people and machines. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2109–2114, 2006.

[132] G. Rizzolatti, L. Fadiga, L. Fogassi, and V. Gallese. From mirror neurons to imitation: Facts and speculations. In *The imitative mind: Development, evolution and brain bases*, pages 163–182. Cambridge University Press, 2002.

[133] B. Rooks. The harmonious robot. *Industrial Robot: An Intl. Journal*, 33(3): 125–130, 2006.

[134] L. Rozo, P. Jiménez, and C. Torras. Sharpening haptic inputs for teaching a manipulation skill to a robot. In *IEEE Intl. Conf. on Applied Bionics and Biomechanis (ICABB)*, pages 370–377, 2010.

[135] L. Rozo, P. Jiménez, and C. Torras. Learning force-based robot skills from haptic demonstration. In *Intl. Conf. of the Catalan Association for Artificial Intelligence (CCIA)*, pages 331–340, 2010.

[136] L. Rozo, P. Jiménez, and C. Torras. Robot learning from demonstration in the force domain. In *Intl. Joint Conf. on Artificial Intelligence (IJCAI), Workshop on Agents Learning Interactively from Human Teachers*, pages 1–6, 2011.

[137] L. Rozo, S. Calinon, D. Caldwell, P. Jiménez, and C. Torras. Learning collaborative impedance-based robot behaviors. In *AAAI Conf. on Artificial Intelligence*, 2013.

[138] L. Rozo, P. Jiménez, and C. Torras. A robot learning from demonstration framework to perform force-based manipulation tasks. *Journal of Intelligent Service Robotics, Special Issue on Artificial Intelligence Techniques for Robotics: Sensing, Representation and Action, Part 2*, 6(1):33–51, 2013.

[139] L. Rozo, P. Jiménez, and C. Torras. Force-based robot learning of pouring skills using parametric hidden Markov models. In *IEEE-RAS Intl. Workshop on Robot Motion and Control (RoMoCo)*, 2013.

[140] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Science*, pages 233–242, 1999.

[141] S. Schaal and C. Atkeson. Constructive incremental learning from only local information. *Neural Computation*, 10:2047–2084, 1998.

[142] S. Schaal, C. Atkeson, and S. Vijayakumar. Scalable techniques from nonparametric statistics for real time robot learning. *Applied Intelligence*, 17(1):49–60, 2002.

[143] S. Schaal, A. Ijspeert, and A. Billard. Computational approaches to motor learning by imitation. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1431):537–547, 2003.

[144] M. Schneider and W. Ertel. Robot learning by demonstration with local Gaussian process regression. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 255–260, 2010.

[145] G. Schreiber, A. Stemmer, and R. Bischoff. The fast research interface for the KUKA lightweight robot. In *IEEE Intl. Conf. on Robotics and Automation (ICRA) - Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications*, pages 15–21, 2010.

[146] C. Shannon. A mathematical theory of communication. *SIGMOBILE Mobile Computing and Communications Review*, 5:3–55, 2001.

[147] A. Shon, J. Storz, and R. Rao. Towards a real-time bayesian imitation system for a humanoid robot. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2847–2852, 2007.

[148] M. Skubic and R. Volz. Acquiring robust, force-based assembly skills from human demonstration. *IEEE Transactions on Robotics and Automation*, 16(6):772–781, 2000.

[149] N. Stefanov, A. Peer, and M. Buss. Role determination in human-human interaction. In *Joint EuroHaptics Conf. and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 51–56, 2009.

[150] N. Stefanov, A. Peer, and M. Buss. Online intention recognition for computer-assisted teleoperation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 5334–5339, 2010.

[151] A. Steinfeld, T. Fong, D. Kaber, M.Lewis, J. Scholtz, A. Schultz, and M. Goodrich. Common metrics for human-robot interaction. In *ACM/IEEE Intl. Conf. on Human-Robot Interaction (HRI)*, pages 33–40, 2006.

[152] K. Strabala, M. Lee, A. Dragan, J. Forlizzi, and S. Srinivasa. Learning the communication of intent prior to physical collaboration. In *IEEE Intl. Symposium on Robot and Human Interactive Communication (Ro-Man)*, pages 968–973, 2012.

[153] W. Takano, K. Yamane, T. Sugihara, K. Yamamoto, and Y. Nakamura. Primitive communication based on motion recognition and generation with hierarchical mimesis model. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3602–3609, 2006.

[154] M. Tamosiunaite, B. Nemec, A. Ude, and F. Wörgötter. Learning to pour with a robot arm combining goal and shape learning for dynamic movement primitives. *Robotics and Autonomous Systems*, 59(11):910–922, 2011.

[155] E. Thorndike. *Animal Intelligence*. Macmillan, 1911.

[156] K. Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research*, 3:1415–1438, 2003.

[157] M. Uchiyama and K. Kitagaki. Dynamic force sensing for high-speed robot manipulation using Kalman filtering techniques. In *IEEE Intl. Conf. on Decision and Control*, pages 2147–2152, 1989.

[158] A. Ude, C. Atkeson, and M. Riley. Programming full-body movements for humanoid robots by observation. *IEEE Robotics and Autonomous Systems*, 47(2-3): 93–108, 2004.

[159] S. Vijayakumar and S. Schaal. Locally weighted projection regression : An o(n) algorithm for incremental real time learning in high dimensional space. In *Intl. Conf. on Machine Learning (ICML)*, pages 1079–1086, 2000.

[160] S. Vijayakumar, A. D'Souza, and S. Schaal. Incremental online learning in high dimensions. *Neural Computation*, 12(11):2602–2634, 2005.

[161] Z. Wang, Y. Takano, Y. Hirata, and K. Kosuge. A pushing leader based decentralized control method for cooperative object transportation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1035–1040, 2004.

[162] Z. Wang, A. Peer, and M. Buss. An HMM approach to realistic haptic human-robot interaction. In *Joint EuroHaptics Conf. and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 374–379, 2009.

[163] G. Wells and C. Torras. Assessing image features for vision-based robot positioning. *Journal of Intelligent and Robotics Systems*, 30(1):95–118, 2001.

[164] A. Wilson and A. Bobick. Parametric hidden Markov models for gesture recognition. *Pattern Analysis and Machine Intelligence*, 21(9):884–900, 1999.

[165] D. Wolpert and Z. Ghahramani. Computational principles of movement neuroscience. *Nature*, 3:1212–1207, 2000.

[166] M. Yeasin and S. Chaudhuri. Toward automatic robot programming: learning human skill from visual data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 30(1):180–185, 2000.

[167] S. Yu. Hidden semi-Markov models. *Artificial Intelligence*, 174(2):215–243, 2010.