Predictive Context-Based Adaptive Compliance for Interaction Control of Robot Manipulators

von José de Gea Fernández

Dissertation

zur Erlangung des Grades eines Doktors der
Ingenieurwissenschaften
- Dr.-Ing. -

Vorgelegt im Fachbereich 3 (Mathematik & Informatik)
der Universität Bremen
im November 2010

Datum des Promotionskolloquiums: 10.02.2011

Gutachter: Prof. Dr. Frank Kirchner (Universität Bremen)
Gutachter: Prof. Dr. Jianwei Zhang (Universität Hamburg)

GERMAN RESEARCH CENTER FOR ARTIFICIAL INTELLIGENCE
ROBOTICS INNOVATION CENTER

and

UNIVERSITY OF BREMEN
ROBOTICS GROUP

PREDICTIVE CONTEXT-BASED ADAPTIVE COMPLIANCE FOR INTERACTION CONTROL OF ROBOT MANIPULATORS

Author
José de Gea Fernández

Ingeniero Superior en Electrónica by
Universitat Politécnica de Catalunya (UPC)
Barcelona, Spain

Director
Prof. Dr. Frank Kirchner

Bremen, 2010

Zusammenfassung

Die klassische Anwendung für Industrieroboter besteht aus wiederkehrenden Bewegungsaufgaben in einem abgeschlossenen und genau beschreibbaren Umfeld. Aktuelle robotische Anwendungen orientieren sich zusehends näher an menschlichen Umgebungen. Roboter interagieren mit Menschen und werden im häuslichen Umfeld eingesetzt. Bei diesen Tätigkeiten agieren sie in einer dynamisch veränderlichen und unstrukturierten Umwelt, die teilweise oder vollständig unbekannt für sie ist. In diesem Zusammenhang ist die verlässliche Kontrolle der bei der Manipulation entstehenden Interaktionskräfte, sei es gegenüber Menschen oder Objekten, von höchster Bedeutung. Ein naheliegender Ansatz ist der Paradigmenwechsel von steifen, starken Industrierobotern, hin zu nachgiebigen und anpassungsfähigen Manipulatoren, die den menschlichen Arm zum Vorbild haben.

Die vorliegende Arbeit entwickelt eine intelligente 'high-level' Kontrolle der aktiven Nachgiebigkeit von robotischen Manipulatoren. Die Architektur für eine nachgiebige Regelung wird vorgeschlagen. Die *Predictive Context-Based Adaptive Compliance* (PCAC) besteht aus drei sich überlagernden Hauptkomponenten, die einen klassischen Impedanzregler ergänzen. Inspiriert von natürlichen Systemen, ist die höchste Stufe ein Bayesscher Kontextprädiktor, der es dem Roboter ermöglicht, die Nachgiebigkeit des Armes auf Basis von vorab Annahmen über die Umgebung anzupassen. Der Roboter aktualisiert sein Weltbild mit den während der Interaktion mit der Umwelt gesammelten Informationen und korrigiert mögliche Fehlannahmen in Echtzeit. Die Vorhersagen sichern daher die erfolgreiche Durchführung einer Bewegungsaufgabe sowohl durch eine vorweggenommene Planung als auch während ihrer Ausführung. Weiterhin wurde im Rahmen dieser Arbeit eine Komponente entwickelt, die es ermöglicht, aus den Daten der Interaktion mit der Umwelt diese anhand bereits gesammelten Weltmodellen zu identifizieren und zu klassifizieren. Auf diese Weise kann jeweils ein geeignet abgestimmter Nachgiebigkeitsregler eingesetzt werden. Die dritte Komponente der Architektur schlägt die Verwendung von neuroevolutionären Techniken zur Auswahl optimierter Parameterwerte des Interaktionskontrollers vor, sobald ein bekanntes Umfeld identifiziert ist.

Abstract

In classical industrial robotics, robots are concealed within structured and well-known environments performing highly-repetitive tasks. In contrast, current robotic applications require more direct interaction with humans, cooperating with them to achieve a common task and entering home scenarios. Above all, robots are leaving the world of certainty to work in dynamically-changing and unstructured environments that might be partially or completely unknown to them. In such environments, controlling the interaction forces that appear when a robot contacts a certain environment (be the environment an object or a person) is of utmost importance. Common sense suggests the need to leave the stiff industrial robots and move towards compliant and adaptive robot manipulators that resemble the properties of their biological counterpart, the human arm.

This thesis focuses on creating a higher level of intelligence for active compliance control methods applied to robot manipulators. This work thus proposes an architecture for compliance regulation named *Predictive Context-Based Adaptive Compliance* (PCAC) which is composed of three main components operating around a 'classical' impedance controller. Inspired by biological systems, the highest-level component is a Bayesian-based context predictor that allows the robot to pre-regulate the arm compliance based on predictions about the context the robot is placed in. The robot can use the information obtained while contacting the environment to update its context predictions and, in case it is necessary, to correct in real time for wrongly predicted contexts. Thus, the predictions are used both for anticipating actions to be taken 'before' proceeding with a task as well as for applying real-time corrective measures 'during' the execution of a in order to ensure a successful performance. Additionally, this thesis investigates a second component to identify the current environment among a set of known environments. This in turn allows the robot to select the proper compliance controller. The third component of the architecture presents the use of neuroevolutionary techniques for selecting the optimal parameters of the interaction controller once a certain environment has been identified.

Resumen

En la considerada robótica industrial clásica, los robots se encuentran recluídos en entornos perfectamente conocidos y estructurados, donde realizan tareas puramente repetitivas. Por contra, la robótica actual tiende a acercarse a entornos humanos, cooperando con ellos para realizar una tarea y entrando en entornos domésticos. Pero sobretodo, la robótica está abandonando el mundo de certidumbre para trabajar en entornos desestructurados y que cambian constante y dinámicamente, siendo muchas veces parcial o totalmente desconocidos para los robots. En éste área, el control de las fuerzas que aparecen cuando un robot entra en contacto con el entorno (sea este entorno un objeto o un humano) es de crucial importancia. El sentido común apunta a dejar de lado el uso de los rígidos robots industriales y dirigirse hacia robots manipuladores que presenten la adaptabilidad y acomodación [1] observada en su homólogo biológico, el brazo humano.

Esta tesis se centra en dotar de un nivel superior de inteligencia a métodos de control activo de la impedancia (o más generalmente, de acomodación) de robots manipuladores. Este trabajo propone una arquitectura para la regulación de la acomodación denominada *Predictive Context-Based Adaptive Compliance* (PCAC), que se compone de tres componentes básicos operando alrededor de un controlador de impedancia 'clásico'. Inspirado por el funcionamiento de los sistemas biológicos, el componente de más alto nivel es un módulo de predicción de contexto basado en modelos Bayesianos que posibilita que el robot pre-regule la impedancia del brazo robótico en base a los resultados de la predicción del contexto en el que se encuentra inmerso. El robot puede utilizar esta información para actualizar sus predicciones mientras se contacta con el entorno, de forma que si fuese necesario, podría corregir en tiempo real su respuesta ante una falsa predicción. Es decir, las predicciones se utilizan tanto para anticipar las acciones a llevar a cabo 'antes' de proceder con la tarea, así como para aplicar correcciones en tiempo real 'durante' la ejecución de ésta, de forma que se asegure su éxito. Adicionalmente, esta tesis investiga un segundo componente que identifica el entorno contactado dentro de una serie de entornos conocidos lo que permite al robot seleccionar el controlador de impedancia más adecuado. El tercer componente de la arquitectura propone el uso de técnicas neuroevolutivas para la selección de los parámetros óptimos que definen el control de impedancia una vez se ha identificado un entorno concreto.

[1] más comúnmente conocido por el vocablo inglés *compliance*

Acknowledgments

This work has only been possible with the help, voluntary or involuntary, conscious or unconscious, of so many people over the years that I fear I will certainly forget someone. I apologise in advance if that is the case.

First of all, I would like to thank Prof. Dr. Frank Kirchner for giving me the opportunity of working at his Lab and on this thesis, be patient enough to let it reach this point, and facilitate its finishing process. Similarly, I would like to thank all the people that are or were at some point working at our Lab, both research staff and administration. All of them somehow influenced this work and made it possible by providing a nice working atmosphere.

Special thanks to Florian Cordes and Thomas Röhr for their useful comments after reading a draft of this work. To Mark Edgington for so many hours of interesting conversations and his valuable amendments. To Dr. Jan Albiez for his willingness to help and provide advices for finishing up this thesis as well as proofreading it. To Dr. Johannes Lemburg for his constant availability to provide assistance with any mechanical or electrical issue concerning the robots. To Dr. Yohannes Kassahun, with whom I spent countless hours discussing about science and life and from whom I learnt many important things for my professional and personal life.

If this thesis reached the point of being printed on paper, my good friend and great researcher Dr. Carlos Pérez is, without any doubt, a person I have to thank. He did the most over the years to encourage me, to revise my work, to criticise it when necessary, and to untiringly help in anything within his reach.

On the personal side, I cannot express with words all my gratitude to my parents for being the real reason that I am here now and doing what I do. They worked hard to provide me with an education that would ensure me a life that they did not have. Unfortunately, and for their despair, I achieved it far away from them. Finally, the most important person to acknowledge is my partner Maria. Her continuous encouragement over the last years, her love, and her support in the difficult moments were crucial ingredients that made this work possible. Without her presence, this thesis would certainly not be here.

José de Gea Fernández   Bremen, 5th of November 2010

To my parents

Contents

Part I

Thesis

Chapter 1

INTRODUCTION

This chapter serves as introduction to this work and describes its motivation, scope, and goals. Moreover, the chapter shows the structure of this document and summarises the most relevant contributions of this work to the state of the art.

1.1 Interacting with the world

Industrial robots have been present in factories for almost 50 years. In 1961 the Unimate robot was deployed at a General Motors plant, following the invention from George Devol in cooperation with Joseph Engelberger in 1956. Ever since, the use of robots has been steadily increasing, especially in the automotive industry, which accounts for almost 60% of the total robot sales [1]. But there are several potential markets in which robots are not yet introduced, mainly due to the challenging requirements and the high costs involved. Nowadays, robots are mainly deployed in large-volume manufacturing scenarios, where tasks are very repetitive, and whose environments are strictly controlled. However, there are many other potential scenarios where robots could be used but their deployment is not only economically unreasonable but also technically very difficult or even impossible. Any scenario outside a well-defined robotic workcell, or in other words, any environment that is not modified and is prepared for deploying a robot, is still nowadays a challenge. Possible scenarios range from logistics, nuclear, and search-and-rescue scenarios to household environments. In these situations, a robot is required to deal with a mostly unknown, dynamically changing environment as well as with variations on the properties and geometry of the objects to handle. The robot cannot rely on pre-defined plans and strategies, but needs to process the available sensory information in order to adapt to the current situation. Classical control techniques must account for it and be enhanced in order to cope with these new challenges.

The use of visual information for object recognition, 3-D pose estimation, and visual tracking is probably the area most advanced in terms of using external sensors to help the robot to overcome environment uncertainties and changes. Another emerging area, as evidenced by the recent industrial interest and number of applications, is the use of force sensors mounted on the robot manipulators in order to regulate the interaction forces between the robot and the environment. Nowadays, most of the industrial robot manufacturers include force sensors as additional equipment for their robots, including user-friendly interfaces in order to set up the desired forces for specific tasks. These are enhancements over the previous state of the art in industrial robots but assume and require knowledge about the robotic workcell. Robots can be easily reprogrammed for new tasks but they are not able to deal with unknown situations.

In the area of force control, researchers have been working for the last 30 years on incorporating force control strategies, originally aiming at limiting the maximum forces that large robot manipulators could exert in an environment. However, as previously mentioned, it has been only in the last couple of years that the important industrial manufacturers have been including force sensors in their portfolios. So far, companies needed to look for tailor-made solutions via subcontracted engineering companies. Needless to say, there is, of course, a substantial gap between the technology used nowadays by the industry and what research centers are working on. From the viewpoint of a research center, it would seem that force control strategies are long a closed research topic. Nowadays, research centers and university departments concentrate on more elaborate interaction control strategies like impedance control [50], which tries to convert a stiff industrial robot manipulator into a compliant one by using feedback control techniques. On the other hand, from the viewpoint of the industry, force control is today's latest enhancement. This comes at no surprise, as the industry-research gap is well-known in many areas, though it is especially significant in robotics.

There are several reasons that might be mentioned in order to explain why the industry lags so far behind research in industrial robotics. One of them is that classical techniques of interaction control (i.e. the use of passive elements attached to the robot's end-effector or simple force control schemes) provide a more stable control and well-known behaviour. The price one pays for theses qualities is a lack of flexibility and adaptivity of the solution to changing environmental conditions. But there is no reason whatsoever to adopt a very complex and adaptive strategy if the environment is not going to change. For current manufacturing scenarios, there is no room for adaptive controllers in this sense (there are, of course, adaptive controllers deployed in industry, for instance, where the parameters of the plant might change over time). However, there are several new and currently interesting areas where robotics is barely present: logistics, home, food industry, etc. Each of them presents specific challenges but all of them share common elements: unknown and changing environments that are difficult to model, and objects to be manipulated that might have unknown properties (in terms of weight, being deformable, etc ...). Specific examples are, for instance, a coffee sack at a harbour container or a sticky and delicate pastry in the food industry. Both are examples of tasks that, despite the tremendous development of robotics in the last decades, still require human handling.

In this thesis, neuroscience-based principles and mechanisms underlying biological motor control are investigated as a possible source of inspiration for the development of new control principles for robotic manipulators. The ultimate aim is to develop a robot with increased performance and adaptation skills, especially in real-world and unstructured environments. The reason to look back to biology appeared to be clear: no artificial machine had shown so far adaptation skills similar or even close to the ones shown by biological systems. Despite the increasing interest on providing robots with complex manipulation capabilities and, acknowledging the fact that some very promising and spectacular results have been reported [98][4], it is a truism that we are still far from reproducing the manipulation skills of a three-year old child. The test platform in this thesis is a dual-arm robotic manipulator and, accordingly, dual-arm manipulation tasks are the ultimate aim of the control schemes proposed. In neuroscience, as we will see in subsequent chapters, there are three theoretical frameworks to explain the bimanual coordination mechanisms used in biological systems: (1) the use of classical theories on behaviour of dynamical systems [71], (2) the use of information processing theories [121], and (3) the current neuroscience hypotheses regarding the use of internal models which are reformulated for explaining bimanual operations [134]. The last of these approaches postulates the use of internal models by the Central Nervous System (CNS): forward models that relate sensory information to a given motor command [129] and inverse models that relate motor commands to a given sensory information [136]. Recently, these ideas were used in some theoretical computational models, and an architecture was proposed in which such internal models form the lowest layer of a multilayer control system [135].

Inspired by these theoretical computational models based on a multilayer control system, and driven by the need for higher levels of intelligence and adaptivity in robotic systems, this work proposes a predictive context-based adaptive compliance controller for interaction control of industrial-like robot manipulators. At its core, this is a multi-instance and adaptive impedance controller, whose Bayesian-based context-based predictions make use of sensory forward models to provide flexibility and adaptivity in the current scenario.

1.2 Goal of the thesis

The goal of this thesis is

> *to develop a Bayesian-based high-level intelligent architecture to control the interaction forces appearing on a robot manipulator through contact with the environment.*

The proposed control architecture is called 'Predictive Context-based Adaptive Compliance' (PCAC). The original aim is to control interaction forces and create a more flexible robot manipulator by incorporating higher control layers of intelligence. It uses an impedance controller as its core element to create a compliant manipulator which can be used on current industrial robots, that is, without internally modifying the existing joint controllers. After an initial study of impedance controllers, three observations arose that shaped the subgoals of this work:

First Knowledge about the environment is crucial for the successful use of an impedance controller in a real scenario. That is, the environment needs to be known or under control, which contradicts the original ideal that the impedance control would compensate for the inaccuracies of robot and environment models. The first subgoal, therefore, is to develop a Bayesian-based estimator that, similarly to biological systems, uses the first contact to gather information for identifying the environment and using it in subsequent contacts.

Second The selection of the impedance controller's parameters can be complex and cumbersome, especially if the environment is not precisely known. The second subgoal, therefore, is to investigate neuroevolutionary techniques to provide a clear and simple methodology for selecting optimal impedance parameters.

Third A higher level of intelligence is required that provides the system with more flexibility and adaptivity to uncertain scenarios. The final subgoal is to develop a Bayesian-based method that predicts the current context. This extra information will allow the robot to anticipate the consequences of self-generated actions over certain environments. Moreover, this information can be used to adapt and immediately apply corrective measures in the case of a wrongly-estimated environment.

1.3 Structure of the thesis

Figure 1.1 shows the structure of the thesis grouped into general topics. As it can be seen, the main contributions are from Chapters 4, 5, 6, and 7. Chapters 2 and

3 are introductory and serve as a basis for developing the proposed architecture over several chapters of this thesis. Chapter 2 introduces classical control methods for interaction control, focusing especially on active impedance control schemes. Chapter 3 introduces neuroscience principles for motor control as well as biologically-inspired control methods that might be used for interaction control. Chapter 4 is a short chapter that introduces the architecture proposed in the thesis and its main subcomponents for the intelligent control of the interaction of robot manipulators. Chapter 5 presents the methodology used for selecting the most suitable impedance controller based on environment identification methods using Bayesian inference. This chapter validates the impedance control scheme in simulation whereas the environment identification methods are experimentally validated. Chapter 6 presents the use of evolutionary strategies to evolve a robust force-tracking optimal interaction controller for a specific environment. Finally, Chapter 7 develops a Bayesian-based context predictor that, together with the previously mentioned contributions, is applied within the proposed architecture to control a real robotic system: a novel autonomous dual-arm robot manipulator. Chapter 8 concludes this thesis with the final conclusions and outlook.

1.4 Frame of the thesis

The temporal span of this work embraces its start around autumn of 2006, when I came across the concept of 'impedance control', at that time with a position at the Robotics Group of the University of Bremen (Germany). Its finalisation occurred within the Robotics Innovation Center (RIC) of the German Research Center for Artificial Intelligence (DFKI) by the end of 2010.

During the realisation of this work, several robotic platforms were used at different stages of the development. Figure 1.2 shows the robot manipulators used during this time. Firstly, experiments with the basic impedance controller and the estimation of the properties of the environment (Chapter 5) were performed using the seven degrees-of-freedom Mitsubishi PA-10 robot. The experiments and results presented in Chapter 7 made use mainly of 'Mr. SemProM', a dual-arm system built at DFKI using two Schunk robot arms. At the final stages of writing down this thesis, the results are being implemented on the robot AILA, an android designed and built at DFKI.

1.5 Contributions

This thesis is supported by a number of peer-reviewed publications for each of the main chapters and related to the previously mentioned goals of this thesis. The first subgoal led to the development of a Bayesian inference method that allows the robot to estimate the most likely current environment in order to adapt the controller accordingly. Chapter 5 deals with this problem and generated two publications. Of special importance is the first publication, which presents a robust methodology to identify the properties of the contact impedance, and thus allowing a safer and reliable contact. The second publication describes the modeling and simulation of robot interaction forces using an impedance control system:

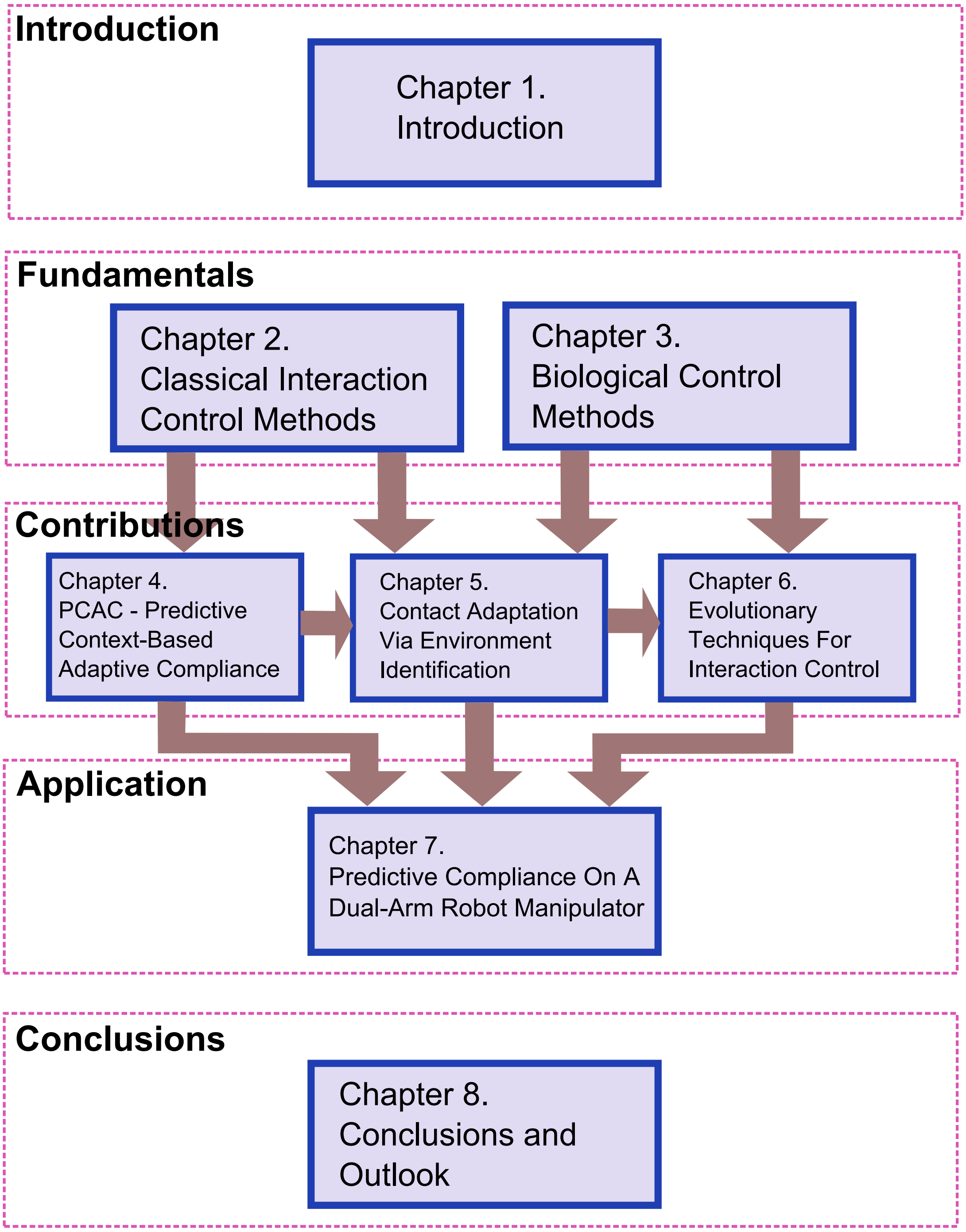


Figure 1.1: Structure of the thesis

- J. de Gea and F. Kirchner. *Contact impedance adaptation via environment identification.* In Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE08), pages 1365–1370, Cambridge, UK, June, 2008. [27]

- J. de Gea and F. Kirchner. *Modelling and simulation of robot arm interaction forces using impedance control.* In Proceedings of the 17th World Congress

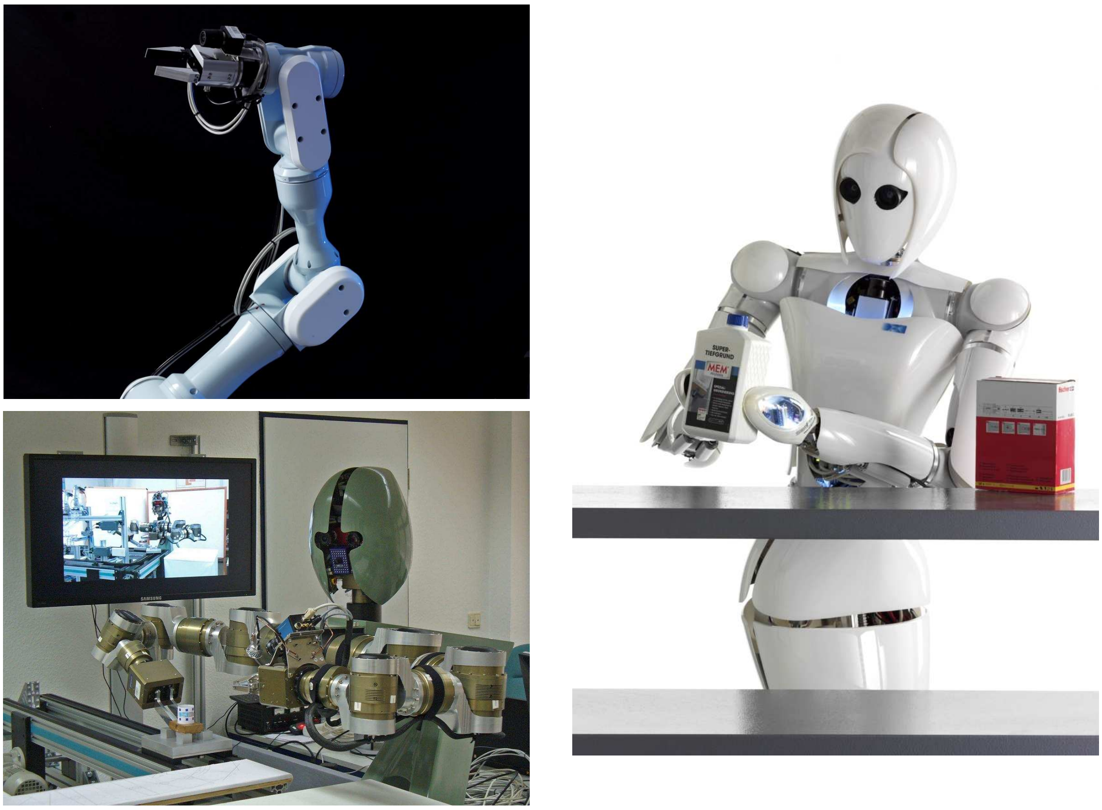


Figure 1.2: Robot manipulators used for the experiments described in this thesis: *top left:* Mitsubishi PA-10, *bottom left:* Dual-arm system built using Schunk modules, *right:* AILA

The International Federation of Automatic Control (IFAC), pages 15589–15594, Seoul, Korea, July 6-1, 2008. [28].

The second subgoal of the thesis led to the investigation of neuroevolutionary techniques in order to evolve impedance controllers with the desired properties. Chapter 6 deals with this topic and is supported by three publications where evolutionary techniques were used for designing an optimal force-tracking impedance controller:

- J. de Gea, Y. Kassahun, and F. Kirchner. Book 'Factory Automation', chapter *Control of Robot Interaction Forces Using Evolutionary Techniques*, pages 445-462, In-Tech, 2009. [25]

- J. de Gea, Y. Kassahun, and F. Kirchner. *On evolving a robust force-tracking neural network-based impedance controller*. In 40th International Symposium on Robotics (ISR'09),Barcelona, Spain, pages 127–132, 2009. [26]

- J. de Gea and F. Kirchner. *Using neuroevolution for optimal impedance control.* In IEEE International Conference on Emerging Technologies and Factory Automation (ETFA-2008), pages 1063–1066, Hamburg, Germany, September 15-18, 2008. [29]

The third subgoal led to the development of Bayesian-based context predictor for anticipating sensory consequences of robot actions and applying corrective measures in the case of wrongly-estimated contexts. Chapter 7 deals with this topic and has generated so far one publication describing the main idea of the role of the Bayesian context predictor on regulating compliance:

- J. de Gea. *The role of prediction in compliance adaptation.* In Robotics: Science and Systems, Workshop on Strategies and Evaluation for Mobile Manipulation in Household Environments (RSS2010), Zaragoza, Spain, 2010. [24]

Finally, the dual-arm robotic system used for the final experiments of this thesis is described in:

- J. de Gea, J. Lemburg, T.M. Roehr, M. Wirkus, I. Gurov, and F. Kirchner. *Design and control of an intelligent dual-arm manipulator for fault-recovery in a production scenario.* In IEEE International Conference on Emerging Technologies and Factory Automation (ETFA-2009), Mallorca, Spain, September 22-26, 2009. [30].

Chapter 2

CLASSICAL INTERACTION CONTROL METHODS

This introductory chapter describes classical interaction control methods used to regulate the forces that appear when a robot contacts the environment. A first division is made between passive and active control methods, from where the focus of the chapter continues towards active methods, especially active impedance control schemes.

2.1 Introduction

Traditionally, robots used to perform repetitive tasks for which the performance figures were the maximum bandwidth, precision, and workspace. In these typical and classical industrial scenarios, the environments are strictly under control as well as the robot tasks. Position (or more generally speaking, motion) control is the best solution as long as no (or little) contact with the environment is necessary. However, in case of unstructured environments, where the knowledge of the environment is imprecise, or in complex motion-constrained tasks, some kind of force control is required in order for the robot to attain its goal. More specifically, robots need to sense and, by sensing, comply with the current environment, regardless of its unknown nature.

One of the required components for a robust and adaptive robot is the ability to adapt the contact interaction forces when manipulating an object or contacting a surface (in this area, any object/surface with which the robot makes contact is generally defined as 'environment'). This problem domain is generally known as compliance control and tries to guarantee that the robot accommodates the interaction forces rather than resisting to the constraints posed by the contact with the environment. Figure 2.1 shows a general overview of classical compliance control methods. The control methods can be divided into passive or active methods:

- passive methods are those on which physical elements are included between the robot and the environment in order to control and limit the forces whereas

- active methods are those which make use of feedback control techniques in order to regulate force and/or motion of the robot when contacting the environment

In essence, both methods pursue the same exact goals: comply with the interaction forces generated by the contact with the environment. Notably, active methods can tackle a broader range of applications and their adaption to different scenarios is simpler and faster, although their initial design is much more complex than passive methods. Although the active compliance control methods are already quite established on research laboratories, they have not found their niche in industry. Basically, the industry solves the same problem by, on the one hand, structuring the work environment and, on the other hand, by adding mechanical elements to achieve passive compliance. The reason is that those methods provide a more stable control. The price to pay is the lack of flexibility and adaptivity of the solution to changing environmental conditions.

Ideally, the best actuator for performing force control would be the one that is a perfect force source, that is, the one that outputs the exact same force as commanded, regardless of the load. Needless to say, real-world actuators are far from an ideal force source. For instance, the load movement will create an additional force at the output of the actuator: the actuator's impedance. As we know from electrical engineering, impedance increases with frequency (i.e. with load motion). Another important indicator of the robot's performance is the bandwidth, which specifies the maximum frequency up to which the forces will be precisely tracked. By definition, an ideal force source has zero output impedance (thus is backdriveable) and has infinite bandwidth.

Thus, in order to succeed in compliance control, there are basically two options: either a compliant actuator is used, one that resembles as close as possible an ideal force source (low output impedance, high bandwidth) or, as a second alternative, an active control system is used that tries to provide a geared electrical motor (a very bad force source) with the properties of an ideal force source by actively controlling and thus lowering its output impedance. The next sections present the alternatives available for the first choice (passive compliance control) and with special focus especially on the second (active compliance control).

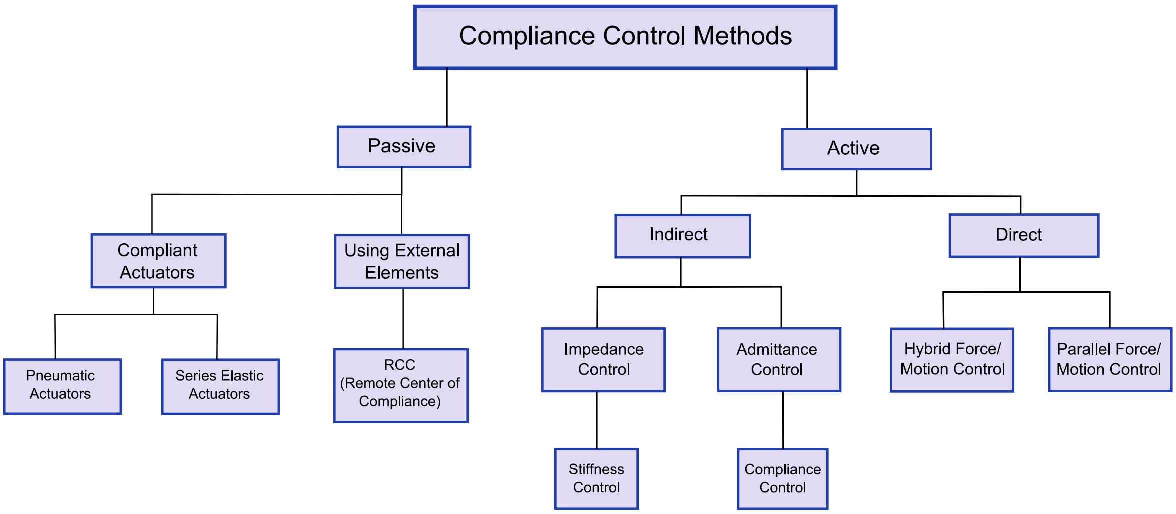


Figure 2.1: Classification of classical compliance control methods

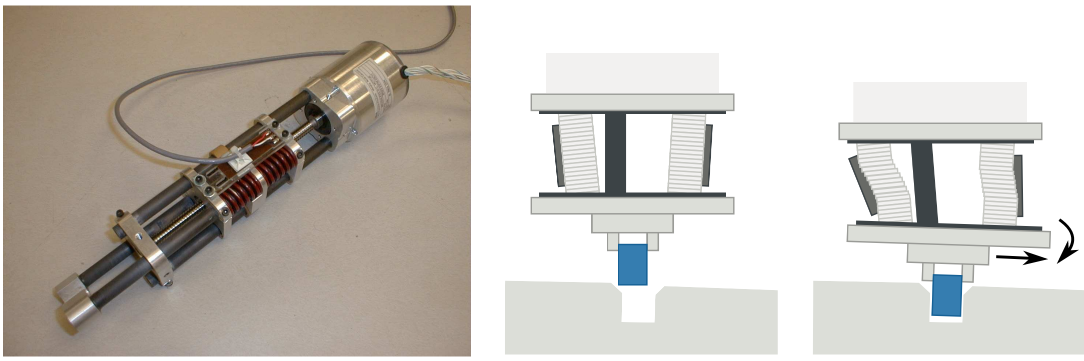
2.2 Passive Control Methods

Passive compliance by using compliant actuators can be used to obtain low mechanical output impedance. The use of compliant actuators provides basically two advantages over non-compliant ones. On the one hand, the impedance at frequencies higher than the control bandwidth is determined by the compliant element of the actuator itself, whereas in a non-compliant actuator the impedance outside the control bandwidth depends on the reflected motor inertia. In case of using gears, the reflected inertia equals the motor inertia times the square of the gear ratio. Note that usually the gear ratio in industrial robots is quite high. On the other hand, compliant actuators have a reasonably good force tracking bandwidth and are able to deal with perturbations by means of the physical compliant element. Of course, they lack the positioning accuracy of stiff actuators. Some of the currently used compliant actuators for robotic applications are pneumatic actuators and series elastic actuators:

- Pneumatic actuator. Pneumatic actuators are interesting because the possibility to compress the air limits the forces exerted. Thus, they ensure soft collisions due to this intrinsical compliance. Furthermore, they can provide accurate force control. Yet more interesting, force and compliance can be regulated independently.

The actuator's force is determined by the difference in pressure between the two cylinder chambers and the compliance is determined by the compression of the air. That also means that it is possible to achieve a high-bandwidth force control if a high-bandwidth control for the differential pressures is used, and that without the intrinsical compliance being modified. A number of examples are found where pneumatic actuation is used to build robot manipulators. Recently, Festo presented the Airic's arm that use their 'Fluidic Muscles' (Fig. 2.3(a)).

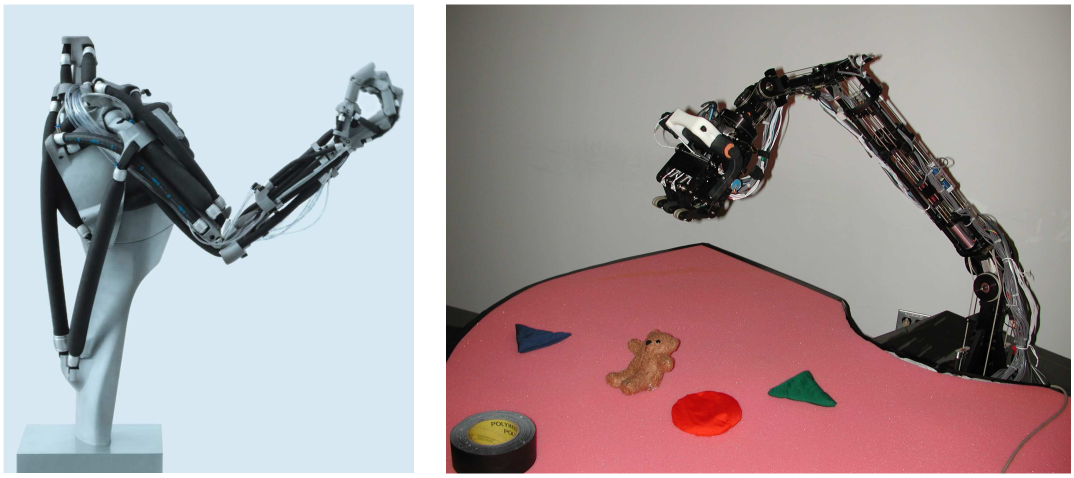
- Series elastic actuator [102]. The basic idea is to include an elastic element (usually a spring) between motor and load as seen in Fig. 2.2(a). Series Elastic Actuators were developed at the MIT Leg Lab and were patented by Gill Pratt and Matt Williamson (US Patent 5650704). The company Yobotics (spin-off from MIT) is now commercialising and managing the license. The Series Elastic Actuator works by measuring the compression of the spring, so that the force applied to the load can be easily determined and modified to track a desired reference. The main result of including the elastic element is a much lower output impedance of the actuator and, to some extent, back-driveability. Moreover, the impedance at high frequencies is limited to the stiffness of the spring, improving intrinsical safety by diminishing the effects of a collision (as we mentioned before, without elastic element, the motor reflected inertia is the gear ratio times the motor inertia and increases with frequency). Mainly robots at MIT Leg Lab used this technology: the walking bipeds 'Spring Turkey'[101] and 'Spring Flamingo'[103], the arms of the humanoid 'Cog'[12], as well as the robots at Yobotics, e.g. the Agile Robot Arm (Fig. 2.3(b)).



(a) (b)

Figure 2.2: (a) Series Elastic Actuator from Yobotics, Inc., (b) Remote Center of Compliance (RCC) device during a peg-in-a-hole task *(Creative Commons image)*

The previous compliant actuators can be used to design intrinsically safe robots that account and accommodate for interaction forces. However, this solution cannot be applied everywhere, for instance, in industrial settings, where robots are built using electrical motors, which are not intrinsically compliant. These robots are especially designed to be as stiff as possible, since positioning accuracy is directly and proportionally related to the stiffness. In those cases, a second alternative is the use of an external



(a) (b)

Figure 2.3: (a) Airic's arm from Festo AG using pneumatic artificial muscles, (b) Agile Robot Arm from Yobotics, Inc. used here on the robot Ripley (MIT Media Lab)

device attached to the robot's end-effector, which provides with passive compliance to the robot. This device is known as Remote Center of Compliance (RCC) [133] (Fig. 2.2(b)) and it is specially designed for every task where compliance is required (like in the peg-in-a-hole task). The mechanical device deals with inaccuracies on the position/orientation of the pre-programmed trajectories and, due to its mechanical design, re-adjusts itself to succeed on the interaction task. It is a cheap and simple solution for industrial peg-in-a-hole tasks since it does not require any extra sensors or control system. Its main disadvantage lies in the need of specifically designing it for each individual task and, even in that case, it can only deal with small deviations around the desired trajectories.

2.3 Active Control Methods

Active control methods use measurements from the contact forces (and moments) and robot's motion, which are fed back to the robot's controller in order to generate appropriate motion commands according to a desired robot's behaviour. Obviously, active control methods can provide the robot controller with a higher degree of flexibility, yet at the cost of being slightly slower on the reaction times. Since these methods are based on sensory feedback signals, which precisely originate only after a contact has occurred, they cannot guarantee safe contact interaction in all situations. For certain scenarios, active control methods might need to be used in combination with passive compliance methods that intervene at the first stages of the contact.

The research on active methods to provide robust force control in any of its flavours has been gaining importance in the last three decades and a great deal of research papers is available. A first description of the state of the art on the 80's can be found at [132]. Similarly, [31] reviews the state of the art on the 90's. Some books also appeared at that time that focused exclusively on force control [116]. More recently, [83] surveys the state-of-the-art of the required subcomponents for active compliant systems. The recent 'Handbook of Robotics' [115] includes also a chapter that reviews the current state of robot force control. Generally speaking, any active control method for compliance control will try to somehow add or combine motion and force errors, and use a controller or set of controllers to send the most proper commands to the robot's joint actuators. The way these errors are combined is what creates the basic distinction between direct and indirect compliance control methods.

> Direct force control methods are those where the controller directly regulates the contact force to a desired reference. A classical force control strategy or a hybrid force/motion control belong to this category.
>
> Indirect force control methods are those where the force is controlled indirectly via motion control. Impedance control in its different 'flavours' belongs to this category.

Both approaches differ on the way to specify the interaction task. For instance, in the hybrid force/motion control (direct method), the task is specified in the geometric space, as we will see later: the user defines which directions will be controlled on force and which ones will be controlled on position. In the case of impedance control (indirect method), the designer will define a dynamic relationship between force and motion. Ultimately, the set of impedance parameters defined will determine the robot's behaviour. When comparing the design methodologies for compliance control in a practical sense, both are quite similar: where a designer would identify a constrained direction in hybrid force/motion control, he would probably define a more compliant behaviour when using an impedance controller; where a designer would identify an unconstraint direction in hybrid force/motion control, he would probably define a stiffer behaviour when using an impedance controller. Interaction control methods might be also classified according to the static or dynamic performance of the control method:

> Dynamic model-based control methods are those that are concerned with the transient, i.e. with the dynamical response of the system. To this category belong impedance and admittance control schemes, as well as hybrid and parallel force/motion strategies. For any of them, a complete dynamic model of the robot is required, thus being more complex, both to be designed and implemented. Furthermore, force measurements are necessary in order to obtain a decoupled and linear interaction model that is easily tractable.
>
> Static model-based control methods are those that are only concerned with the steady-state response of the system. In the case of impedance control, the static case is called stiffness control. In the case of admittance control, the static case is known as compliance control. That is, compliance and stiffness control are subsets of admittance and impedance control, respectively. These methods are thus easier to implement, as they do not require dynamic models but only knowledge about the gravity terms.

A review of several interaction control methods, divided into dynamic and static model-based methods, is given in [18] that also includes experimental evaluation.

2.3.1 Direct Control Methods

Direct control methods require explicit environment models and the task to be solved, since they rely on the classification of degrees of freedom that are to be controlled in position and those controlled in force. Thus, the task and the environment need to be perfectly known so as to design the proper controller.

2.3.1.1 Hybrid Force/Motion Control

This method [106] treats the contact interaction as a geometric problem, where there are a set of geometric constraints to be taken into account. After careful examination of the interaction task, a number of robot's degrees of freedom are regarded as 'force-controlled', whereas the rest are considered to be 'motion-controlled'. That means that the hybrid controller will control exclusively motion along unconstrained, 'motion-controlled', directions, and force/moment along constrained, 'force-controlled', directions. This approach is based on the assumption that for most of the usual constrained robotic tasks, it is possible to split the task into two mutually independent subspaces, one controlling contact forces and one controlling robot's motion. Thus, making use of different subspaces, motion and force can be controlled simultaneously. Fig. 2.4(a) shows a diagram of a hybrid force/motion controller. In this diagram, as well as for the diagrams that follow in the next sections, the following symbols are used:

> The end-effector position and orientation is denoted as $\boldsymbol{X}$. This 6×1 vector is defined as $\boldsymbol{X} = (\boldsymbol{p}^T \boldsymbol{\varphi}^T)^T$, where the vector $\boldsymbol{p}$ describes the end-effector position and $\boldsymbol{\varphi}$ is a set of Euler Angles from the rotation matrix describing the orientation of the end-effector.
>
> The three-dimensional force $\boldsymbol{f}$ as well as the the three-dimensional moment $\boldsymbol{m}$ exerted by the robot's end-effector are the components of the wrench $\boldsymbol{h} = (\boldsymbol{f}^T \boldsymbol{m}^T)^T$.
>
> The commanded torques are denoted by $\boldsymbol{\tau}$.
>
> The subscript d denotes a 'desired' reference value for the specific magnitude (either force or position/orientation in this case).

2.3.1.2 Parallel Force/Motion Control

This approach [17] (Fig. 2.4(b)) is classified as direct control method as it starts with the geometric constraints also used for hybrid force/motion control. Note that the diagram is similar to that of Admittance Control (Fig. 2.7(a)), but in this case, the parallel force/motion approach requires knowledge of the interaction task in order to define the geometric constraints. Thus, the difference with admittance control is

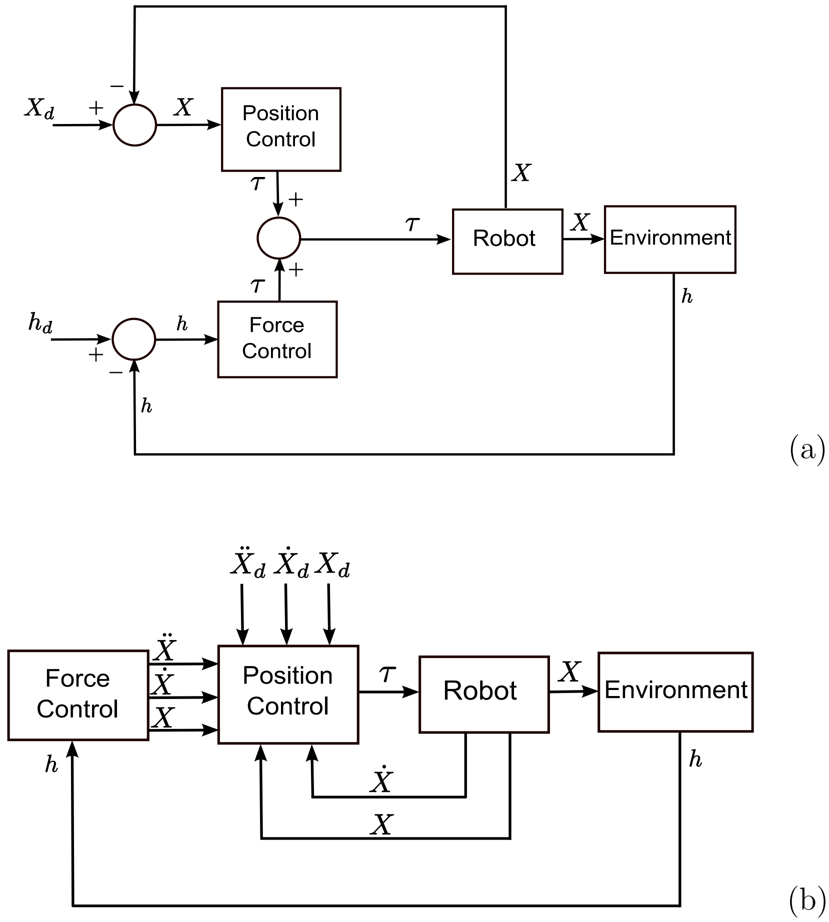


Figure 2.4: (a) Hybrid force/motion control, (b) parallel force/motion control

the criteria to define the force controller (physical impedance parameters versus force-controlled task directions). The difference with a hybrid force/motion approach is that parallel force/motion control does not use different control subspaces for force and motion, but combines and weights the contributions of motion and force controllers into one controller using a single matrix. In this case, the controller gives priority to force errors, which dominate the controller's response. Thus, a position error would be 'tolerated' along a constrained direction in order to ensure proper force tracking.

2.3.2 Indirect Control Methods

The general term impedance control is commonly used indistinctly when referring to either impedance or admittance control. Both pursue the same goal -active modification of the mechanical impedance of the robot- but they do it from different perspectives. Impedance control basically works by measuring position and outputting force, whereas admittance control works by measuring force and outputting position. Due to the different ways of solving the control problem, the accuracy of the compliance method will lie on different factors: the impedance control depends on the accuracy of the position sensors and the bandwidth and accuracy of the force-controlled actuators, whereas in admittance control the accuracy depends on the force sensors used and the bandwidth and accuracy of the position-controlled actuators.

2.3.2.1 Impedance Control

Active impedance control (Figure 2.5(a)) indirectly regulates the contact forces by generating an appropriate motion that ends up in a desired dynamic relationship between the robot and the environment. In contrast to hybrid force/control methods, impedance control uses a single control law to regulate simultaneously both position and force by specifying a target dynamic relationship between them. In other words, it weights the contributions of the force and motion controllers using a set of weighting matrices. In contrast to hybrid force/control where there are also weighting matrices, the matrices used in active impedance control have physical dimensions of impedance: that is, stiffness, damping, and inertia parameters. These parameters will thus shape the dynamical behaviour of the robot as if mechanical springs, dampers, and extra inertia were included into the robot's end-effector. The design of the target impedance that ensures a proper behaviour is not an easy task and one of the aims of this thesis was to provide a clearer methodology (Chapter 6). It is clear to see that the behaviour of the robot when contacting an environment needs to be different than when moving freely, especially when a good position tracking is desired in free space (remember that a stiff robot is a good position tracker, whereas by definition a stiff robot will not be compliant when contacting the environment). Moreover, even a well-defined target impedance depends finally on the dynamics of the environment, that need to be estimated as good as necessary to ensure not only stability on a first level, but also the necessary robot's performance.

The work from Neville Hogan [50] is considered the reference paper about impedance control as he was the first to describe the concept of a virtual mechanical impedance for controlling robot-environment interaction forces. However, it was not the first idea on the use of virtual mechanical elements to control forces: in [97] a generalized spring and damper system for force control is presented that is later implemented in [131]. One of the best current examples of the use of active impedance control are the LWR lightweight arms from DLR [2](Figure 2.6). Ever since a vast literature has been published around the topic in order to cope with pitfalls of the original description or deal with implementation issues. For instance, Hogan itself [53] and later others authors [120] analysed the stability of the impedance control law or the instability originated after contacting stiff environments [70]. In order to tackle the problem of the uncertainty on the parameters of the environment model as well as of the robot, some works propose adaptive [111], [21], [117], [89] or robust [87], [63] impedance control strategies that would deal with uncertainties. Moreover, as the original description lacks of force tracking capabilities, some works propose methods for enhancing the controller to track forces [112], [64]. The impedance control area has seen as well the use of neural network implementations of the controller [62], [65], in most cases used as a method to minimise the problems arising from the model's uncertainties. Learning algorithms have been also proposed [16], [19], as well as impedance control methods have been not only applied to single-arm systems but also to multiple-arm systems [10], [85], [94].

From the literature survey it is clear that active impedance control requires basically two components: a reliable estimation of the environment properties as well as a controller that executes the motion commands according to the information provided by feedback and estimated models in a robust and optimal manner. The first objective has been also a contribution of the thesis (see Chapter 4 for a general description, and Chapter 5 for details) by providing a robust Bayesian-based identification algorithm that estimates the probabilities of a given environment in order to allow the impedance controller to adapt its properties precisely to the nature of the current environment.

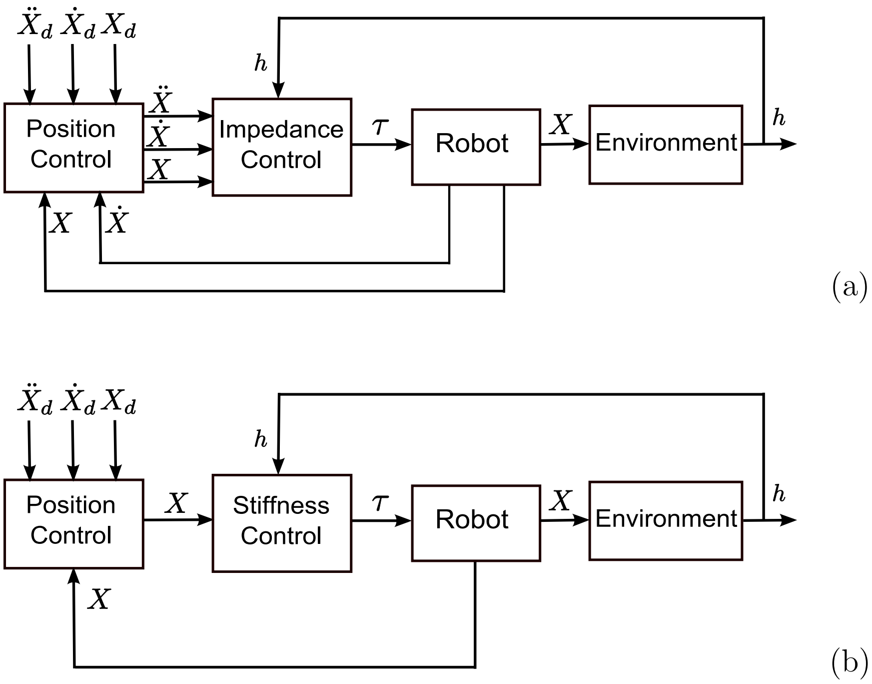


Figure 2.5: (a) impedance control, (b) stiffness control

Stiffness Control Stiffness control is only concerned with the steady state of the end-effector, thus this is a simplification of a complete impedance control. Only a proportional action (a single matrix) is necessary to define the behaviour of the controller, which will control the behaviour of the robot in such a way that behaves as a six-degree-of-freedom spring with respect to the forces (and moments) applied to the robot's end-effector. Since this method focuses specifically on the steady-state response, it does not require knowledge about the complete robot's dynamics. In this case, only the gravity terms of the Lagrangian equation representing the dynamics of the system is necessary. In brief, the stiffness control regulates the static relationship between the forces exerted on the environment and the deviation (if and as much as necessary) of the position and orientation from the desired values. Figure 2.5(b) shows a diagram sketching the stiffness control that is not more than an impedance control that only receives information about position, since it will not deal with higher derivates of this variable.

Figure 2.6: LWRIII lightweight arm from DLR

2.3.2.2 Admittance Control

Admittance control holds basically the same working principle as impedance control. However, it separates explicitly the position control from the impedance control. The position controller is an inner loop, as in industrial robots, designed to be stiff to reject position disturbances robustly. The trick is that the position controller does not receive directly the desired motion but the output of the 'impedance controller'. Figure 2.7(a) shows a diagram sketching the admittance control. As it can be seen, the impedance controller ('admittance controller' to be more correct, since the input is force (wrench) and the output is position) receives two inputs: the desired position and the end-effector wrench (force and moment). A proper target impedance will generate an output that is a suitable position/orientation for the robot that maintains the desired dynamical relationship between force and motion.

Compliance Control As stiffness control is for impedance control, compliance control is a subclass of admittance control, where the controller is only concerned with the static relationship between the forces exerted by the robot and the deviation of the position/orientation from the desired values. Figure 2.7(b) shows a sketch of the compliance control, where the controller only generates a reference position (no higher derivatives) for the position controller.

2.4 Impedance control

As we have seen, impedance control represents a strategy for constrained motion rather than a specific control scheme. There is no unified 'off-the-shelf' control scheme but rather depends on the application scenario and its constraints. Thus several control

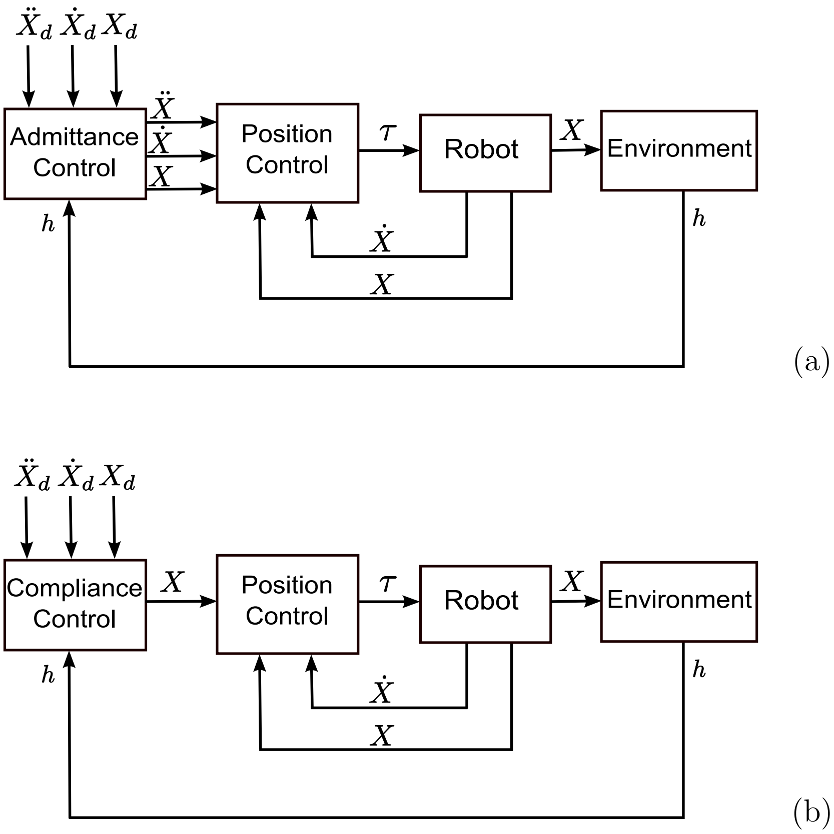


Figure 2.7: (a) admittance control, (b) compliance control

schemes have been proposed for controlling the relation between robot motion and interaction force. Certainly, the first scheme that can be considered as impedance control was presented by Whitney in 1977 [131]. He used the terms 'damping' and 'accommodation control' where the force feedback was closed around a velocity controller and the interaction was converted to a velocity by a certain constant factor. Later on, Salisbury in 1980 [109] proposed to modify such scheme to directly modify the end-effector Cartesian position due to the interaction forces. Both approaches are very simple to implement in a real system but it is extremely difficult to achieve a high dynamical performance. The reason is that a typical robot manipulator has highly complex dynamics and these approaches use as virtual model for the manipulator a simple linear time-invariant mass-spring-damper target system. For that reason, these methods require of an extra block: a model-based dynamical control law (usually called 'computed-torque method') that decouples and linearises the robot's dynamical model in order to cancel the effects of non-linearities on the performance of the impedance control law. The implementation of the model-based dynamical controller obviously requires of solving complex dynamical models of the robot which in turn makes the implementation of these impedance control strategies quite cumbersome and complex. Moreover, its implementation is prone to be sensitive to uncertainties both on the robot and environment models and their parameters. Five years later, in 1985, Hogan [50] created a unified theoretical framework embracing the term impedance control (note that previous schemes were just using stiffness or damping elements). Nowadays, his publication is considered a seminal work and is of obligatory reference for any work on impedance control.

2.4.1 Force-based impedance control

The scheme proposed by Hogan [52] started assuming that a Cartesian dynamical model of the robot is available and matches perfectly the real model. If we recall the Lagrangian formulation of the dynamics of a rigid manipulator interacting with the environment:

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{B}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{G}(\boldsymbol{\theta}) = \boldsymbol{\tau} - \boldsymbol{J}^T(\boldsymbol{\theta})\boldsymbol{F} \tag{2.1}$$

where $\boldsymbol{\theta}$ is the joint variable n-vector and $\boldsymbol{\tau}$ is the vector of generalized joint driving torques, $\boldsymbol{M}(\boldsymbol{\theta})$ is the nxn inertia matrix, $\boldsymbol{B}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ are the Coriolis/centripetal forces, $\boldsymbol{G}(\boldsymbol{\theta})$ is the gravity vector, $\boldsymbol{J}^T$ is the nxm Jacobian matrix relating joint space velocity to task space velocity and $\boldsymbol{F}$ is the m-dimensional vector of external forces acting on the robot.

Likewise, Equation (2.1) can be expressed on the Cartesian space (also called task or configuration space) where the formulation of a planning strategy is simpler than in joint space.

$$\boldsymbol{\Lambda}(\boldsymbol{X})\ddot{\boldsymbol{X}} + \boldsymbol{\Gamma}(\boldsymbol{X}, \dot{\boldsymbol{X}}) + \boldsymbol{\eta}(\boldsymbol{X}) = \boldsymbol{u} - \boldsymbol{F} \tag{2.2}$$

where $\boldsymbol{\Lambda}$ is the nxn Cartesian space inertia matrix, $\boldsymbol{\Gamma}(\boldsymbol{X}, \dot{\boldsymbol{X}})$ is the Cartesian space term including centrifugal and Coriolis effects, and $\boldsymbol{\eta}(\boldsymbol{X})$ is the Cartesian space term that expresses the gravity effects. The vector $\boldsymbol{u}$ is the equivalent end-effector torque corresponding to the input joint torques $\boldsymbol{\tau}$.

The relations between Cartesian and joint space terms are given by the following relations:

$$\begin{aligned}
\boldsymbol{\Lambda} &= (\boldsymbol{J}\boldsymbol{M}(\boldsymbol{\theta})^{-1}\boldsymbol{J}^T)^{-1} && (2.3)\\
\boldsymbol{\Gamma}(\boldsymbol{X}, \dot{\boldsymbol{X}}) &= \boldsymbol{J}^{-T}\boldsymbol{B}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\boldsymbol{J}^{-1} - \boldsymbol{\Lambda}(\boldsymbol{\theta})\dot{\boldsymbol{J}}\boldsymbol{J}^{-1} && (2.4)\\
\boldsymbol{\eta}(\boldsymbol{X}) &= \boldsymbol{J}^{-T}\boldsymbol{G}(\boldsymbol{\theta}) && (2.5)\\
\boldsymbol{u} &= \boldsymbol{J}^{-T}\boldsymbol{\tau} && (2.6)
\end{aligned}$$

To obtain a target impedance that in the ideal case would resemble:

$$\boldsymbol{F} = \boldsymbol{M_T}\ddot{\boldsymbol{X}} + \boldsymbol{D_T}\dot{\boldsymbol{X}} + \boldsymbol{K_T}(\boldsymbol{X_d} - \boldsymbol{X}) \tag{2.7}$$

where $\boldsymbol{M_T}$, $\boldsymbol{D_T}$ and $\boldsymbol{K_T}$ are the inertia, damping and the stiffness coefficients, respectively. In the Laplace domain, the target impedance is expressed as:

$$F(s) = G_T(s)(X_d(s) - X(s)) \tag{2.8}$$

from where

$$G_T(s) = \boldsymbol{M_T} s^2 + \boldsymbol{D_T} s + \boldsymbol{K_T} \tag{2.9}$$

Hogan [52] proposed the following non-linear control law:

$$\boldsymbol{\tau} = \boldsymbol{\Lambda} \boldsymbol{M_T}^{-1} \left[\boldsymbol{K_T}(\boldsymbol{X} - \boldsymbol{X_d}) - \boldsymbol{D_T}\dot{\boldsymbol{X}} + \boldsymbol{F} \right] + \boldsymbol{\Gamma}(\boldsymbol{X}, \dot{\boldsymbol{X}}) + \boldsymbol{\eta}(\boldsymbol{X}) - \boldsymbol{F} \tag{2.10}$$

This non-linear control law is represented in Fig. 2.8 [126]. The scheme is composed of a computed-torque control law (blocks enclosed by dashed lines) and an inner loop with a force-based impedance control. The block G_e represents the environment model. The use of this and later proposed methods provide a high dynamical performance assuming a precise knowledge of the robot and environment models. Nonetheless, its practical deployment encounters another impediment. As we have seen, the result of the previous control law are joint torques, however, most industrial robots are nowadays position-controlled and are not equipped with joint torque feedback by default. Although the use of motor current measurements would provide with a good estimation of the joint torques, the high friction and other non-linearities on the transmission would cause inaccuracies of the current-torque causality that would lead to more imprecise dynamical model as expected. For that reason, these type of schemes are primarily suited for new designs, especially the ones using direct drive systems, and those who permit and require a complete new control system.

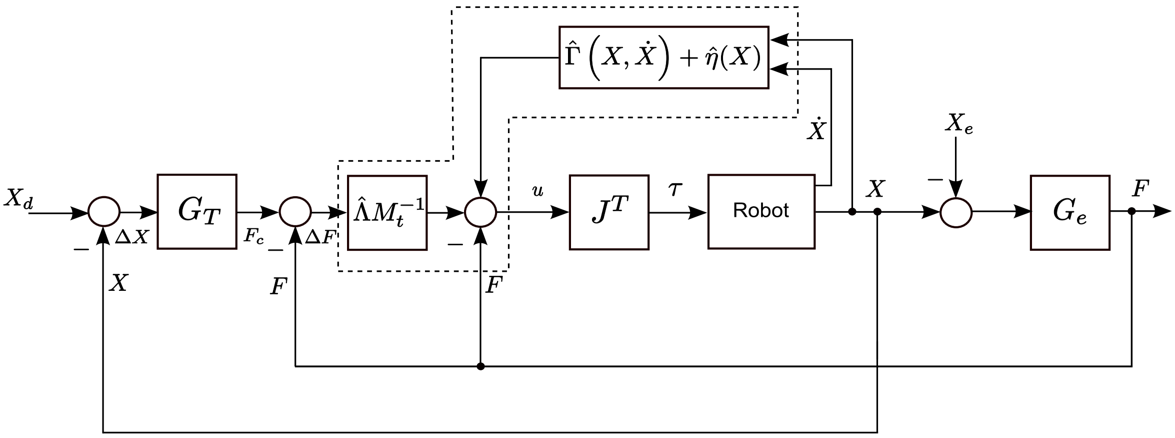


Figure 2.8: Force-based dynamical impedance control proposed by Hogan [52]

2.4.2 Position-based impedance control

For the reasons mentioned in the last section, position-based impedance control strategies are the most suitable solution for its use in already-existing industrial or commercial robotic systems. The main idea is to close a force loop around the existing position controller thus not needing any modification on the existing system. According to the error to minimise (force or position), the position-based schemes can be further divided into two subclasses: position model error impedance control and force model error impedance control. The following sections will describe their main features and differences.

2.4.2.1 Position model error impedance control

The simplest and most obvious implementation is that of using the existing internal position controller and close an outer loop around it based on force feedback. Figure 2.9 shows the control scheme. Notice that the outer force loop is only closed in case that end-effector is in contact with the environment ($F \neq 0$).

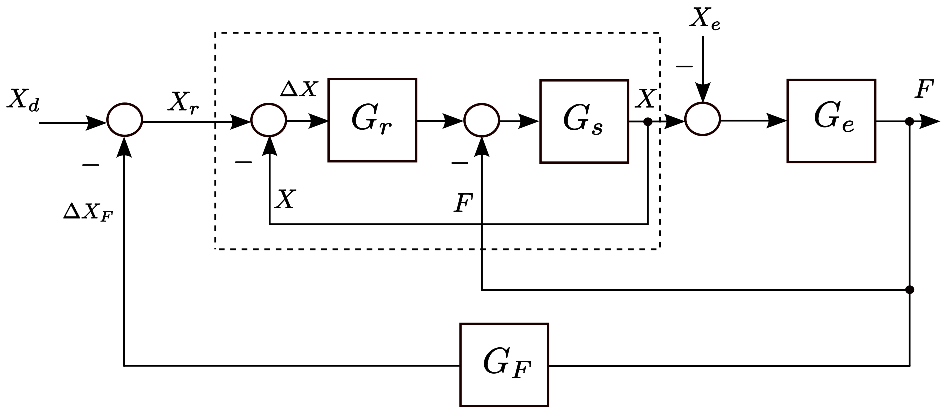


Figure 2.9: Position model error impedance control scheme. Robot model (G_s) and internal controller (G_r) are enclosed by dotted lines, representing the blocks that are not to be modified

The block G_s represents that plant model, i.e. the robot, G_e is the environment model, and G_r represents the existing internal position controller. The block G_F is an admittance controller (input is force and output is a position), whose target admittance is G_T^{-1}. Thus, the position error input to the existing position controller is:

$$\boldsymbol{\Delta X_r} = \boldsymbol{X_r} - \boldsymbol{X} = \boldsymbol{X_d} - \boldsymbol{\Delta X_F} - \boldsymbol{X} = \boldsymbol{X_d} - \boldsymbol{X} - G_T^{-1}\boldsymbol{F} \tag{2.11}$$

that is, $\boldsymbol{\Delta X_r} = \boldsymbol{e_p}$, where $\boldsymbol{e_p}$ is known as the position model error [126]. That means that the impedance model error is fed forward to the existing position controller. In other words, the performance of the impedance controller is mostly dependent on the tracking capabilities of the existing position controller.

2.4.2.2 Force model error impedance control

The second scheme is also known as outer/inner loop stiffness control and, in this case, builds two parallel feedback loops upon the existing position controller, one based on force measurements and the second based on position measurements. Figure 2.10 shows the control scheme. Notice that in this case the external impedance control loop is always closed, even in free space when there are no interaction forces ($F = 0$).

The first feedback loop uses the robot position measurements to generate the position error $\boldsymbol{\Delta X_d} = \boldsymbol{X_d} - \boldsymbol{X}$ that is passed through the impedance controller block G_T. The output of that block is a nominal reference force $\boldsymbol{F_d}$. The second external loop will track this reference force using the measurements of the end-effector forces. Thus in contrast to the previous scheme, the model error is not eliminated using the existing position controller but by the compensator G_F, that is usually implemented as an admittance filter.

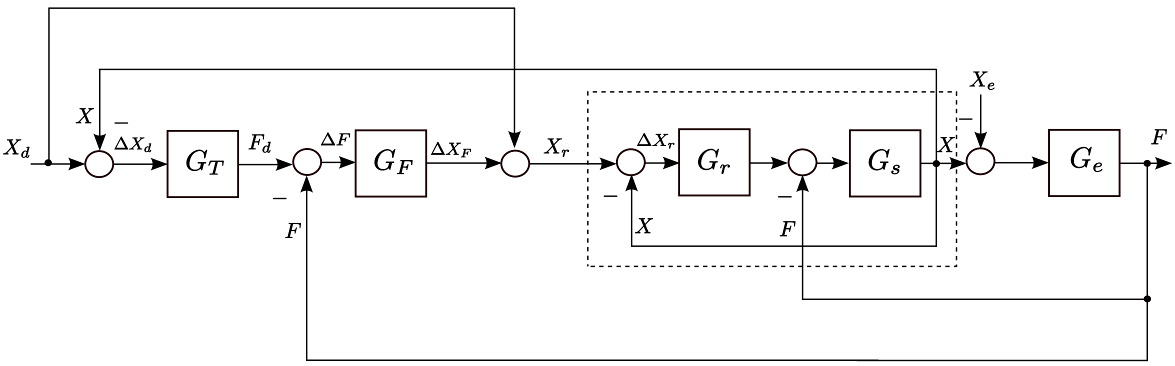


Figure 2.10: Force model error impedance control scheme

Ideally, the measured contact force $\boldsymbol{F}$ should equate to $\boldsymbol{F_d}$ thus the system's target behaviour is:

$$\boldsymbol{\Delta F} = \boldsymbol{F_d} - \boldsymbol{F} = G_T(\boldsymbol{X_d} - \boldsymbol{X}) - \boldsymbol{F} \tag{2.12}$$

thus $\boldsymbol{\Delta F} = \boldsymbol{e_f}$, where $\boldsymbol{e_f}$ is known as the force model error [126].

2.4.3 Force vs position-based impedance control

Both previously presented schemes (force and position-based in its two flavours) use similar ideas to achieve the target impedance by trying to reduce the impedance model errors $\boldsymbol{e_f}$ and $\boldsymbol{e_p}$, respectively, to zero. The position-based approach is much simpler and easier to implement although as drawback might be mentioned that the scheme is an open-loop control for the target impedance. The block G_F will set an admittance to be tracked but the only feedback control loop is closed around the position controller. Moreover, the accuracy of the existing position controller will limit the range of possible target impedances (very soft impedances might not be reachable). On the other hand, the force-based approach ensures the tracking of the selected impedance, as a feedback loop is closed around it using force measurements. The major drawback is the fact that the external impedance loop is closed even when there are no interaction forces. This might lead to lower the position performance of the system in free space but also makes more complex the control of the transitions (i.e. from contact to free-space and viceversa). This method might be computationally reasonable if there is no need to compute the complete robot dynamics as when the manipulator has low gravity terms and is not used at high speeds.

In summary, position-based impedance control is more suitable for applications where it is required to maintain the high position accuracy of the existing position controller for some directions. The force-based impedance control is more suitable for applications where dynamics effects are not so dominant (slow motion or systems using direct drive motors).

2.5 Environment Estimation

It has been seen in previous sections that the control schemes used for active impedance control always include a model of the environment as well as of the robot dynamics. In order to be able to determine and select the desired steady-state forces between robot and environment after the contact, it is necessary to know which is (at least) the stiffness of that particular environment. Environments are usually modeled as a linear Kevin-Volgt model following Equation (2.13), as it is easier to anaylise its properties, especially when analysing the complete control system.

$$\boldsymbol{F} = \boldsymbol{K_e}(\boldsymbol{X} - \boldsymbol{X_e}) \tag{2.13}$$

where $\boldsymbol{F}$ is the contact force, $\boldsymbol{K_e}$ is the stiffness of the environment, $\boldsymbol{X}$ is the end-effector position at the contact point, and $\boldsymbol{X_e}$ is the static position of the environment. Figure 2.11 depicts such a concept, where a manipulator of mass $\boldsymbol{M}$ contacts the environment at position $\boldsymbol{X_e}$ trying to reach the desired end-effector position $\boldsymbol{X_d}$.

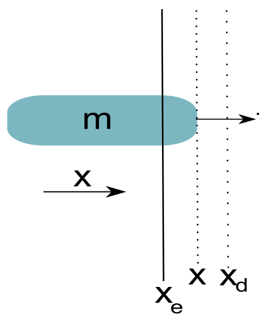


Figure 2.11: Manipulator in contact with the environment

A better and more physically-consistent environment model has been proposed in literature [34] that is known as Hunt-Crossley relation, a non-linear relation instead of the previous linear model (or spring-like model). Besides, it would allow to describe the behaviour of both stiff and soft objects, and it is computationally simple to be computed on-line. The model obeys the following relation:

$$\boldsymbol{F(t)} = k\boldsymbol{X}^n(t) + \lambda\boldsymbol{X}^n(t)\dot{\boldsymbol{X}}(t), \boldsymbol{X} \geq 0 \tag{2.14}$$

where n is a real number that takes into account the geometry of the contact surfaces.

Irrespective of the use of these models or any other, it is clear to see that the desired steady-state contact force (and transient behaviour) will not be achieved if the knowledge about the environment (even of a simple spring-like model) is not known in a quite accurate form. It is there where estimation of the properties of the current environment plays an important role to build a adaptive impedance controller that is able to on-line adapt its properties in order to comply with the current estimated environment. Thus estimation of the dynamical properties of the contact is a important component for successfully implementing impedance control in a real application. Several schemes are proposed to regulate the robot-environment contact forces and to deal with model uncertainties. In [89] a model reference adaptive algorithm is proposed to deal with the uncertainty of the parameters that describe the environment. In [36] an artificial neural network-based PI gain scheduling controller is proposed that uses estimated human arm parameters to select appropriate PI gains when adapting forces in robotic rehabilitation applications. In [62] a neural network approach is also used to compensate both for the uncertainties in the robot model, the environmental stiffness, and the force sensor noise. Similarly, in [111] and [88] adaptive impedance control schemes are presented to deal with uncertainty of the environmental stiffness as well as uncertainty in the parameters of the dynamical model of the robot or the force measurement. These methods adapt the desired trajectory according to the current scenario, though using cumbersome or unclear methodologies for the selection of impedance parameters. Moreover, some of them might not be applied where the environmental properties are of non-linear nature [111]. In [138], a Model Referenced Adaptive System (MRAS) is used to recursively estimate the dynamic properties of the robot-environment interaction by using measurements of the contact forces and the estimate forces (based on position/velocities measurements and previous estimations). This might be considered a direct method of estimating the properties of the environment.

There are also indirect methods to estimate the properties of the environment. In [111], a reference trajectory $\boldsymbol{X_r}$ based on estimates of the environment stiffness and location ($\hat{\boldsymbol{K}}_{\boldsymbol{e}}$ and $\hat{\boldsymbol{X}}_{\boldsymbol{e}}$, respectively) is generated in order to track a desired force trajectory $\boldsymbol{F_r}$. The authors in [117] used a similar approach by means of a Model Reference Adaptive Control (MRAC) strategy. Both works used a pure spring-like model for the environment, together with measurements of the contact force and end-effector position. In [86], recursive least square method (RLS) is used to estimate the location and dynamic properties of the contact. Kalman filtering has been also used [118] to estimate on-line the environment properties. The damping and stiffness parameters of the dynamic equation describing the interaction appear multiplied with state variables and thus, the problem is non-linear in its parameters. The proposed filters are then extended Kalman filters that deal with non-linear models. They also provide good accuracy and fast convergence for soft environments (characterised by being highly non-linear).

2.6 Conclusions

This chapter reviewed most of the fundamental classical robot interaction control methods, including both passive and active alternatives. Focusing on active control systems, the chapter disclosed several control schemes, especially impedance control schemes, discussing advantages, pitfalls, and suitable application areas for them. As it was stated, there is no 'universal' interaction controller that would be suitable for all situations. At least, it is not yet found in the current state of the art. Each of the presented alternatives requires an analysis of the problem at hand in order to choose among them depending on several factors. These can range from how much information about the environment is available, how computationally expensive the solution can be, or whether the internal robot's position controller might be modified or not.

From the review on interaction control methods arose the observation that the estimation and identification of the properties of the environment is of utmost importance for the successful performance of an impedance controller. Especially if this controller is deployed in a real uncertain environment.

Chapter 3

BIOLOGICAL CONTROL METHODS

This chapter introduces biological control concepts for arm reaching movements as well as the use of neuroevolutionary techniques. These techniques inspire part of the solutions later developed in the chapters that describe the contributions of this thesis.

3.1 Introduction

Biology can provide useful examples on how to adapt to changing environments. Biological systems show superb performance under environments that in the robotics realm still nowadays would be considered as challenging and, possibly unfeasible, due to its unstructured nature. Notably, the brain shows as well levels of adaptability yet to be seen in other human-engineered machines.

3.2 Biological motor control

In the past two decades there has been an increasing interest for unveiling the secrets of the brain's internal processes and an explosion on the number of publications in cognitive neuroscience [60]. Cognitive neuroscience (an area that began to emerge in the 1980s) is the multidisciplinary area that focuses on the psychological, computational, and biological mechanisms that control cognition and behaviour, with the ultimate aim at understanding how the brain perceives and initiates action, how it learns, and how it remembers. The quest for understanding brain processes and the recent technological advances that promise a new way of looking at the brain (non-invasive brain imaging methods like fMRI [82]) have generated a big deal of studies. Whether brain imaging results are always being used in a proper way is still a topic of concern [9], [127], though it is clear that many findings are being revealed and that our understanding about the brain processes has improved over the last two decades more than in the second half of the last century, when the term 'neuroscience' was coined by the American biologist Francis O. Schmitt in 1962 by organising the MIT Neurosciences Research Program.

The processing of the available sensory information is a fundamental factor on the brain's efforts to adapt to the current situation, a task that just by itself poses an enormous challenge. Not only sensory information is highly noisy and might be incomplete, but also the actuation signals might be degraded by noise. Last but not least, the environment conditions are constantly changing, what poses yet another challenge for a precise motor control. Despite those problems, the brain seems to easily control our motor actions. Current knowledge hypothesizes about how the brain achieves this great performance and primarily names three methods: estimation, prediction, and impedance control.

3.2.1 Reaching movement

Current neuroscience studies show some interesting facts about the way arm motions are generated. According to these studies, in the initial phase (during the reaching movement towards an object), on-line feedback is used to reduce errors while performing the movement [69]. The approach is similar to classical control theory, that uses feedback error signals to minimise the plant's error with respect to a given reference. More interesting is the current hypothesis that postulates that these errors also modify the internal models with which the motor plan computes the necessary motor actions [80]. A good example and major claim to prove this hypothesis is found when comparing the movement performance of patients with cerebellar lesions and those suffering of Huntington's disease. Patients with cerebellar lesions exhibit errors in their reaching behaviours when adapting to novel arm dynamics from movement to movement, suggesting malfunctioning on the learning of internal models, but they do not exhibit problems on real-time correction of ongoing movements. On the other hand, patients with Huntington's disease exhibit poor performance on tasks that depend on the ability to produce real-time corrections, but not on movement-to-movement corrections, suggesting that there are different control paths for the correction during an ongoing movement and the correction between different trials [119]. That is to say, Huntington's patients impairments are related to deficiencies on monitoring and producing real-time corrections, whereas patients with cerebellar lesions might have problems updating the internal models.

Many years have been devoted to prove whether biological movements are planned in Cartesian space [95],[42] or, on the contrary, in the joint space [78]. Nowadays, the most widely accepted opinion is that movements are not planned in a single frame of reference, but many different frames are involved [32]. Based on this idea, [47] develops a multi-referential controller based on biological principles which is applied for the control of the reaching movements for a redundant serial robot manipulator. The controller is composed of two dynamical systems that are simultaneously active but in different frames of reference, as experiments in biological systems suggest.

3.2.2 Estimation

Estimation in biological systems is basically concerned with generating a reliable estimation of the state of the body while dealing with partially incomplete or noisy information. Since usually the information about our body can be measured from different sources (for instance, our hand position can be obtained by using proprioceptive or by visual information), scientists formulated the hypothesis that our brain combines multimodal information to reconstruct the degraded information about the state of the body.

Estimations can be improved by including sensory feedback to the forward model that estimates the current state. In computational terms, this is similar to a Kalman filter [67] that estimates the state of a linear dynamical system that is perturbed by a Gaussian noise. The Kalman filter recursively estimates the current state based on the current measurement and the estimate from the previous state. The filter has basically two distinct phases: a *prediction* phase that makes use of a dynamical model and a *update* phase that updates the prediction as soon as sensory information is available. In neuroscience, such a filter would be used to compensate for sensorimotor delays and to reduce the uncertainties in current state estimation. Experimental neurophysiological studies have reproduced the expected outcomes of the brain using similar processes to a Kalman filter [76], [3].

3.2.3 Prediction

Prediction has been shown to play a very important role in biological systems [23]. Human motor control relies on predictions rather than on sensory feedback as have been experimentally verified [40], [59]. The main reason stems from the fact that the considerable delays in transmission of the nervous system (information travels at around speed of sound) cannot explain the quick and gracious movements observed in biological systems by just using a pure feedback control system [92], [136]. Needless to say, these predictive strategies apply to deliberative manipulation of ordinary (and so, predictable) objects. Reactive mechanisms also co-exist in biological control systems that are enabled under unpredictable loads.

The notion of the brain using internal models [68] has emerged as a powerful theoretical concept to explain human motor control since Ito introduced the concept forty years ago [58]. Forward models are transformations that relate actions to their sensory consequences [129][37][91]. It would seem that the brain would not require to model the relation between motor actions and sensory consequences, as this is actually the model of the physical world, but it has been proven that the brain internally computes it. The reason appears clear: the brain needs to predict the consequences of our actions and the behaviour of the world. Similarly, the brain needs to be using inverse models to generate the proper commands for obtaining the desired change in state of the body. These internal models are not static, they need to be continuously updated, both on short and long time scales. While performing a movement we might notice forces acting against us that were not predicted and are not from external sources. Thus they might be an indication of a wrong model (and consequently, wrong prediction). Similarly, our muscles change their properties with time (by exercising, for instance, the muscles grow). This needs to be taken into account in order to update our internal models.

The means to gather information for updating internal models is only one: the sensory feedback obtained from the environment when performing actions; this is the training signal for the internal models. The forward models can be easily tuned and adjusted by comparing sensory predictions with actual sensory information. Inverse models are not so immediately trained since motor command errors are not signals available to the brain. Imagine you are learning to play tennis; if you fail to hit the ball, there is no signal telling you which was the correct motor torque to send to your muscles. The only source of information left is simply the sensory information. This information will pass through forward models to convert to motor commands that might be used to compute an error signal. Some experiments in the oculomotor system (where inverse models are perhaps best understood) indicate that the brain might be using feedback-error learning techniques [114], that is, it uses the feedback errors in order to train the inverse models. If the feedback error signal is zero, it means that the inverse model is perfect and there is nothing to be done. Otherwise, the feedback error signal is used to update and adjust the inverse model [61].

The seminal work from Johansson [59] presented an architecture for predictive sensorimotor coordination in human object grasping. In this work, the author discusses how we adapt fingertip forces to the constraints imposed by the properties of the objects to handle (light/heavy, slippery, complex shape, etc ...). He postulates about the use of feedforward control mechanisms to predict physical properties of objects in order to adapt force motor commands. The results of the study are summarised in a sensorimotor control architecture (Figure 3.1) that uses visual and proprioceptive information together with internal models (memories) for the adjustment of fingertip forces to the target objects. First, the objects are identified using visual information and, subsequently, relevant force information is retrieved from the stored internal models in order to ensure a successful grasp.

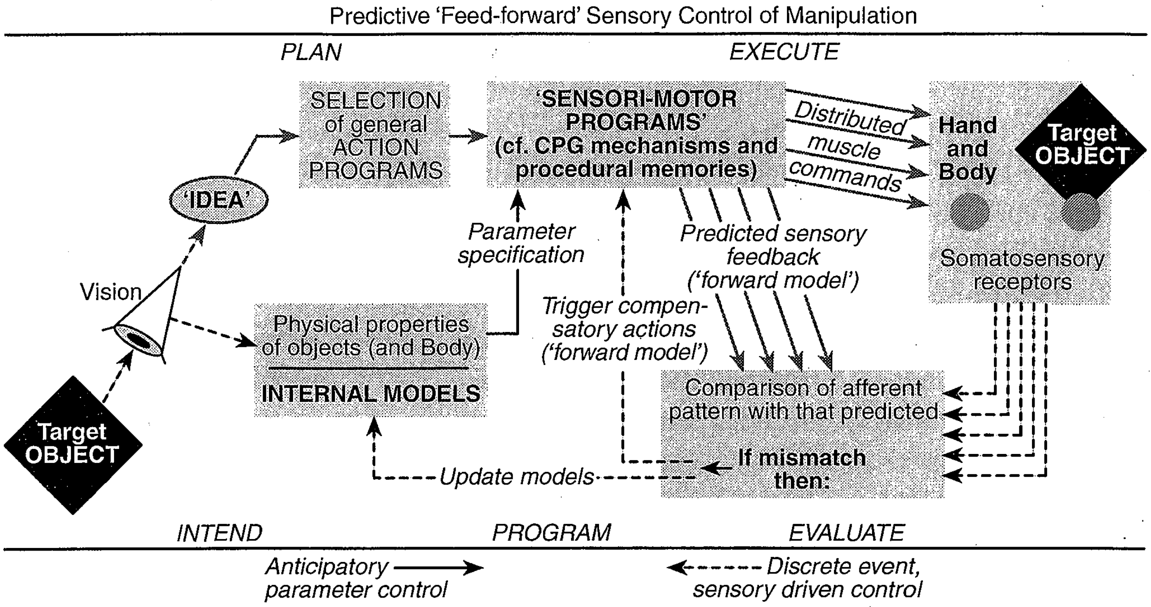


Figure 3.1: Sensorimotor coordination model for human object manipulation from Johansson (1998) [59] (Wiley, permission granted)

A bio-inspired predictive sensorimotor coordination scheme based on the previous architecture has been recently presented for the control of a robotic arm [81]. The scheme controls the coordination of hand, arm, and head in order to reach and object and preshape the fingers according to the required grasp. Furthermore, the architecture incorporates a predictive module to predict the tactile information that will be received after grasping.

On a higher level of the control hierarchy, recently some studies showed how the human brain is able to use context cues of an ongoing situation to predict and thus select the proper internal model for the situation [75]. A very thought experiment to prove that claim was presented in [20] that shows how the arm compensates for Coriolis forces using context-specific information. Coriolis forces appear in a body when it moves in a rotating frame and that also holds true for arm reaching movements during a simultaneous rotation of the torso. Coriolis forces are proportional to the rotation velocity of the moving frame and the arm reaching linear velocity; the force generated is orthogonal to the forward movement of the hand. We do not consciously perceive these forces and, nonetheless, we are able to reach objects successfully while rotating the torso, which indicates that our brain is able to compensate for these forces. The experiment presented here used a virtual environment to induce the subjects to believe that the body was rotating. Thus the subjects would think that the arm movements needed to be compensated for Coriolis forces, when it was actually just an illusory movement. In an actual rotation of the body to the left, the Coriolis force caused by a forward reaching movement would act rightward; therefore a subject would have to compensate leftward to successfully achieve the desired movement path. Thus, if the rotation is just illusory, in case of the brain anticipates and compensates for Coriolis forces, the described movement would end up reaching too far to the left. And that is actually was what observed. Moreover, additional experiments placed the subjects in the center of a slow moving room rotating at constant velocity, what made them feel as if the room was stationary. However, as Coriolis forces would anyway appear, a fact the subjects' brains were unaware of, large reaching errors were observed in the direction of those forces.

These observations indicate that the brain plans trajectories depending on the context in which they are occurring. The theory of a network of multiple internal models that can be acquired by learning and combined according to the current context has been gaining interest and some theoretical models have been also presented. Figure 3.2 shows the architecture of multiple internal models proposed by Kawato [68] and Wolpert [137]. The architecture serves to explain manipulation actions of different objects with a finite set of internal models for different objects. Using visual information of an object, the proper internal model is activated. If during the manipulation of an object, the forward model output (the prediction) does not match the received sensory feedback, the internal model will be switched to the model of the most suitable object. The switching between modules occurs on the basis of the predictions of a Bayes predictor that is based on the context of that specific action.

Recent studies in primates also show how their brains determine the likelihood of sensory data given an environment [73]. As stated later by [136] and [74], Bayesian inference methods can reproduce these empirical observations. Bayesian probability is composed of a prior probability and a likelihood function. The prior probability represents how likely is an environment before contacting it, and the likelihood represents the probability of the perceived sensory feedback given that the hypothetical environment is true. In other words, the brain is able to refine the multiple sensory information by using prior knowledge to yield a better estimate of the current environment and/or state.

Even an initial imperfect estimate can be refined by using prior knowledge. On this basis, recently Wolpert [135] also presented a theoretical computational model that incorporates the use of multiple internal models in the lowest layer of a Bayesian-based multilayer control system. The architecture is called MOSAIC (MOdular Selection And Identification for Control) and contains pairs of predictor-controllers. The predictors are responsible for providing the probability of each of the possible contexts, which in turn will be used to weight the contributions of the different controllers. Thus these multiple predictor-controller pairs can be seen as motor primitives that can be used and combined to create more complex movements. The information provided by internal models might be also used for example to remove the sensory information due to self-generated actions. An efferent-copy of the action to be pursued is used by the forward model to predict the sensory information to be received. This prediction is later subtracted from the sensory information, so that what is left is just due to the external world acting upon the body. A great discrepancy between sensory information and prediction would suggest that an external event caused this sensory input and thus

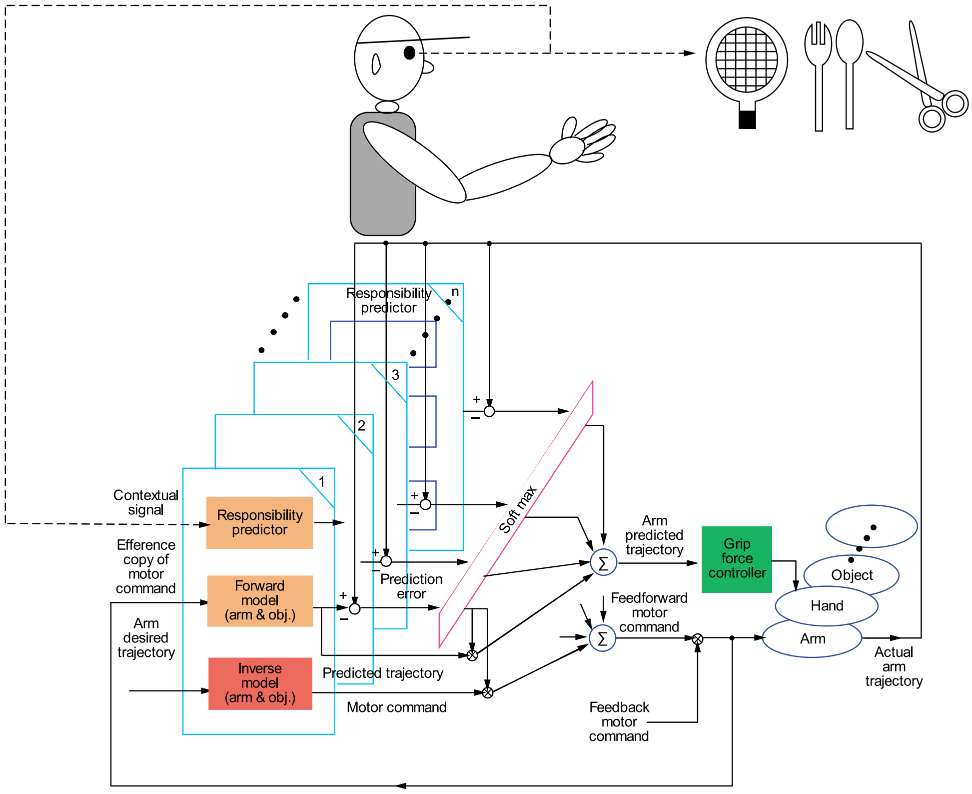


Figure 3.2: Architecture based on multiple paired forward and inverse models from Kawato (1999) [68] (*Elsevier, permission granted)*

we can differentiate between self-generated and external actions.

3.2.4 Trajectory generation

Another interesting topic is the selection of a trajectory to follow. It would seem that a task might be achieved by a theoretical infinite number of movements, but what it is observed is a characteristic pattern on how we reach for objects. Unconstrained arm reaching movements are characterized by showing approximately bell-shaped velocity profiles and straight paths [95]. Another observation points to an inverse relationship between the radius of curvature of the trajectory and the velocity that is known as Power Law [79]. The law is formulated as $V(t) = kR(t)^{-1/3}$, where $V(t)$ is the instantaneous angular velocity, k is a gain factor that remains constant during the execution of the motion, and $R(t)$ is the radius of curvature of the movement. Similarly, Fitts' Law postulates that the execution time is kept approximately independent of the length of the trajectory [39].

Those observations lead to the hypothesis that the brain might be optimising some performance measure and choosing the trajectory yielding the minimum value. Several criteria have been proposed in the past decades. In [96] the trajectories minimized various measures of physical cost (for example, movement time, maximum force, impulse, energy etc.) but no conclusive results were obtained. Later, [125] proposed as criterion the minimum torque-change. Nowadays, two models share most acceptance, both having large data sustaining its claim. The first considers and explains the kinematics of the movement and is called the minimum jerk model [42]. The jerk (third derivative of position) seems to be a criterion that is minimised in reaching tasks. That is to say, the brain tries to produce the smoothest possible movement of the hand. This model is able to predict experimentally the straight paths and bell-shaped speed profiles of the studies indicated previously. However, it is not clear how the brain would measure the jerk and the model does not take into account the dynamics of the musculoskeletal system. A second model have been presented that takes into account dynamics and postulates that the criterion to minimise by the Central Nervous System (CNS) might be the variability of the end-effector position, what was proved for both arm reaching movements and saccadic eye movements [46]. This criterion, unlike previous hypothesis, is easy to compute from sensory information and, indirectly, minimises the jerk thus generating as well smooth trajectories

The previously described methods focus on open-loop control, that is, they did not take into account sensory feedback. However, they yield accurate predictions of movements averaged over multiple repetitions of a task. A second class of methods focus on optimal closed-loop control [123]. These methods construct simultaneously the feedback control law required to achieve the best performance by taking into account sensorimotor noise and delays. Thus, their predictions are not only regarding average behaviour and are able to unify in a theoretical framework both high-level task goals and low-level real-time sensorimotor commands. In summary, although a series of simple costs elucidated the performance criteria that is most relevant to certain controlled movements, it seems clear that there is no single performance criterion, or in other words, the performance criterion to optimise is likely to be a mix of different cost measures.

3.2.5 Trajectory execution

So far so good. The brain estimates the state of the body, predicts sensory consequences of actions, selects an appropriate action, and generates a trajectory plan to follow. The problem remains now of how to realise this trajectory, as the motor commands are very dependent on the body dynamics and the current environment. Several theories are proposed but, in this area, there is probably less general consensus than on previously described strategies.

Inspired by knowledge on controlling robotic systems, one proposal is that the CNS uses inverse dynamics to determine the required motor commands [55],[113],[49]. The ideas is that the CNS uses internal models that relate muscle activation patterns to change in body state. These models allow the brain to compensate for the effects of predictable environmental forces. Another hypothesis postulates about what are called the equilibrium points [100],[41]. This theory proposes that the musculoskeletal system can be compared to a spring, and that the movement is achieved by simply moving along a trajectory composed by the equilibrium points of this dynamically-changing spring. In other words, the CNS might execute a movement by simply generating a set of neuronal activations required to achieve a new equilibrium position. That leads naturally to the idea of a virtual potential field, which smooth and naturally drives the arm towards its goal during a reaching movement.

A posterior theory of motor control tries to find a unified answer for explaining both equilibrium point and inverse dynamics hypotheses [43]. By microstimulation of the frog's spinal cord (surgically separated from the rest of the CNS), the experimenters discovered that repeatable and well-organized motor responses were elicited. The force responses were recorded and the spatial variation of the measured force vectors resulted in a force field that astonishingly converged to a single equilibrium point. Thus, these force fields might be motor primitives that the brain uses to solve the inverse dynamics. Moreover, the temporal evolution of the force fields from one equilibrium point to the next reproduces what has been called a virtual trajectory [51]. The virtual trajectory is a trajectory that has as via points the equilibrium points for a limb and allows a unified description of the posture and the movement of a multi-joint system.

3.2.6 Impedance control

There are situations where the brain cannot predict the changes on the environment and, for those cases, the brain uses a compensation method named impedance control [50]. In an uncertain scenario, the CNS will modify the mechanical stiffness of the arm in order to increase the stability and reject force disturbances [93]. Although this compensation method is mainly useful for unpredictable changes, it also provides with an additional asset by increasing the robustness against inaccurate estimates of the state of the body and in anticipation of predictable external events [77].

Neurophysiological studies on the modification of the mechanical impedance in humans have focused in four main situations:

- under static conditions, i.e. hand/arm at constant position
- under dynamic conditions, i.e. during reaching movements under unstable environments that perturb the motion of the arm
- in anticipation of predictable external forces
- at early stages of the response in order to counteract for unknown and unpredicted disturbances

Under static conditions, the work in [22] shows how learning and adaption mechanisms are used to modify the stiffness of the arm. The subjects' training took place over three consecutive days showing how subjects gradually adapted the orientation of the maximum stiffness of the arm to the direction of the perturbation load. By contrast, in [13] the experiments took place in unstable dynamic environments created by a robotic interface. Similarly, the results showed how humans learn to stabilize unstable dynamics by selectively controlling the orientation of the maximum mechanical impedance of the arm in reaching movements. They were able to show how changes in mechanical impedance of the arm matched the direction of the perturbation force fields.

The anticipatory mechanisms used by the CNS to modify the mechanical impedance preceding predictable impacts was studied in [77]. The experiments aimed at describing the time dependency of the impedance parameters in a dynamic movement, in this case, during ball catching. Interestingly, the experiments discovered anticipatory muscle activation and change in mechanical stiffness around 100-200 ms before impact in a ball-catching task. The mechanisms underlying the initial adaption to unknown dynamics has been also investigated in [93]. They showed how anticipatory actions were used by the CNS to counteract unknown disturbances by using two mechanisms: increase on the mechanical impedance of the arm and creating an internal dynamical model for its subsequent use in following interactions. The results pointed out interesting facts about the learning of novel dynamics:

the first contact with an unknown and unpredictable environment is controlled by feedback mechanisms, that is, basically by using reactive and reflexive actions for the voluntary correction of errors.

the second and posterior contacts will use the knowledge obtained from the first contact in two ways:

- in order to predict the required activation of the muscles to modify the mechanical impedance of the arm, and
- to start creating a internal dynamical model of the contacted object

Another key question about the regulation of the arm's mechanical impedance is the selection process used to choose a particular set of impedance parameters for a given task. Similarly to the generation of a specfic arm trajectory, there are theoretical infinite possibilities for choosing a set of impedance parameters. The work from [105] studied this problem and suggests that the CNS chooses the stiffness as a trade-off between the stiffness level and the endpoint variance. Moreover, it is also observed that stiffness decreases as long as the task demands are met, that is, the optimal stiffness is sought that achieves the task with lowest stiffness value.

3.3 Evolutionary Algorithms

Evolutionary Algorithms (EAs) are direct search optimization methods that are inspired from the Darwinian concept of natural selection or survival of the fittest. There are different variants of EAs but all share the common feature of being strategies based on principle of natural selection and random mutation and recombination of individuals. Populations of solutions are iteratively evaluated to generate better and better solutions. Evaluation is the process of determining the fitness value of an individual. The fitness value measures how good an individual is with respect to other individuals. In general, the pseudo-code of an evolutionary algorithm would look like:

```
Initialise population with random candidate individuals
Evaluate each individual
repeat
   Select
   Recombine
   Mutate
   Evaluate
until Fitness level is satisfied
```

Evolutionary algorithms are based on two fundamental concepts: operators like recombination and mutation create *diversity* among the populations of individuals, whereas a *selection* process pushes the concept of survival of the fittest, increasing the fitness level over generations. Over the years, several evolutionary algorithms were developed that differentiate themselves slightly, usually on the application area. Some of the EA presented are:

Genetic Algorithms (GA), which are the most popular type of EA and focuses on optimizing general combinatorial (discrete) problems

Genetic Programming (GP), which is used to evolve programs to solve a computational problem

Evolutionary Programming (EP), which is similar to GP but works with fixed program structures

Evolutionary Strategies (ES), which focuses on optimizing continuous functions

Neuroevolution (NE), which uses artificial neural networks as structure and any other evolutionary algorithm to evolve the connection weights

3.3.1 Neuroevolution

Neuroevolutionary techniques merges two biologically-inspired areas: the use of artificial neural networks (ANNs) and evolutionary algorithms (EAs). Commonly, the class of evolutionary algorithms used in neuroevolution is either a genetic algorithm (GA) or an evolution strategy (ES). The first works usually with strings of binary vectors, whilst the latter works with real-valued vectors for the representations of a solution. Recombination (operation that merges the genetic information of two or more parent individuals producing one descendant) and mutation are the main nature-inspired operators used in ESs, whereas crossover (a type of recombination where two parents produce two offspring by exchanging genetic information) and mutation is mostly employed in GAs.

Artificial Neural Networks Artificial neural networks are mathematical models originally inspired by the biological nervous system. In 1871, Santiago Ramon y Cajal described for the first time the structure of the neuron and its dynamical behaviour. This work, which won him the Nobel Prize in 1906, revealed that neurons are separated from one another by narrow gaps (called synapses) in opposition to the reigning model of the time that the nervous system was made up of a network of continuous elements [38]. The biological neuron collects signals from neighbouring neurons through multiple inputs called dendrites (Figure 3.3(a)). If a certain level of activation is reached, the neuron 'fires' and sends a spike of electrical activity along the axon, which in turn ends in multiple branches. The endpoints of the axon's branches contain neurotransmitters, which are the chemical medium through which signals flow from one neuron to the next at chemical synapses.

Artificial neuronal networks tried originally to reproduce the basic working principle of the biological neuron (Figure 3.3(b)). This ideal model of the neuron is partly due to our limited knowledge about how the neuron works and partly due to limited computational constraints that require of simple models for each single neuron. ANNs are composed of interconnected processing units (neurons) that form a layered and structured network of neurons (Figure 3.4). ANNs contain an input layer receiving input from the environment, an output layer producing results to the environment, and one or more so-called hidden or intermediate layers. Each neuron collects all weighted outputs from the previous layer and passes it through an activation function that usually is in the form of a sigmoid function. The resulting activation is subsequently passed to the next layer through weighted outputs (synapses). Interestingly, despite of the simplicity of the individual neurons, the neural network as whole is able to produce complex behaviour. For a given input vector x, the output $y(x)$ of each neuron is computed as:

$$y(x) = g(\sum_{j=1}^{p} w_j x_j) \tag{3.1}$$

where p is the number of inputs, w_j are the weights of each input, and g is an activation function that weights how strong the output (if any) from the neuron should be. The usual choice for activation function g is the sigmoid function $g(z) = \frac{1}{1+e^{-z}}$ that outputs a continuous value between 0 and 1.

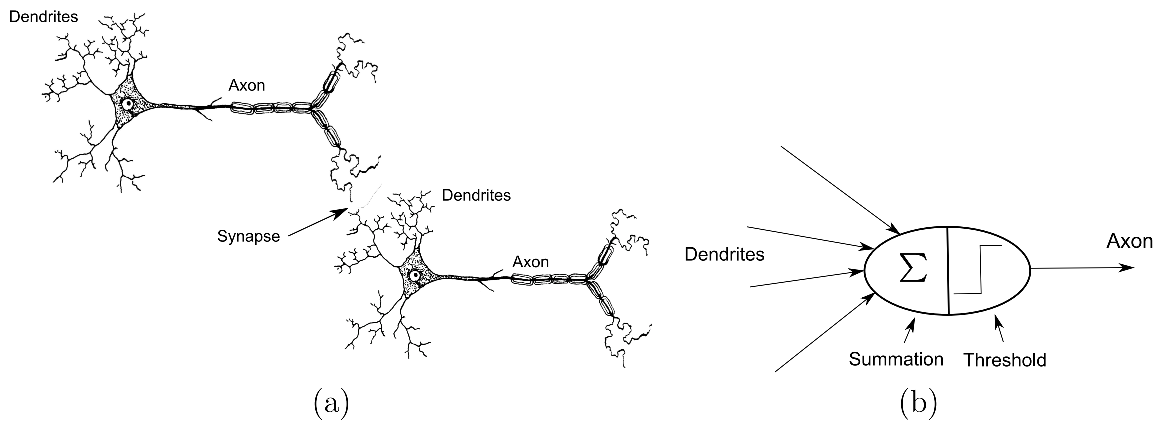


Figure 3.3: (a) Biological neuron components, (b) model used for the artificial neuron inspired by its biological counterpart

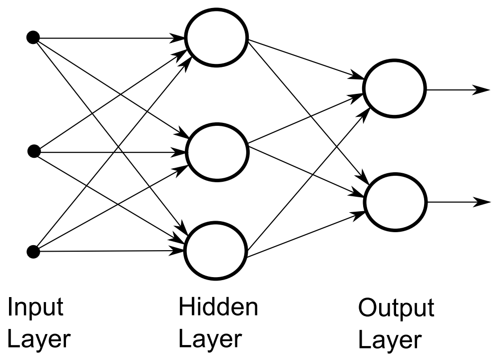


Figure 3.4: Artificial neural network representation

Artificial neurons were first proposed in 1943 by Warren McCulloch and Walter Pitts [90], but the widespread use of neural networks had to wait until the rediscovery of the backpropagation algorithm in 1986 [108]. ANNs are regarded as universal function approximators [56] that can generalise to novel input patterns without requiring to be exposed to all possible situations. Moreover, recurrent ANNs (networks that include feedback connections) can memorise previous input patterns to influence next outputs. Thus, ANNs can be used for solving non-Markov problems.

Genetic Algorithms Genetic Algorithms are a class of stochastic search methods inspired on biological principles of natural selection. John Holland proposed the idea of genetic algorithms as an abstraction of biological evolution in 1960 and later brought genetic algorithms into wider use after publishing his book 'Adaptation in Natural and Artificial Systems' [54]. In a genetic algorithm, the chromosome or genotype (a population of strings) encodes candidate solutions (phenotypes) that evolve toward better solutions over generations. Commonly, the evolution starts from a population of randomly generated individuals. In each generation, the fitness of every individual in the population is determined, a selection process chooses for reproduction the fittest individuals based on their fitness, and finally, new individuals are created by crossover and mutation from the mating pool in order to generate a new population. The new population will be used in the subsequent iteration of the genetic algorithm. The algorithm will terminate when either a maximum number of generations has been reached, or a satisfactory fitness level has been achieved for the population.

One fundamental difference between GAs and conventional search methods is that GAs sample simultaneously many points in the search space, thus most likely avoiding local minima and producing faster results in high-dimensional spaces. As was previously mentioned, the weights of an artificial neural network cannot be tuned by hand and thus algorithms like backpropagation appeared. GAs provide an efficient and fast search strategy for the design of neural networks. Thus, coming back to neuroevolutionary methods, the combination of a neural network and a evolutionary algorithm generates a system that can approximate any (differentiable) function, can generalise very well over the input space, with the possibility to include memory elements, and with an efficient methodology to evolve the optimal weights of the neural network.

Notably, neuroevolution methods work as well on partially observable and continuous spaces.

Evolution strategies Evolution Strategies (ESs) are a class of Evolutionary Algorithms (EAs) which use nature-inspired concepts like mutation, recombination, and selection applied to a population of individuals containing candidate solutions in order to evolve iteratively better and better solutions. These ESs were introduced by a (back then) unofficial workgroup on Evolution Techniques at the Technical University of Berlin in the late 1960s [107]. In contrast to genetic algorithms, which work with discrete domains, evolution strategies were developed to be used in continuous domains, which make them suitable for continuous-space optimization problems and real-world experiments.

CMA-ES CMA-ES is an advanced form of evolution strategy [110] which can perform efficient optimization even for small population sizes. It avoids random adaptation of the strategy parameters by adapting the covariance matrix at each step depending on the fitness values of the current population. The individuals are in this algorithm represented by n-dimensional real-valued solution vectors which are altered by recombination and mutation. Mutation is realized by adding a normally distributed random vector with zero mean, where the covariance matrix of the distribution is itself adapted during evolution to improve the search strategy. CMA-ES uses important concepts like derandomization and cumulation. Derandomization is a deterministic way of altering the mutation distribution such that the probability of reproducing steps in the search space that lead to better individuals is increased. A sigma value represents the standard deviation of the mutation distribution. The extent to which an evolution has converged is indicated by this sigma value (smaller values indicate greater convergence).

3.4 Conclusions

This chapter presented current knowledge in the area of human arm control. The focus was on reaching movements and especially the control mechanisms employed under novel environments. We have seen that the CNS uses concepts like *estimation* for dealing with noisy or partially observable information about the state of the body, *prediction* to anticipate consequences of motor actions given a context, and *impedance control* strategies to modify the mechanical stiffness of the arm to compensate for unpredictable environments. In particular, the human arm uses reactive mechanisms in a first interaction with a novel environment and employs the acquired knowledge in subsequent interactions in order to adapt the stiffness of the arm and generate appropriate internal dynamical models. Moreover, we have seen the most recent biological architectures for human arm control that propose a multiple set of predictor-controllers (or internal models) from which the CNS selects the most suitable for a given context using Bayesian-like inference techniques. These results shaped and inspired the architecture developed and proposed in this thesis for the predictive-based control of compliance of a robotic manipulator.

The chapter included as well an overview of evolutionary algorithms, especially focused on neurevolutionary techniques. As we have seen in the section of impedance control, a fundamental problem not solved yet is how the CNS chooses a task-appropriate set of impedance parameters from the theoretically infinite set of possibilities. Current studies postulate the use of optimality criteria and these results motivated the use in thesis of neuroevolutionary techniques for evolving optimal impedance controllers given a certain criteria and environment.

Chapter 4

PCAC - PREDICTIVE CONTEXT-BASED ADAPTIVE COMPLIANCE

This chapter aims at introducing the contributions of this thesis in the form of components of a robot control architecture, whose single components will be subsequently described in the following chapters.

4.1 Introduction

As we have seen in previous chapters, controlling the forces between the robot and the environment is at the same a crucial and complex task to achieve in previously unknown or unstructured environments. Impedance control, in any of its flavours, promises a way to control both position and forces within an unified control law by regulating the relationship between the forces and the motion errors. However, impedance control in its original description lacks of important qualities: first, uncertainty in the robot and/or the environment models might compromise the position tracking. Secondly, the original description fails to provide force-tracking capabilities and third, the robustness of the controller has to be guaranteed when dealing with unknown/uncertain environments. Last but not least, the selection of a proper target impedance given a current environment is a difficult enterprise. This thesis focuses on aspects of the abovementioned problems by using and combining different methodologies into a high-level Bayesian-based framework. As De Schutter [31] mentions on his excellent review on force control, future research needed to concentrate as well on high-level performance:

> "High-level performance. This level is (too) slowly getting more attention. It should make a force-controlled system robust against unmodeled events, using 'intelligent' force/motion signal processing and reasoning tools to decide (semi)autonomously and robustly when to perform control model switches, when to re-plan (parts of) the user-specified task, when to add active sensing, etc. The required intelligence could be model-based or not (e.g. neural networks, etc.)."

This work aims at focusing on that high-level performance, building a robotic manipulator system that is robust against uncertainties, using an intelligent level above the classical (adaptive) impedance controller, that is able to reason and select appropriate target impedance parameters given the properties of the current environment. Similarly, the system applies corrective actions in case of a wrongly estimated environment by making use of predictions of the sensory feedback to be received after self-generated actions. Moreover, the system is open to the use of learning techniques for acquiring knowledge about newly discovered environments. In recent years there has been an increasing effort towards trying to understand the neurophysiological mechanisms underlying human behaviour. Interestingly, both roboticists and neurocientists are interested on applying the current neuroscience knowledge into robotic systems. On the one side, scientists can easily reproduce and control experiments with the aim of understanding back the biological mechanisms [128][44]. On the other side, roboticists regard neuroscience results as a source of inspiration for the development of new control principles for robotic systems [11][8] with the ultimate goal of developing robots with the performance and adaptation skills observed in biological systems, especially when robots need to deal with real-world situations and unstructured environments.

The final experiment and indirect goal of this work is to control the interaction forces in a dual-arm robotic system. The learning of bi-manual co-ordination is crucial in human performance of everyday tasks, and some studies have highlighted the complexity underlying its development [134]. The use of bimanual operations is an interesting capability to be implemented in a robotic system especially in two situations: (a) when there is the intention to manipulate heterogeneous objects of possible deformable nature. In such environments, a bimanual operation might be indispensable to successfully handle the objects; and (b) the intention to manipulate big and/or heavy objects, for which a dual-arm strategy might be the only solution to transport them.

Yet although there is an increasing interest on providing robots with the capability of performing complex manipulation actions and some very promising and spectacular results have been reported [98][4], we are still far from reproducing the manipulation skills of a 3-year old child, as previously indicated in Chapter 1. In neuroscience there are three main approaches for explaining how the coordination of two arms works in biological systems:

The use of classical theories on behaviour of dynamical systems [71]. This theory postulates that biological systems are composed of many subcomponents where behaviour is not dictated by a superior command but emerges as a consequence of cooperation among those subcomponents

The use of information processing theories [121][14]. This approach exploits the idea of neural cross-talk between neural controllers. In a bimanual movement and due to the limited neural resources the commands sent to an arm appear also in the contralateral arm by means of structural interference. That is, the left arm receives an attenuated mirror image of the commands that were sent to the right hand (in case of a right-handed person). The focus of this theoretical framework has been on studying the limitations of bimanual tasks due to these interferences as the task can only be successful if the neural interference can be suppressed (usually as a result of practice)

More recently, the current neuroscience hypotheses used for explaining single-hand movements are also formulated for the case of bimanual operations [134]. This approach postulates the use of internal models by the Central Nervous System (CNS), in other words, that the CNS runs internal models that relate sensory information to a given motor command (forward models [129]) and motor commands to a given sensory information (inverse models [136]). The ability to learn, acquire, and use new internal models is a key factor for the motor system to cope with the variety of daily objects to interact with. Recently some theoretical computational models have been also presented that incorporate such internal models in the lowest layer of a multilayer control system [135].

Inspired by these studies, Figure 4.1 shows the Predictive Context-Based Adaptive Compliance (PCAC) architecture proposed in this thesis for controlling the robot's compliance via context prediction. The core element is an interaction controller in the form of an impedance controller that actively regulates the robot's compliance with regard to the environment. The first component is a Bayesian-based estimator that based on interaction selects the most likely contacted environment among a set of predefined sets of environments that have been previously learned. Chapter 5 describes

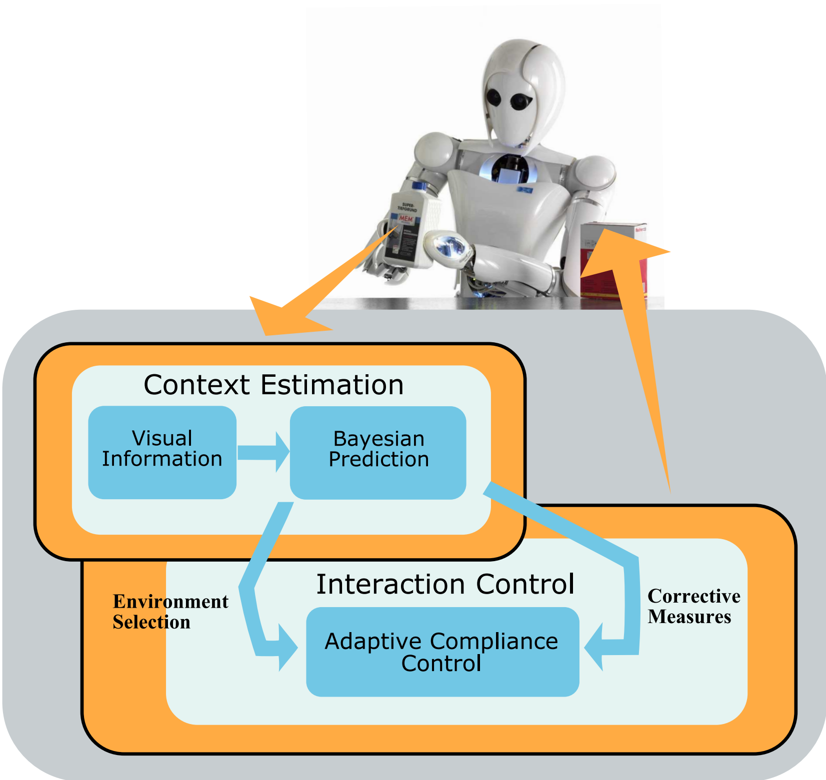


Figure 4.1: Predictive Context-Based Adaptive Compliance (PCAC) architecture proposed in this thesis

the design and experiments of this first component. As stated in previous chapters, the selection of the parameters of an impedance controller given a certain environment does not posses any clear methodology yet. For that reason, Chapter 6 investigates the use of neuroevolutionary algorithms for evolving the parameters of an impedance controller, proposing a methodology and criteria to select optimal parameters. Moreover, the resultant impedance controller shows robustness and can be easily evolved to incorporate force-tracking capabilities, a quality not present on the original description of the impedance controller. Finally, Chapter 7 describes a Bayesian predictor used to predict the sensorimotor consequences of self-generated robot actions, in order to apply corrective measures in case of wrongly estimated environments by using a multi-instance impedance controller. The Bayesian predictor will make use of visual information in order to discriminate with a certain probability among a set of discrete contexts. Moreover, the experiments used in this final chapter combine the three previously described thesis contributions to form the PCAC architecture shown in Figure 4.1.

4.2 Environment identification to adapt to a changing environment

One of the major drawbacks hindering the widespread deployment of impedance control schemes is the fact that its performance and reliability is directly compromised by the quality of the information with regard to the environment properties. The approach taken in this thesis is to use an identification method that allows the impedance controller to switch between different previously-tuned impedance controllers for a set of environments. The robot is first trained for a set of environments, which can be anytime expanded with information about additional environments. The information gathered about the environment will serve to tune a multi-instance impedance controller (that is, a common impedance control law whose parameters can be modified on-the-fly). These steps are considered to happen 'off-line', that is, in a training phase before the robot's real deployment in a certain scenario. In the 'on-line' phase, the robot will make a test contact with the environment in order to perform a single-trial identification of the environment, similarly to what it is seen in biology, where a first contact basically makes use of reflexive actions and serves to create a first predictive model of the environment for its use in subsequent contacts (see section 3.2.6). The most likely environment will be selected and thus the proper impedance controller parameters for it.

In case of new unknown (not seen before) environments, the system has two possibilities:

(a) if there has not been a training phase for that environment, the robot will eventually select the most similar environment within the set of trained environments. In this case, a minimum threshold for the likelihood of a given environment can be established that would request for a new training phase in case of an environment being very dissimilar to all (known) others.

(b) a new training phase can be started in order to 'learn' the properties of the environment and expand the knowledge about environments.

Figure 4.2 shows a scheme of the identification module used to discriminate among different environments. In this figure, $\boldsymbol{d}$ is the data vector, composed of the vectors $\boldsymbol{f}$ of contact forces and the vector $\boldsymbol{x}$ of robot positions. During a training phase, the likelihood $p(d|E_j)$ for each environment $E_j, E_j \in E$ is computed that defines the probability of sensory data vector $\boldsymbol{d}$ occurring given that the hypothesis of contacting the environment E_j is true. The prior probability $p(E_j)$ is assigned so that all environments have the same a priori probability to be contacted. In the on-line phase, after a contact with the environment, the posterior (or final) probability $p(E_j|d)$ will be computed and the maximum among all the environments will be chosen as the contacted environment. This result will be used to determine the impedance controller instance to be selected. Chapter 5 will deal with the Bayesian formulation used for identifying the properties of the environment.

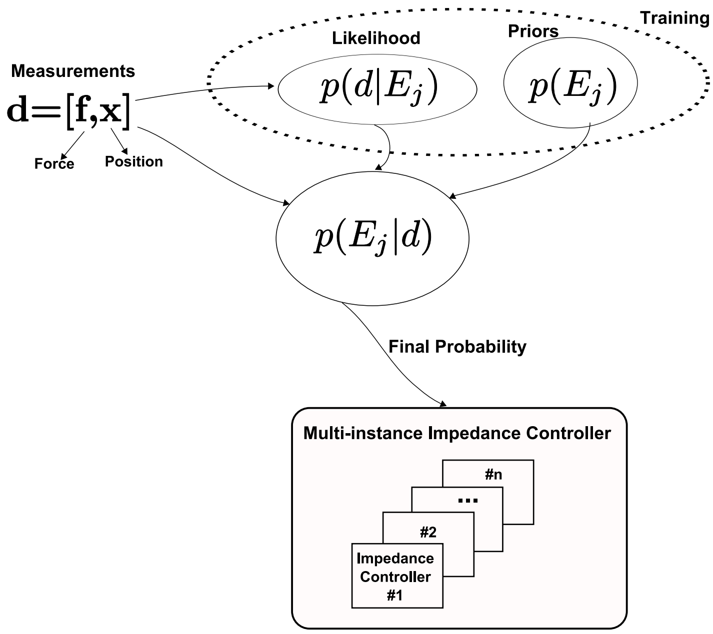


Figure 4.2: Bayesian-based environment identification for the selection of the impedance controller

4.3 Neuroevolutionary techniques to evolve an adaptive impedance controller

As soon as good knowledge about the environment can be assumed (regardless of the method used for obtaining that information), another source of difficulty when using impedance controllers in real scenarios is the search for optimal parameters that will generate a robust and stable impedance controller. Due to the similarity of the impedance control formulation with a second-order system, methods from classical control theory can be used to select a proper damping factor and system frequency of the close-loop system. It is however not so immediate to select which is the proper system frequency or damping ratio for a specific environment. Moreover, the final system needs (in most applications) to ensure contact stability as some scenarios (mainly industrial) might not allow oscillations (not even one) during the contact.

In order to provide with a simple and automatic methodology to select optimal parameters for the impedance controller, this thesis looked at neuroevolutionary methods for solving this task. Neuroevolution is the combination of neural networks (which will determine the structure of the controller) and evolutionary algorithms to search for optimal weights of this neural network (which will determine the control law or policy). Neuroevolutionary methods are used as methods to solve learning tasks, especially those which are stochastic, partially observable, and noisy.

In this thesis, neuroevolutionary methods will be used to find optimal parameters for the impedance controller. The criteria to optimise (the 'fitness' function) has been defined as to search for solutions that fullfil two measures: (a) minimisation of the force error between a given reference and the measurements and (b) ensure that the contact

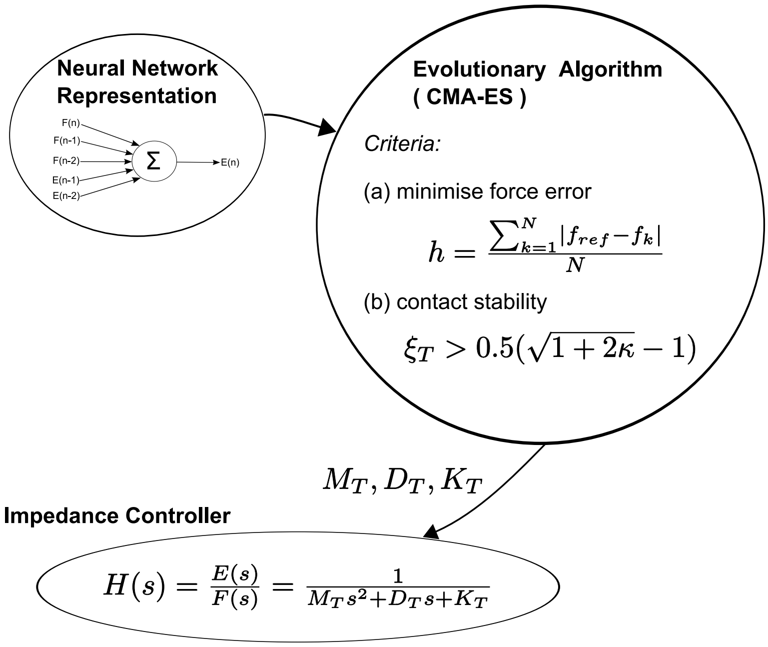


Figure 4.3: Scheme of the process used to evolve the parameters of the impedance controller using neuroevolutionary techniques

with a given environment will be always stable. Thus, the parameters of the controller are found by the evolutionary algorithm for a certain environment. Additionally, stability criteria has been introduced in the search in order to select controller parameters that will ensure stability as well as provide with force-tracking capabilities. Figure 4.3 shows a scheme of the process to obtain the set of parameters of the impedance controller given the properties of an environment. First, the impedance controller is represented as an artificial neural network. Second, the weights of this neural network are evolved by using the evolutionary algorithm (in this case, CMA-ES -see Chapter 3-) that uses the two previous criteria as fitness function to guide the search. The algorithm will make use of a model of the robot as well as of the chosen environment. The output of the algorithm are the parameters of the impedance controller (M_T, D_T, K_T). The details of using neuroevolutionary techniques for evolving optimal, stable, and force-tracking impedance controllers will be given in Chapter 6.

4.4 Context estimation to anticipate sensorimotor consequences from self-generated actions

A higher level of autonomy might be achieved if the robot would be able to predict the context on which it is currently being deployed. Although the word 'context' is a very much used and popular word in many areas, it is not easy to find a definition that satisfies a large number of different disciplines [7]. A widespread definition that suits the use given to the word in the present work is:

> *Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*[33].

The following definition also suits the meaning given in this text as well as its subsequent use as predictive mechanism:

> *A context is any identifiable configuration of environmental, mission-related, and agent-related features that has predictive power for behavior*[124].

Regardless of the specifics of the definition, this context might be, in a manipulation task (regardless of being performed by a robot or a human), described as being discrete. The typical example is the one that involves lifting a carton of milk. In terms of the context, and in order to successfully solve this task, prior to lifting the carton of milk, a human would generate some internal probabilities regarding whether the carton might be full or empty [136]. The prediction would serve two purposes: on the one hand, to pre-regulate the arm's compliance and determine the forces to exert, and on the other hand, to anticipate the sensory consequences of the chosen action. The latter would allow the initiation of corrective measures in case of a false predicted context. Imagine that the human predicted a full carton of milk and once he begins to lift it, there is a big discrepancy between the expected arm speed and the actual one. This error between predicted and actual sensory feedback would immediately be translated to a corrective measure to reduce the speed of motion.

One area of neuroscience research concentrates on the underlying processes controlling the arms in so-called 'asymmetric' tasks, that is, where there is a dominant arm that actually performs the action, and a non-dominant arm that just 'helps' to achieve it. A classical experiment used for clarifying the underlying control processes is the so-called Waiter Task [57][35]. This experiment allows to describe the predictive mechanisms controlling the unloading of objects from a tray with one hand (dominant) while the other hand (non-dominant) holds and keeps the tray and arm at a fixed posture. The results evidence that both the position of the tray and the non-dominant arm remain unchanged despite the changes on weight and forces during the unloading. From these observations, two conclusions about the control of the arm can be extracted:

(a) first, the fact that the arm's *compliance* is regulated in order to adapt to the *changing context* and

(b) the *predictive* nature of the anticipatory actions to be performed.

The latter comes from the observation that the non-dominant arm posture would change if an external person unloads the objects from the tray. Yet even if the person holding the tray is informed by some signal that the tray is going to be unloaded, the holding tray will move. These results indicate that a prediction mechanism anticipates the consequences of our own self-generated actions and the motor system acts accordingly to synchronise the change in compliance of the arm with the object's removal. It is known that sensory feedback signals in biological systems suffer from considerable delays. For instance, visual feedback can take around 100-200ms to be processed and 'fast' spinal-cord feedback loops need around 30-50ms [68]. Clearly, these delays would not permit an immediate reaction to a sudden load change. When the information arrived at the CNS, it would be already too late to react, and that is what it is observed in the case that an external person removes the loads from the tray: the reaction comes, but too late to hold steadily the tray. In the robotics domain, active compliance is obtained by using sensory feedback and a control algorithm that creates a virtual mechanical impedance for the robot. The joint positions or torques are regulated in order to achieve a desired compliant behaviour with the environment. Yet one of the main limitations of this technique are feedback control constraints which might lead to unstability [72][15].

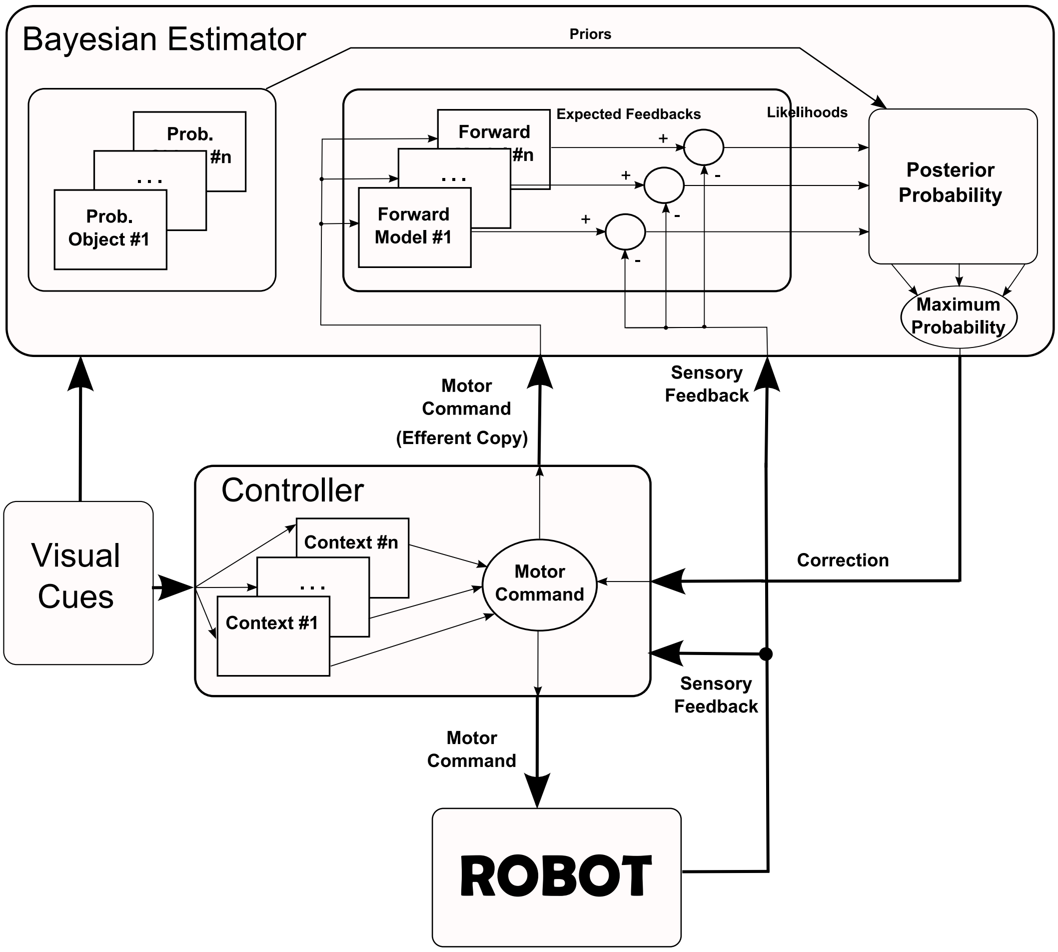


Figure 4.4: Bayesian prediction for compensation of erroneous initial predictions about the nature of an object

Figure 4.4 shows the scheme proposed in this thesis which is inspired by the previous studies. The robot uses visual cues to generate a prior probability for the most probable object that lies in front of it. This information sets the priors on the Bayesian model. Using forward models learned after training with several objects, the robot generates expectations for the sensory feedback after issuing a certain action. The action is performed onto the real robot and, simultaneously, an efferent copy [1] of that action is sent to the input of the forward models. If the sensory expectations meet the current sensory feedback, no measures need to be taken. If otherwise, the Bayesian model indicates with a high probability that the sensory feedback belongs to a different object to the one predicted, the necessary correction actions are undertaken in order to change the controller's behaviour according to the most probable object. Chapter 7 will deal with the implementation of the predictive component as well as the tests of the complete architecture using a real dual-arm robot manipulator.

[1] In biology, an efferent copy is a copy of the issued motor command that is used to predict the sensory consequences of a given action. The German physicist Hermann von Helmholtz (1821-1894) proposed for the first time the idea of efferent copies

Chapter 5

CONTACT IMPEDANCE ADAPTATION VIA ENVIRONMENT IDENTIFICATION

In this chapter a Cartesian impedance controller is synthesized to regulate the interaction forces between a robotic arm and the environment. A major feature is a robust identification module based on Bayesian inference that allows the impedance controller to adapt to different environments.

The chapter concludes showing the simulated results on a two-link planar arm as well as the experimental results of applying this method on a seven degree-of-freedom robot manipulator.

5.1 Introduction

For a robot manipulator to interact in a safe and human-friendly manner in unknown environments, it is necessary to include an interaction control method that reliably adapts the forces exerted on the environment in order to avoid damages both in the environment and in the manipulator itself. As we have seen, a force control method, or strictly speaking, a direct force control method, can be used on those applications where the maximum or the desired force to exert is known beforehand. In some industrial applications the objects to handle or work with are completely known as well as the precise moment on which these contacts are going to happen. In a more general scenario, such as one outside a well-defined robotic workcell, sometimes neither the objects nor the time when a contact is occurring are known. In such case, indirect force control methods find their niche. These methods do not seek to control maximum or desired force, but they try to make the manipulator compliant with the object being contacted. The major role in the control loop is given to the positioning but the interaction is also being controlled so as to ensure a safe and clear contact. In case contact's interaction forces have exceeded the desired levels, the positioning accuracy will be diminished to account and take care of the (at that moment) most important task: the control of the forces.

Impedance control [50] is one of these indirect force control methods. Its aim is to control the dynamic behaviour of the robot manipulator when contacting the environment, not by controlling the exact contact forces but the properties of the contact, namely, controlling the stiffness and the damping of the interaction. Moreover, the steady-state force can be easily set to a desired maximum value. The main idea is that the impedance control system creates a virtual new impedance for the manipulator, which is being able to interact with the environment as if new mechanical elements had been included in the real manipulator.

As we saw in Chapter 2, one of the major drawbacks of impedance control is that it was originally designed for controlling forces in a known environment. There have been posterior approaches to deal with uncertainties in the environment and robot models, but the topic that has not been successfully closed so far, according to the vast amount of literature that can be found with regard to this problem. This chapter proposes a Bayesian inference-based estimation algorithm that outputs the most likely environment that has been contacted among a set of possible environments. Using this information, the impedance controller can optimally adapt to the current situation. Bayesian inference is a statistical method where the probabilities of an event happening are continuously updated as new evidence is available. That is, Bayesian inference will start with a a prior probability representing the knowledge about the occurrence of an event before any evidence has been gathered, and a likelihood function representing the probability that a particular evidence is observed given that a particular hypothesis is true. The Bayesian estimator will then generate a new final probability after an evidence is observed, that updates the current probability.

In this chapter we will first design, synthesize, and prove the stability of a Cartesian impedance controller. A simple model of a two-link robotic arm will be used to prove and demonstrate the concepts. This theoretical analysis will be then ported and tested in a real robot: a Mitsubishi PA-10 manipulator with seven degrees of freedom. In that real scenario, the identification module based on Bayesian inference that estimates the current environment will be tested in order to provide the impedance controller with augmented information about the nature of the current environment. That is to say, this chapter focuses on the two components highlighted in Fig. 5.1, the Bayesian-based estimator that selects the most likely environment and the synthesis of the impedance controller. The chapter will finalise by presenting and discussing the results of that combined method.

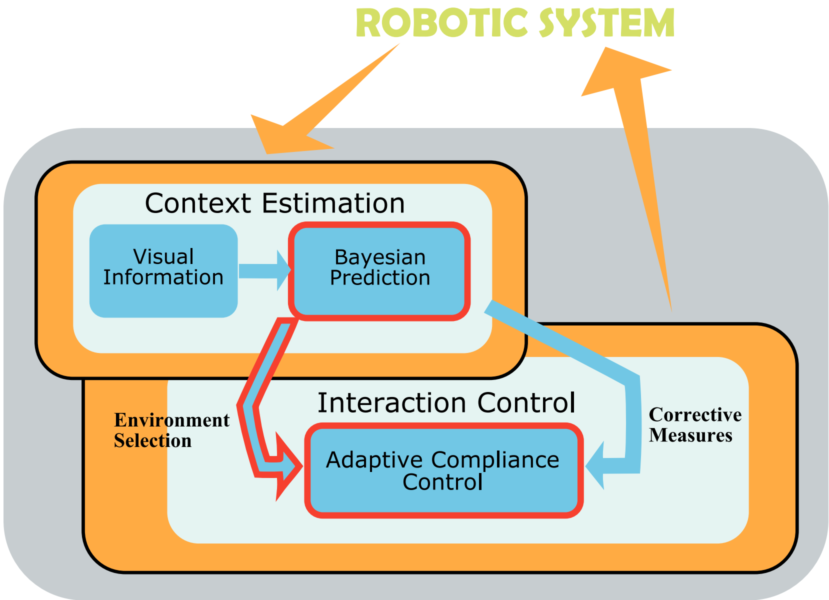


Figure 5.1: Proposed architecture for compliance control via context estimation. The components highlighted are discussed in this chapter

5.2 Impedance controller

Chapter 2 presented several schemes for active interaction control and, more specifically, for the implementation of impedance control strategies. Given the fact that the focus of this work is on industrial-like robots, whose internal controllers are not accessible (without major redesign of a new robot controller), we chose to use a position-model-error-based impedance controller (Fig. 2.9) that builds up upon the existing robot position controller.

5.2.1 Controller synthesis

The synthesis of an impedance controller requires two initial decisions: the selection of an appropriate model for the target impedance and the tuning of the parameters of that model in order to fulfill the desired contact properties. The first point is usually not much discussed in literature, as most of the works select a second-order target model due to its well-known behaviour and the methods available to study it. On the other hand, the second point, the tuning of model parameters is most of the times done empirically as there is no common methodology on how to select those parameters. Chapter 6 will propose the use of evolutionary techniques to select optimal parameters for the impedance controller. In this chapter, we propose the use of a Bayesian-based estimator for the current environment, whose result will help selecting the properties of a multi-instance impedance controller that has been previously tuned for a set of possible environments.

5.2.2 Open and closed loop behaviour

The control scheme on Fig. 2.9 can be represented in the form of a signal-flow graph as seen in Figure 5.2. Making use of Mason's Rule, the properties of the complete control system in open and closed loop can be easily analysed:

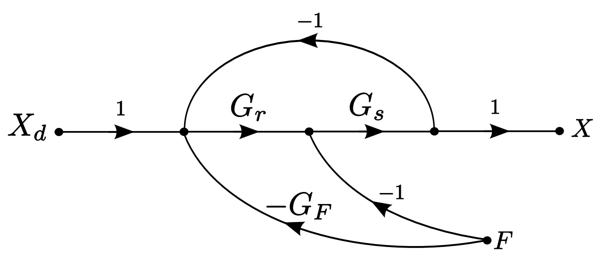


Figure 5.2: Signal-flow graph of the control scheme in Fig. 2.9

The output of this system is the position X. As inputs we can consider the reference position X_d and the contact force F. Thus we can define the transfer functions $H(s)$ and $T(s)$ to describe the relation between the current position and the reference position ($H(s) = \frac{X(s)}{X_d(s)}$), and the relation between the current position and the contact force ($T(s) = \frac{X(s)}{F(s)}$). That leads to the relation:

$$X(s) = H(s)X_d + T(s)F \tag{5.1}$$

Mason's rule was developed to find out the transfer function of a control scheme represented as a signal-flow graph. The general gain formula is formulated as:

$$G = \frac{Output}{Input} = \frac{\sum_{k=1}^{N} G_k \Delta_k}{\Delta} \tag{5.2}$$

$$\Delta = 1 - \sum L_i + \sum L_i L_j - \sum L_i L_j L_k + ... + (-1)^m \sum ... + ... \tag{5.3}$$

where G is the transfer function between output and input, Δ is the determinant of the graph, N is the total number of direct paths between input and output, G_k is the gain of the k_{th} direct path, Δ_k is the cofactor value of Δ for the k_{th} direct path, with the loops touching the k_{th} direct path removed. L_i is the loop gain of each closed loop, L_iL_j is the product of loop gains of any two non-touching loops, $L_iL_jL_k$ is the product of the loop gains of any three pairwise non-touching loops and so on.

Using Mason's rule to find out the transfer function $H(s)$, we obtain: $G_1 = G_rG_s$, $L_1 = -G_rG_s$, $\Delta_1 = 1$, and $\Delta = 1 + G_rG_s$. Thus, the transfer function $H(s)$ is defined as:

$$H(s) = \frac{X(s)}{X_d(s)} = \frac{G_rG_s}{1 + G_rG_s} \tag{5.4}$$

Similarly, using Mason's rule for the transfer function $T(s)$, we obtain: $G_1 = -G_s$, $L_1 = -G_rG_s$, $\Delta_1 = 1$, $G_2 = -G_FG_rG_s$, $L_2 = -G_rG_s$, $\Delta_2 = 1$. Thus, the transfer function $T(s)$ is defined as:

$$T(s) = \frac{X(s)}{F(s)} = \frac{-G_s}{1 + G_rG_s} + \frac{-G_FG_rG_s}{1 + G_rG_s} = \frac{G_s(-G_FG_r - 1)}{1 + G_rG_s} \tag{5.5}$$

Following Eq. (5.1), the complete closed-loop transfer function of the system shown in Fig. 2.9 is described as:

$$X = G_rG_s\left[1 + G_rG_s\right]^{-1} X_d - G_FG_rG_s\left[1 + G_rG_s\right]^{-1} F - G_s\left[1 + G_rG_s\right]^{-1} F \tag{5.6}$$

Let's define common terms found in Eq. (5.6), to rewrite it in a form which allows to clearly see which are the components of the closed-loop response:

$$X = M(s)X_d(s) - G_F(s)M(s)F(s) - G_s(s)D(s)F(s) \tag{5.7}$$

where $M(s) = G_rG_s\left[1 + G_rG_s\right]^{-1}$ and $D(s) = \left[1 + G_rG_s\right]^{-1}$.

If the loop is not closed using the impedance control block (that is, $G_F = 0$), the system's transfer function becomes the open-loop transfer function:

$$X = M(s)X_d(s) - G_s(s)D(s)F(s) \tag{5.8}$$

Eq. (5.8) is the inherent system's transfer function of the robot we originally receive, i.e. is the industrial-like robot that is given to us, whose internal controllers are designed and tuned to fulfill their task: provide a high-accurate positioning. For that reason, we can certainly assume that the value of $M(s)$ will have been chosen as to be follow as close as possible the signal $X_d(s)$ by making $M(s) \approx I$. Similarly, for such a position-controlled system, the forces F are considered disturbances that need to be rejected. The industrial-like robot is generally design to follow with high gain the position commands and to fiercely reject any contact force that acts against its movement (reason for its hazardous nature). That means that the value of $D(s)$ will have been certainly chosen as to minimise the effect of contact forces F, that is, $D(s) \approx 0$. Given those facts, it is clear to see that the impedance controller G_F is not affecting the tracking performance of the existing internal controller in case that the robot is moving on free space. In a contact task, the extra term in Eq. (5.7) shows that G_F will just 'shape' the close-loop transfer function to achieve a compliant robot by modifying the relation between the position error and the external contact forces.

Commonly, the impedance controller block (G_F) in Fig. 2.9 is modelled as an target admittance:

$$G_F(s) = Z^{-1} = M_T(s) + D_T(s) + K_T(s) \tag{5.9}$$

where M_T, D_T and K_T are the inertia, damping and the stiffness coefficients, respectively. Figure 5.3 shows the structure of the impedance controller. M_T, D_T and K_T will define the dynamic behaviour of the robot that could be compared to the effect of including physical masses, springs, and dampers on the robot.

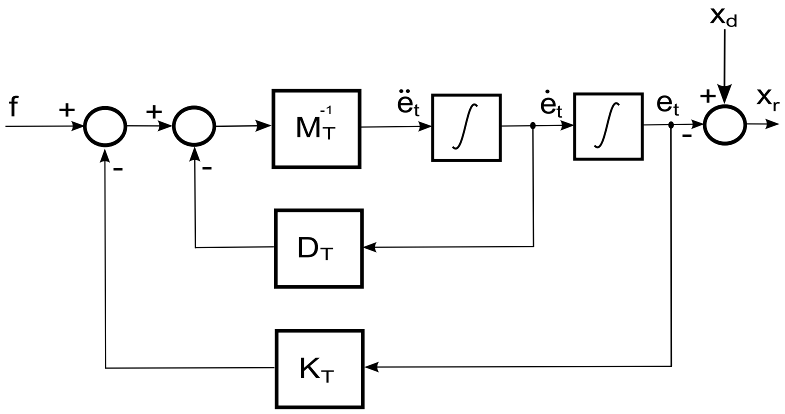


Figure 5.3: Representation as a control scheme of the impedance controller described by Eq. (5.9)

From Chapter 2, Eq. (2.11), we know that the position model error (e_p) of the position-model-error-based impedance control scheme that we are using is defined as:

$$e_p = x_d - x - G_F F \tag{5.10}$$

That means that the desired impedance model error e_p will be achieved when two conditions are met:

(a) a well-designed impedance controller G_F that compensates for the contact forces F, and

(b) a stiff robot with a high-gain position controller than is able to achieve $x_d = x$

Those conditions will lead to $e_p = 0$ in an ideal case. Needless to say, real systems will have problems complying with both conditions. First, the impedance controller is dependent on the performance of the internal controller. And the real internal controller will not have an unlimited bandwidth, what might lead to inaccuracies on the position tracking, especially at high speeds, and ultimately, to inaccuracies on the tracking of the target impedance. These problems lead us to the next section concerned with the assessment of the stability of the control system in presence of small errors between desired and realized target impedances that inevitably will appear in a real system.

5.2.3 Stability

Previous sections undertook the design of the impedance controller and its effects on minimising positioning and impedance control errors. Though certainly the above presented formulae describes with high fidelity the ideal performance of including an impedance controller closing the loop of an existing position controller, it will undoubtedly show a lower performance in its real application. One critical point that has not been explicitly tackled is related to the interaction with the environment and the effects that the exchange of energy between robot and environment will have on both. It is thus required that the analysis of the system includes the environment model (as good as it can be estimated) in order to ascertain which are the stability margins and how to ensure that a certain controller will be always stable while interacting with a certain environment.

Stability generally means that system output will be not significantly modified due to small changes in the system input, in the initial conditions of the system, or by small changes on its parameters. A linear time-invariant system is considered stable if two conditions are observed: (a) a bounded input signal causes an output response that is also bounded, and (b) in the absence of any input signal, the output converges towards the equilibrium point (usually zero), regardless of the initial conditions of the system. Although most real processes are inherently stable (though exceptions might be found in certain areas, like in chemical reactors), the critical factor is that the feedback controller used to control the process is the one that might actually cause a system that was inherently stable to become unstable. In this section, we will analyse the coupled stability, that is, the behaviour of the impedance controller coupled with the environment as seen in the descriptive diagram on Fig. 5.4, where $G_e(s)$ is the model of the environment, $G_F(s)$ is the target impedance, p is defined in literature as the penetration, that is, the distance that the robot penetrated the environment, and p_d is the rest position of the environment.

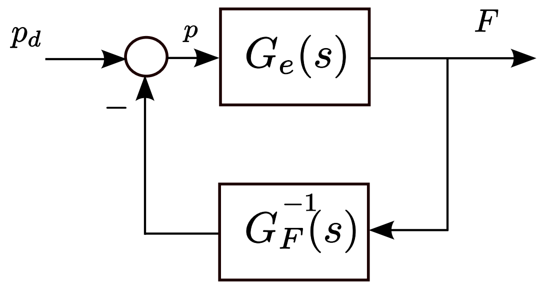


Figure 5.4: Descriptive diagram representing the coupled system (impedance controller - environment) to be evaluated

In order to prove the stability of the system, we need to analyse the steady-state values for p, e, and F, which are commonly denoted as p_{ss}, e_{ss}, and F_{ss}, respectively. Needless to say, we need first to compute the coupled transfer function for the system shown in Fig. 5.4. Assuming a general environment model as

$$-F = G_e(s)p(s) \tag{5.11}$$

where $p(s) = p_d - p$, we can use this relation in the desired target impedance model $-F = G_F(s)e(s)$, where the error e for the impedance controller is actually equal to the penetration p from the viewpoint of the environment. That is, $G_e(s)p(s) = G_F(s)e(s)$, which after using $p(s) = e(s)$ and isolating $p(s)$ will lead to the value for the penetration p in the coupled environment/controller system.

$$p(s) = \left[G_e(s) + G_F(s)\right]^{-1} G_F(s)p_d(s) \tag{5.12}$$

Using this result back to Eq. (5.11), we can obtain the value for the force F for the coupled environment/controller system:

$$-F(s) = G_e(s)\left[G_e(s) + G_F(s)\right]^{-1} G_F(s)p_d(s) = G_e(s)p_d(s)\left[I + G_e(s)G_F^{-1}(s)\right]^{-1} \tag{5.13}$$

Finally, we can similarly compute the value for the error $e(s)$ of the coupled environment/controller system:

$$e(s) = \frac{-F}{G_F(s)} = \frac{G_e(s)p_d(s)\left[I + G_e(s)G_F^{-1}(s)\right]^{-1}}{G_F(s)} = G_eG_F^{-1}(s)\left[I + G_e(s)G_F^{-1}(s)\right]^{-1} p_d(s) \tag{5.14}$$

From the previous relationships, it is possible now to find the equilibrium or steady-state points of those equations. From Eq. (5.12), we can compute the steady-state value p_{ss} by analysing it when s tends to zero, that is, $p_{ss} = p(s)_{s=0}$.

$$p_{ss} = \left[G_e(0) + G_F(0)\right]^{-1} G_F(0)p_{dss} \tag{5.15}$$

where p_{dss} denotes the steady-state value for the reference penetration.

If we assume an impedance target model described as

$$G_F(s) = M_T(s) + B_T(s) + K_T \tag{5.16}$$

it is clear to see that

$$G_F(0) = K_T \tag{5.17}$$

Similarly, regardless of the order of the environment model, the steady-state value will be given by

$$G_e(0) = K_e \tag{5.18}$$

Using these two relations, we can simplify previous equation Eq. (5.15) as:

$$p_{ss} = \frac{K_T}{K_e + K_T} p_{dss} \tag{5.19}$$

From Eq. (5.13), we can compute the steady-state value F_{ss} by analyzing it when s tends to zero, that is, $F_{ss} = F(s)_{s=0}$.

$$F_{ss} = \frac{K_e}{I + \frac{K_e}{K_T}} p_{dss} \tag{5.20}$$

From Eq. (5.14), we can compute the steady-state value e_{ss} by analyzing it when s tends to zero, that is, $e_{ss} = e(s)_{s=0}$.

$$e_{ss} = \frac{\frac{K_e}{K_T}}{I + \frac{K_e}{K_T}} p_{dss} \tag{5.21}$$

Stability Proofs Several criteria can be used to assess the stability of the robot-environment interaction after introducing an impedance controller in the feedback loop as in Fig. 2.9.

- The interaction can be said to be stable if the robot using impedance control contacts a passive environment and the steady-state values for p_{ss} (Eq. (5.19)) and F_{ss} (Eq. (5.20)) are both stable (in the sense of Lyapunov, meaning that any steady-state solution of the dynamical coupled system will stay near to this equilibrium for infinite time or even converge to it).

- Given the fact that we are using an industrial-like robot, with internal controllers, we can assume that both the controller (G_r in Fig. 2.9) and the robot model (G_s in Fig. 2.9) are stable. In this case, the stability of the coupled robot/environment can be assessed by looking at the stability of the feedback loop in Fig. 5.4. Mathematically, the stability of a closed-loop linear system can be determined by analysing the poles of the characteristic equation, that is, the roots of the denominator of the closed-loop transfer function.

  The closed-loop transfer function is easily extracted as

  $$\frac{F(s)}{p_d(s)} = \frac{G_e(s)}{1 + G_F^{-1}(s) G_e(s)} \tag{5.22}$$

  If the previous transfer function in Eq. (5.22) is stable, we can also assure that the coupled system is stable. The stability of Eq. (5.22) can be satisfied for an ideal environment if and only if there are no poles in the right half plane S.

- Passivity theory can also be used to prove the stability of the coupled system. Passivity, or the theory of positive real systems, originated in the analysis of networks and has been also lately being used to determine the stability of control systems. The main idea behind designing passive (or positive real) controllers comes from the fact that a stable passive real system will remain stable when connected in a negative feedback loop to a controller that is positive real, irrespective of system variations as long as the system remains positive real.

  A function $G(s)$ is denoted as positive real if meets two conditions:

- $G(s)$ is real when s is real
- $Re\{G(s)\} \geq 0$ for all s such that $Re\{s\} \geq 0$

Thus, if the environment transfer function $G_e(s)$ is positive real then a necessary and sufficient condition to ensure stability of the coupled environment-controller system is that the impedance controller is also passive, that is, positive real. If we assume an impedance target model as the second-order system described by Eq. (5.9), the previous condition implies that inertia, damping, and stiffness matrices need to be positive real to satisfy the passivity stability criterion.

5.3 Modeling

This section describes the models used for the simulation experiments of this chapter, particularly, of the modules that compose the general control scheme. The equations describing the robot dynamics, the impedance controller, and the environment model are formulated. Finally, the development of a model-based dynamic controller used for compensating for the non-linearities of the robot dynamical model is also described.

5.3.1 Control scheme

Figure 5.5 shows the complete control scheme corresponding to the scheme presented in Figure 2.9, known as position-model-error-based impedance control. The control scheme is composed of the following submodules: Trajectory Generation module, Impedance Controller, Direct and Inverse Kinematics modules, Dynamical Model-Based Controller module, Two-link Arm Dynamical Model, and Environment model.

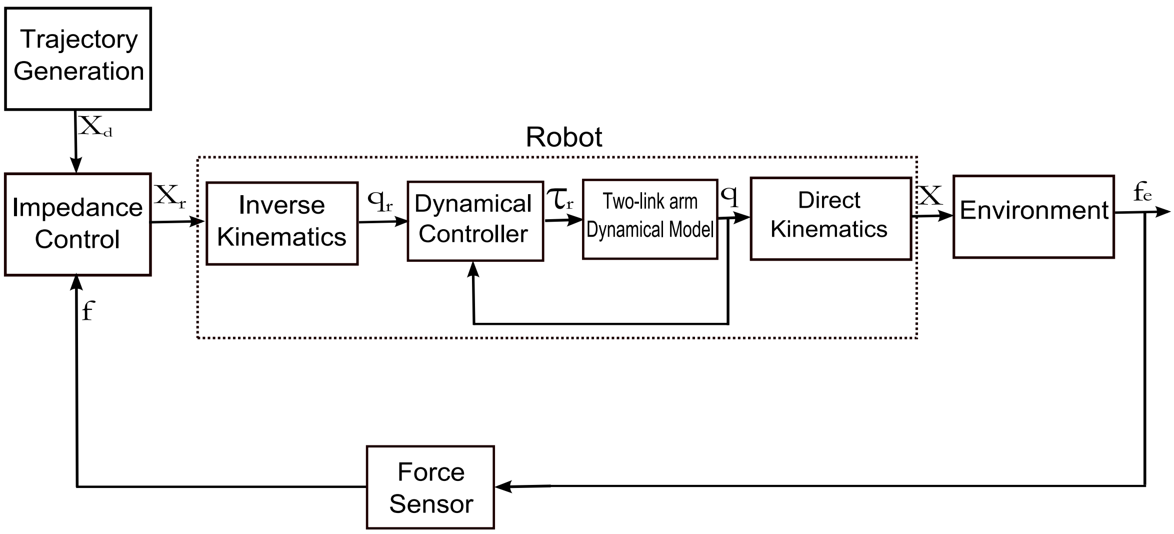


Figure 5.5: Robot's Cartesian position control scheme

5.3.2 Robot Dynamics

The dynamic model of a robot manipulator relates the forces acting on the mechanical structure with the resulting displacements, velocities and accelerations. These forces can have different sources: the torques delivered by the motors, the inertia of the mechanical links, the gravity, Coriolis and centripetal forces, the friction forces and the possible forces exerted for the environment on the robot. Given an initial state of the mechanical structure and the time history of torques $\tau(t)$ acting at joints, the direct dynamic model allows to predict the resulting motion $\theta(t)$ (and its derivatives) in joint space. With this information and the direct kinematic model, a prediction of the trajectory $x(t)$ in Cartesian coordinates can be performed.

The dynamic model of a n-joint robot manipulator can be written in the Lagrangian form as

$$M(\theta)\ddot{\theta} + B(\theta, \dot{\theta}) + G(\theta) = u, \tag{5.23}$$

where θ is the joint variable n-vector and u is the vector of generalized forces acting on the robot manipulator. $M(\theta)$ is the inertia matrix, $B(\theta, \dot{\theta})$ are the Coriolis/centripetal forces, and $G(\theta)$ is the gravity vector. In Equation (5.23) we are not taking into account the friction torques that are always to be found in a real robot manipulator. For the sake of simplicity, our simulation experiments will use a dynamic model of a two-link planar arm. The dynamical equations following the Lagrangian formulation [84] are:

$$M(\theta)\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + B(\theta, \dot{\theta}) + G(\theta) = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{5.24}$$

where

$$B(\dot{\theta}, \ddot{\theta}) = \begin{bmatrix} -m_2 a_1 a_2 (2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^{\,2}) \sin\theta_2 \\ m_2 a_1 a_2 \dot{\theta}_1^{\,2} \sin\theta_2 \end{bmatrix} \tag{5.25}$$

$$G(\theta) = \begin{bmatrix} (m_1 + m_2) g a_1 \cos\theta_1 + m_2 g a_2 \cos(\theta_1 + \theta_2) \\ m_2 g a_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \tag{5.26}$$

$$M(\theta) = \begin{bmatrix} h & i \\ j & k \end{bmatrix} \tag{5.27}$$

with

$$h = (m_1 + m_2) a_1^2 + m_2 a_2^2 + 2 m_2 a_1 a_2 \cos\theta_2 \tag{5.28}$$
$$i = m_2 a_2^2 + m_2 a_1 a_2 \cos\theta_2 \tag{5.29}$$
$$j = m_2 a_2^2 + m_2 a_1 a_2 \cos\theta_2 \tag{5.30}$$
$$k = m_2 a_2^2 \tag{5.31}$$

The terms a_1 and a_2 are the lengths of links 1 and 2, respectively and m_1 and m_2 their masses.

5.3.3 Impedance controller

The dynamic relation between the end-effector position $x(t)$, the desired Cartesian trajectory ($x_d(t)$) and the force $f(t)$ can be described as

$$\begin{aligned} M(\ddot{x_d}(t) - \ddot{x}(t)) + B(\dot{x_d}(t) - \dot{x}(t)) + \\ K(x_d(t) - x(t)) = f(t) \end{aligned} \tag{5.32}$$

Note that if no impedance controller is present, $x_d(t)$ equals $x_r(t)$ in Eq. (5.32). Defining an impedance controller as $G(s) = M_T s^2 + D_T s + K_T$, we can use an equation of the same form as (5.32) to modify the dynamical response of the robot by modifying the reference trajectory $x_r(t)$ as:

$$\begin{aligned} M_T(\ddot{x_d}(t) - \ddot{x_r}(t)) + B_T(\dot{x_d}(t) - \dot{x_r}(t)) + \\ K_T(x_d(t) - x_r(t)) = f(t) \end{aligned} \tag{5.33}$$

where M_T, D_T and K_T are the inertia, damping and stiffness coefficients, respectively, that will define the dynamic behaviour of the robot and produce the same effect as if physical masses, springs and dampers were included in the robot. Assuming that our contact desired accelerations and velocities are zero ($\ddot{x_d}(t) = \dot{x_d}(t) = 0$) and we have ideal joint controllers that achieve $x_r(t) = x(t)$, the following equation describes the behaviour of the impedance controller:

$$M_T \ddot{x(t)} + B_T \dot{x}(t) + K_T(x(t) - x_d(t)) = -f(t) \tag{5.34}$$

An impedance controller as in Eq. (5.34) is used in our experiment to modify the dynamical properties of our robotic manipulator and make it compliant with the identified object. This controller has as inputs the robot's desired Cartesian trajectory $x_d(t)$ at each time step and the measured contact forces to get immediate feedback of the contact state. The output is a modified trajectory $x_r(t)$ which takes into account the contact forces to control the contact impedance by modifying the relation between force and position. In other words, if no forces are sensed, the robot's position trajectory is strictly followed. Otherwise, when forces are measured, the trajectory is modified in order to limit the maximum steady-state forces and to dynamically behave as the mass-spring-damper system described in the control law given by Eq. (5.32).

5.3.4 Environment model

Our simulations will make use of two different environment models, one linear and one non-linear.

Linear model Often a simple linear spring model is used as model for the environment:

$$f = K_e(x - x_e) \tag{5.35}$$

where f is the contact force, K_e is the stiffness of the environment, x is the end-effector position at the contact point and x_e is the static position of the environment.

We assume that the environmental stiffness can be modelled as a linear spring with a spring constant K_e.

Hunt-Crossley model A better and more physically-consistent environment model is a non-linear Hunt-Crossley relation ([34]) instead of the previous classical linear Kelvin-Voigt model (or spring-like model). Besides, it allows to describe the behaviour of both stiff and soft objects, and it is computationally simple to be computed on-line. The model obeys the following relation:

$$F(t) = kx^n(t) + \lambda x^n(t)\dot{x}(t), x \geq 0 \tag{5.36}$$

where n is a real number that takes into account the geometry of the contact surfaces.

5.3.5 Contact forces

The dynamic equation describing the behaviour of our system was defined in Eq. (5.23). In that case, u was considered to include the effects of all the forces acting on the robot, i.e. also the external contact forces. To make it clearer, we will modify Eq. (5.23) to show the effect of those forces and will compensate for them in our model. The dynamic equation governing the robot's behaviour might be defined as

$$M(\theta)\ddot{\theta} + B(\theta, \dot{\theta}) + G(\theta) = u - J^T(\theta)f \tag{5.37}$$

where now u defines the driving torques and the term $J^T(\theta)f$ translates the task-space forces f acting on the end-effector to the joint space making use of the traspose of the Jacobian. Our MATLAB/Simulink model will include Eq. (5.37) in order to take account for contact forces on the dynamic response. In the case of a two-link planar arm, and knowing that the relation between torques and forces is defined as $\tau_c = J^T f$, the following relation can be written:

$$\tau_{1c} = J_{11}f_x + J_{21}f_y \tag{5.38}$$
$$\tau_{2c} = J_{12}f_x + J_{22}f_y \tag{5.39}$$

where τ_c are the contact torques, J_{ij} are the elements of the traspose of the 2x2 Jacobian matrix and f_x and f_y are the forces along the X and Y directions, respectively. Since we assume that no forces are acting along the Y axis, because its path in this direction is free and we do not consider friction forces, we can assume that $f_y = 0$ so that

$$\tau_{1c} = J_{11}f_x \tag{5.40}$$
$$\tau_{2c} = J_{12}f_x \tag{5.41}$$

These torques are to be included in equation Eq. (5.37).

5.3.6 Model-based dynamic controller

The dynamical model of the robot arm is that of a complex, highly non-linear, and coupled system. In order to control such a system, a model-based dynamical controller can be used that linearises and decouples the system. Once the system is linear and non-coupled, a simple PD controller can be used to control the robotic arm. Needless to say, this method is especially interesting for simulation purposes, as we probably would already have a dynamical model of our robot. In practical cases, model-based techniques will have to deal with the uncertainty on the parameters of the model, and of the model itself. The following paragraphs will describe the technique to linearise and decouple the system. We start by defining a controller such as

$$\alpha u' + \beta \tag{5.42}$$

being u' the new control input, and define

$$\alpha = M(\theta)\ddot{\theta} \tag{5.43}$$

$$\beta = B(\theta, \dot{\theta}) + G(\theta) \tag{5.44}$$

Combining the controller with the dynamic model $M(\theta)\ddot{\theta}+B(\theta,\dot{\theta})+G(\theta) = \alpha u'+\beta$ and simplifying that leads to the system

$$\ddot{\theta} = u' \tag{5.45}$$

Our control input will have to deal with a linear and very simple model. This solution will work as long as it is possible to accurately represent and implement α and β. As we will see later with the simulations in MATLAB, the analytical solution for α and β matches precisely the dynamics described by the mechanical model in SimMechanics, where we modelled the two-link planar arm. The new control input u' might be then easily implemented as a typical PD controller:

$$u' = -K_P(\theta_d - \theta) - K_V(\dot{\theta}_d - \dot{\theta}) \tag{5.46}$$

In other words, Equations (5.43) and (5.44) describe a dynamic model-based controller for the manipulator that will be used to cancel the non-linearities of the manipulator in order to achieve a model to control as simple as a double integrator represented in Eq. (5.45). The general structure of the dynamic model-based controller can be seen in Figure 5.6.

5.4 Simulation Experiments

Figure 5.7 shows the complete MATLAB model of the control system. It includes the mechanical model of the two-link arm described using SimMechanics, the dynamic model-based controller, the inverse kinematics module, the model of the environment

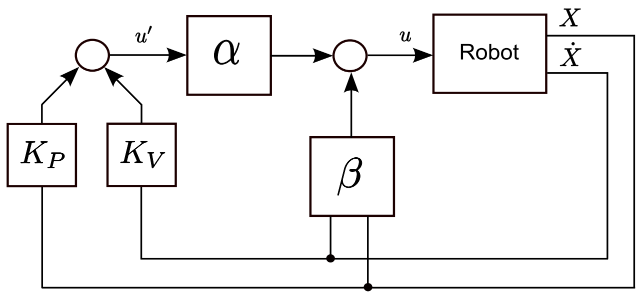


Figure 5.6: General structure of the dynamic model-based controller

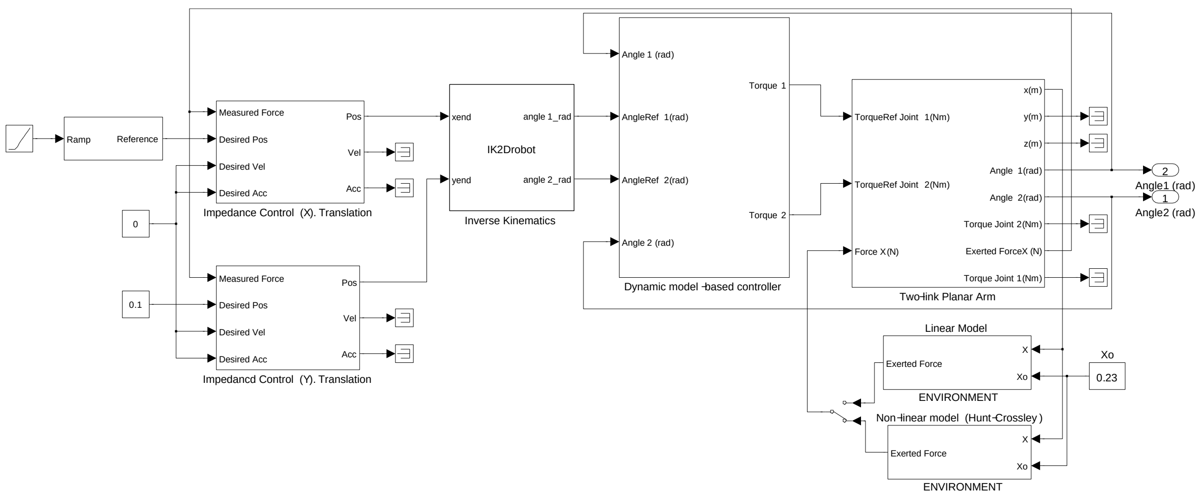


Figure 5.7: Complete control system modelled with MATLAB/Simulink

(linear and Hunt-Crossley), the impedance controller, and the desired reference trajectories.

In order to test the performance of the control system, the robot is commanded to track a Cartesian position trajectory defined as:

$$x(t) = 0.06t + 0.2 \tag{5.47}$$

The reference for the Y-axis is a constant height at $Y = 0.1$. At $X = 0.23$ a wall modelled with two different models is installed:

(1) a linear model as defined in Eq. (5.35) with $K_e = 25000N/m$, and

(2) a non-linear Hunt-Crossley model as defined in Eq. (6.16) with $k = 250N/m$, $n = 0.5$, and $\lambda = 0.0072$.

The values used for the PD controller of the dynamical model-based controller were $K_P = 10000$ and $K_V = 100$. The robot dynamical model uses link lengths of $a_1 = a_2 = 0.2m$, and link masses of $m_1 = m_2 = 10kg$. The impedance controller parameters are $M_T = 30kg$, $D_T = 100Ns/m$ and $K_T = 5000N/m$. The steady-state contact force can be selected by choosing an appropriate value for the stiffness value K_T of the impedance controller. Using Eq (5.20), the value for K_T was computed aiming at a steady-state contact force of approximately 20N. The value for the penetration p_{dss} is computed as the difference between the desired end position (in this case, $X = 0.235$), and the position of the environment (in this case, $X = 0.23$). For the experiments, two instances of the two-link robot arm models were used simultaneously, one equipped with an impedance controller and the second one without it. Figure 5.8 shows a mechanical diagram depicting both robot arms at a resting state after contacting the wall placed at $X = 0.23$.

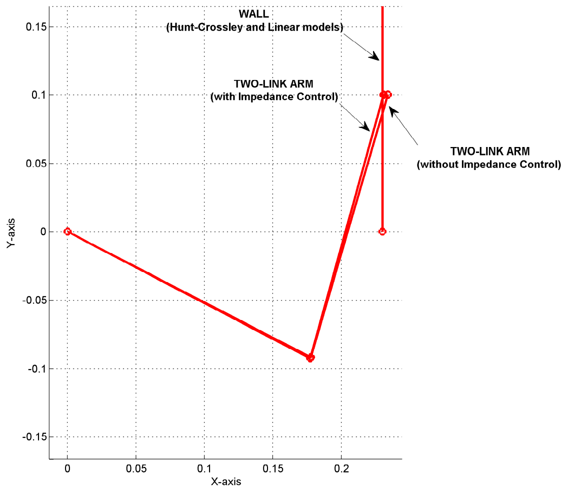


Figure 5.8: Mechanical diagram of the two two-link robot arms (with and without impedance controller) after contacting the environment

Figure 5.9 shows the response of both robot arms while approaching and contacting the linear model for the environment with and without including the impedance controller in the feedback loop. At contact point ($X = 0.23$), the contact force f increases and the impedance controller reacts to redefine a new position trajectory that limits the steady-state forces to a value of around 20N by limiting accordingly the value for the X reference position. Notice that the position tracking performance of the internal position controller (visible before the contact occurs) has not been degraded by including the impedance controller. The same figure includes the contact forces arising on the robot arm that did not include the impedance controller, which are noticeably higher. The vertical dotted line indicates the position of the environment (that is to say, the instant where the contact occurs), whereas the vertical dashed line indicates the instant where the motion trajectory finishes (the value for the desired position remains at a constant value). Notice that the steady-state values computed in Eq. (5.19), Eq. (5.20), and Eq. (5.21) apply only for computation of the steady-state error to a constant input.

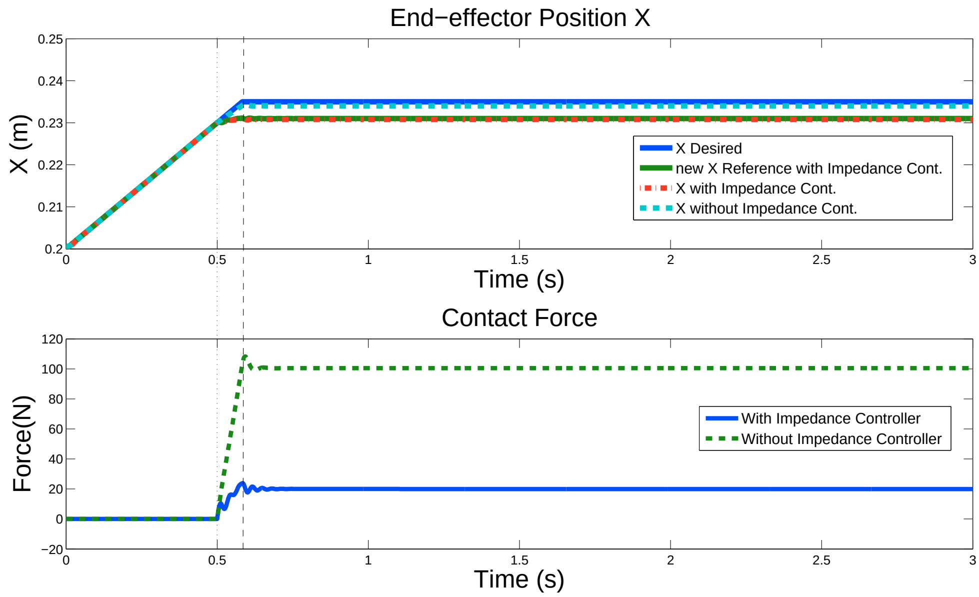


Figure 5.9: Contact response using the impedance controller with the linear environment model *top:* desired, reference and current X Position (with and without impedance controller) *bottom*: contact force (with and without impedance controller)

Figure 5.10 shows the steady-state contact force for different values of the stiffness K_T of the impedance controller. Using Eq. (5.20), the required values for K_T were computed to achieve contact forces $f = 5, 10, 15, 20N$, yielding as a result the values $K_T = 1000, 2200, 3500, 5000$, respectively.

Figure 5.11 repeats the previous contact experiment using a non-linear Hunt-Crossley environment model. Similar results are obtained where at the contact point $(X = 0.23)$, the contact force f increases and how the impedance controller reacts to redefine a new position trajectory that limits the steady-state forces to a value of around 12N. Notice that the steady-state force value previously compute applies only to linear environment models. Nonetheless, the contact remains stable and the steady-state contact force can be easily set to settle at a desired value.

Stability As we saw in Section 5.2.3, Eq. (5.22) can be used to check the stability of the coupled system environment-impedance controller. Using the values of the first impedance controller, that is, $M_T = 30kg$, $D_T = 100Ns/m$ and $K_T = 5000N/m$, and a linear environment model with $K_e = 25000N/m$, we can check for the location of the poles of the system using the system depicted in Fig. 5.4. Figure 5.12(a) shows the root locus of the feedback system, where we can observe that the system will remain always stable for any gain. The poles of the system are $r_{1,2} = -1.667 \mp 12.8j$, thus located at the half-left plane, a necessary condition for stability. Figure 5.12(b) shows the frequency response of the couple system (Bode plot) showing the gain and phase margins. Notice that since the gain margin is infinite due to the fact that the system

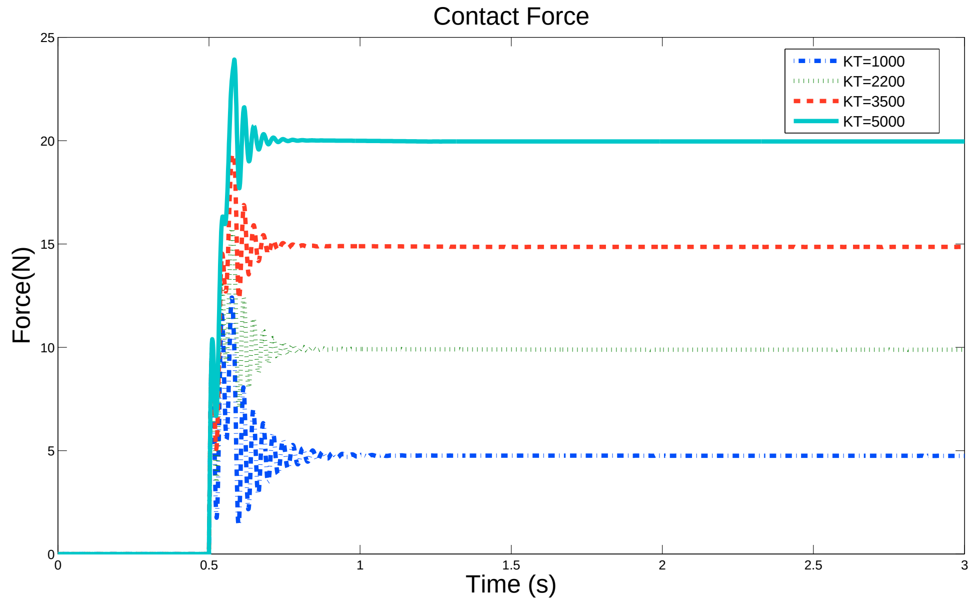


Figure 5.10: Contact response for different values of K_T in order to achieve a specific steady-state contact force

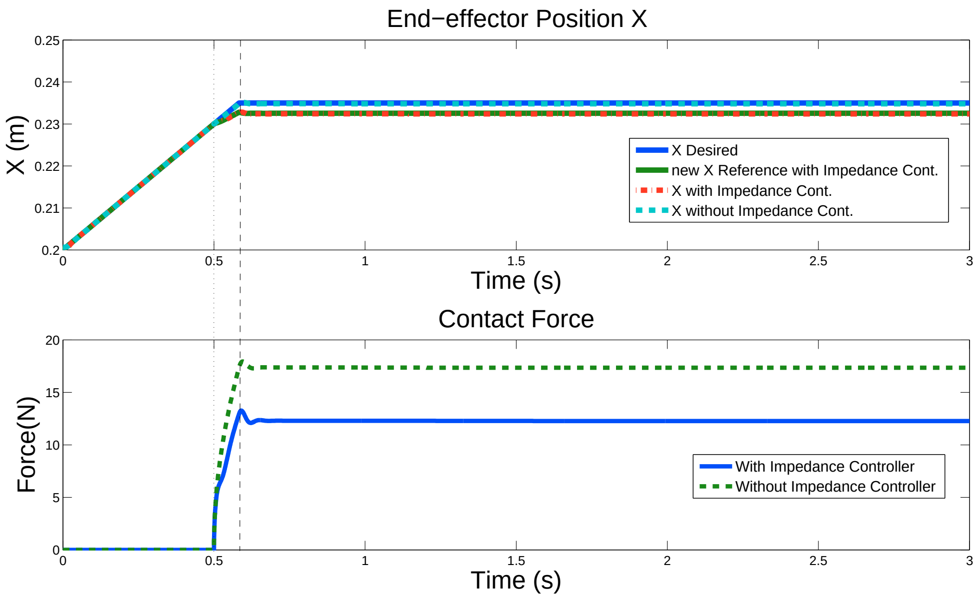
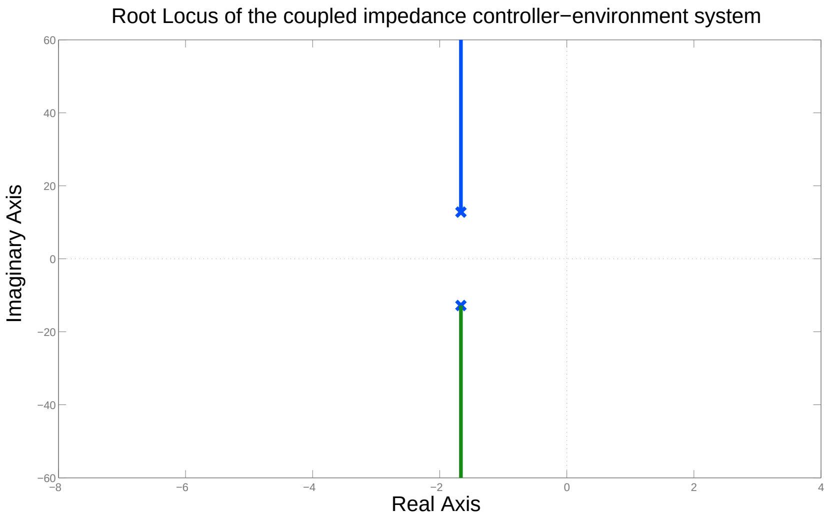
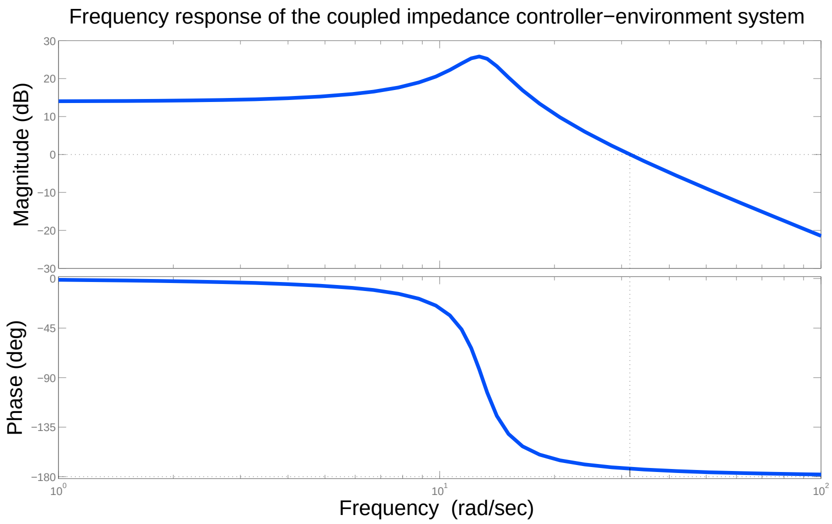


Figure 5.11: Contact response using the impedance controller with the non-linear Hunt-Crossley environment model *top:* desired, reference and current X Position (with and without impedance controller) *bottom*: contact force (with and without impedance controller)

has no gain for which the phase is -180 , the system will be always stable.



(a)



(b)

Figure 5.12: Stability of the couple system impedance control and environment *top:* root locus of the coupled system *bottom*: gain and phase margins within the Bode plot of the coupled system

5.5 Environment Estimation

Before we proceed with the experimentation with a real robot to test the performance of the impedance controller, the following sections will describe the development of an on-line Bayesian-based estimation algorithm that is able to robustly identify the most likely contacted environment from within a set of previously known environments. This information will be later used to adapt the impedance controller to the specifics of the encountered environment.

5.5.1 Off-line data analysis

Bayesian inference is a statistical method which tries to derive conclusions in a similar way to the scientific method: by collecting evidence about the trueness or falseness of a hypothesis. The probabilities of an event happening are continuously updated as new evidence is available. Basically it is composed of a prior probability that represents the knowledge about the occurrence of an event before any evidence has been gathered and a likelihood function representing the probability that a particular evidence is observed given that a particular hypothesis is true. Bayesian inference gives as an outcome a new final probability after an evidence is observed, that updates the current probability.

Using this statistical method, we analyse a priori our data and update our predictions after new sensory data is available. In our real experimental setup, the sensory data vector x is formed by the readings of the position and force along the Cartesian axis Y at a fixed sampling time t while contacting the environment. Assuming n time steps, the sensory data vector is in the form $x = [x_1, x_2]$ where $x_1 = [x_{pos_Y}(0), ..., x_{pos_Y}(n)]$ represents the Cartesian position along the Y axis and $x_2 = [x_{f_Y}(0), ..., x_{f_Y}(n)]$, stands for the Cartesian force along that axis. Given a set of environments $E = \{E_1, E_2, ..., E_N\}$, the probability of contacting one of them in the current state is considered as

$$p(E_j) = \frac{1}{N}, E_j \quad \in E \tag{5.48}$$

That is, all environments have the same a priori probability to be contacted. The likelihood function $p(x|E_j)$, the probability of sensory data vector x occurring given that the hypothesis of E_j is true, is approximated with the bivariate normal density function:

$$p(x_1, x_2) = \frac{1}{2\pi\sqrt{\sigma_1^2\sigma_2^2(1-\rho^2)}} e^{\frac{-z}{2(1-\rho^2)}} \tag{5.49}$$

where

$$z \equiv \frac{\|x_1 - \mu_1\|^2}{\sigma_1^2} - \frac{2\rho\,\|x_1 - \mu_1\|\,\|x_2 - \mu_2\|}{\sigma_1\sigma_2} + \frac{\|x_2 - \mu_2\|^2}{\sigma_2^2} \tag{5.50}$$

being μ_1, μ_2 the mean and σ_1^2, σ_2^2 the variance of vector x, extracted from the training data and the term ρ represents the correlation between x_1 and x_2.

5.5.2 On-line identification of the environment

Just immediately after a contact, the robot computes the posterior probability of an environment $E_j, E_j \in E$ given a sensory data vector x as:

$$p(E_j|x) = \frac{p(x|E_j)p(E_j)}{p(x)} \tag{5.51}$$

As stated in section 5.5.1, the probability $p(E_j)$ is considered as

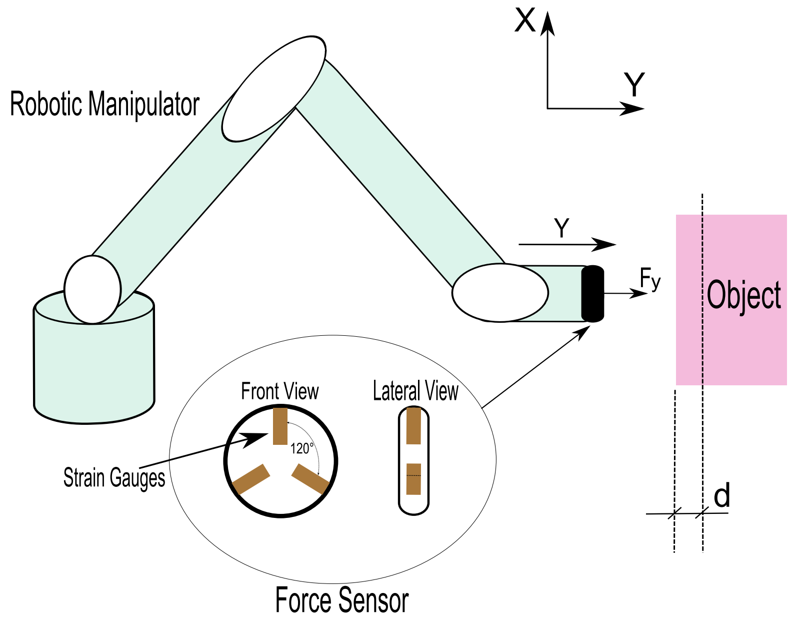


Figure 5.13: Experimental set-up for the Mitsubishi PA-10 robot

$$p(E_j) = \frac{1}{N}, E_j \quad \in E \tag{5.52}$$

The marginal probability $p(x)$ can be calculated as the sum of the product of all probabilities of mutually exclusive hypotheses and corresponding conditional probabilities:

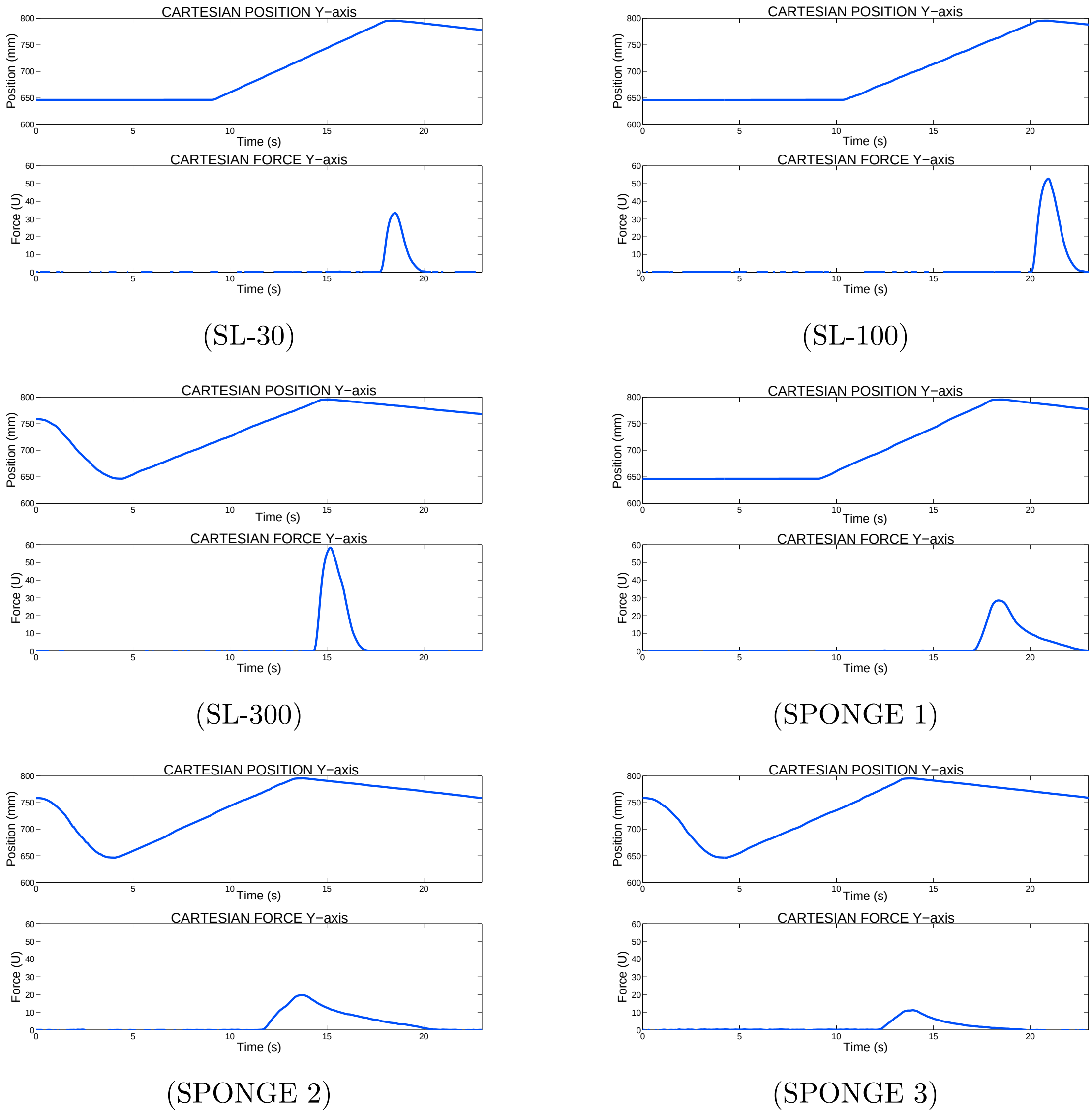
$$p(x) = \sum_j p(x|E_j)p(E_j) \tag{5.53}$$

In order to relate a given sensory data vector x to an environment E_j, the posterior probabilities for all possible environments are computed and the environment with the maximum probability value is selected as most probable environment.

5.5.3 Multi-instance impedance controller

Recent neuroscience results postulate about the use of internal models by the Central Nervous System (CNS), in other words, that the CNS runs internal models that relate sensory information to a given motor command (forward models [129]) and motor commands to a given sensory information (inverse models [136]). Recently some theoretical computational models haven been also presented that incorporate such internal models in the lowest layer of a multilayer control system [135].

Inspired by the notion of multiple instances of controllers which are selected given the current most likely estimated context, this section aims at generating a set of



(SL-30) (SL-100)

(SL-300) (SPONGE 1)

(SPONGE 2) (SPONGE 3)

Figure 5.14: One instance of the force data responses for each of the six objects

impedance controllers that are ‘tuned’ for the specifics of each of the environments with which the robot will be trained. Any new environment found and trained will be accompanied by the corresponding controller. Chapter 7 will later use some of these results and expand the concept to include a Bayesian estimator that predicts the context on which the robot is currently at, in order to accordingly switch to the most favourable impedance controller.

In our current setup, once sensory data information for each of the environments is available, this data is used to create a model for each environment E_j of the set of environments E. This function will model the relation between force and position for each environment (relation shown in Fig. 5.15). The model structure will be given to a optimisation algorithm so that it searches for the optimal parameters of the model structure where the sensory data vector fits best. In this case, the environment is modelled with a 5th-order polynomial that fits the data in the vector x. To solve our optimisation problem, the CMA-ES [45] algorithm is used. CMA-ES is an efficient evolutionary strategy which can provide excellent results in optimization problems.

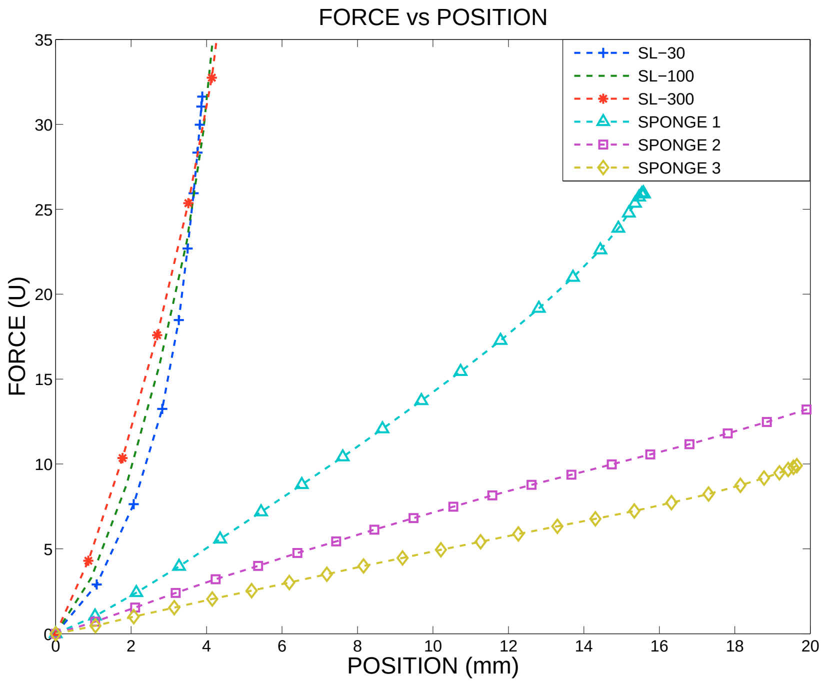


Figure 5.15: Data sensory vector x (force-position response) for each of the objects used in the experiment

That is in summary, for each of the training data sets x, a CMA-ES model is generated which estimates the force-position relation of each environment. In order to implement the impedance controller in a computer, first it needs to be discretised. The mechanical impedance of a manipulator, that is, the relation between the robot's end-effector position and force, can be expressed in the Laplace domain as:

$$H(s) = \frac{E(s)}{F(s)} = \frac{1}{M_T s^2 + D_T s + K_T} \tag{5.54}$$

where $E(s)$ and $F(s)$ stand for the relative Cartesian error position and Cartesian force of the robot's end-effector, respectively, and M_T, D_T and K_T represent the target inertia, damping, and stiffness of the contact. A discrete version of the continuous second-order system described by Eq. (5.54) is obtained applying a Tustin bilinear transformation as:

$$H(z) = \frac{E(z)}{F(z)} = \frac{a_1 + a_2 z^{-1} + a_3 z^{-2}}{1 - b_1 z^{-1} + b_2 z^{-2}} \tag{5.55}$$

where, for our experiment, $E(z)$ stands for the relative Cartesian error position of the robot's end-effector and $F(z)$ for the Cartesian force measured along the Y-axis.

5.6 Experimental results

The previous simulated experiments showed good performance with well-known environments. This is however not a realistic scenario and the following experiments will assess the performance of the proposed schemes in a real experimental setup. Prior to that, the performance tests of the Bayesian-based estimation algorithm will be presented.

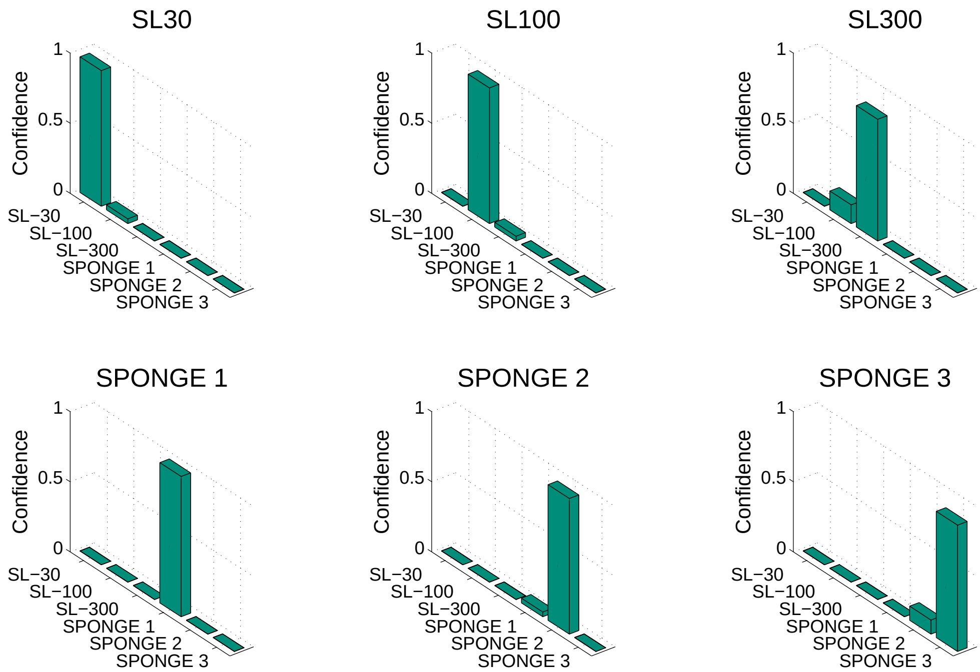


Figure 5.16: Normalised confidence of the algorithm for each environment after being exposed 30 times to each of them

5.6.1 Setup

In order to identify the environment, the experiment makes use of a 7 degree-of-freedom Mitsubishi PA-10 industrial robotic arm, whose end-effector has been equipped with a custom-built three-dimensional force sensor. Note that the force sensor has not been calibrated to provide measurements in Newtons thus they are provided as unitless. Due to the high repeatability of the measurements and the nature of the identification process, qualitative rather than quantitative results are necessary. Later experiments with a commercial 6-axis force-torque sensor showed increased performance of the identification process thus validating the methodology. The experiment proceeds as follows (a diagram can be seen on Figure 5.13):

(1) The robot is commanded to follow a straight trajectory at constant velocity along the Cartesian Y axis.

(2) An object is placed along the robot's trajectory in order that the robot collides with it.

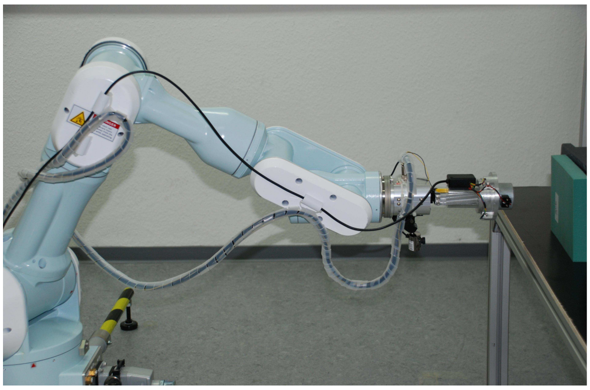
(3) The penetration distance once the object has been contacted (d in Fig. 5.13) is a fixed parameter.

(4) Force and position along the Y-axis are recorded for posterior data analysis.

The experimental environment is made up of a set of six objects: 3 structured and well-characterised damping plates and 3 non-structured common sponges. The damping plates (manufactured by ACE) are from the series SL-30, SL-100 and SL-300 of viscoelastic PUR (polyether urethane) material, especially designed to absorb shock loads. This material is characterized by its very high inner damping. These plates were also chosen because they offer very similar responses to an impact, thus being suitable to test the ability of our algorithm to distinguish subtle differences occurring in the environment but using a well-characterised and structured material. The non-structured objects are three common sponges of different softness, which are selected because of their non-linear behaviour as opposed to the damping plates.

Initially a series of experiments were performed in order to collect data for computing the conditional probability of seeing the sensory data vector x given that the hypothesis about the environment E_j is true, which is also known as likelihood probability (Eq. (5.49)). In order to extract μ_1, μ_2 and σ_1^2, σ_2^2 for our sensory vector x (see Eq. (5.49)) from the training data, the contact experiments are repeated 23 times and averages for mean and variance of each distribution are generated. Figure 5.14 shows one instance of the force responses obtained for each of the objects presented to the robot. Figure 5.15 summarises the averaged force-position responses of these three damping plates and the three sponges, when being contacted with the robot. During the on-line estimation phase, the data will be collected only in the first 4mm after contacting the object and will be used to obtain the final (posterior) probability (Eq. (5.51)) of an environment, given a sensory data vector x.

To test our identification module, the robot was exposed 30 times to each environment and the posterior probabilities for each of the possible environments were computed on-line. Figure 5.16 shows the result of this evaluation and how the robot was able to recognise the correct environment with a very high confidence. Each subfigure represents the environment under test and for each of the environments (abscissa axis) a confidence index is depicted. This normalised confidence index is computed by normalising the probabilities to the maximum obtained for a given vector x under test of the environment E_j for each $E_j \in E$. Note for example that the object SL-300 is identified correctly in most of the cases but it is the object with more probabilities to be misclassified due to its similarities to the object SL-100 (see Fig. 5.15). SL-100 achieves a confidence index of around 0.17 when evaluating object SL-300. Whether this ratio is acceptable or not might be subject to discussion. Given the fact that the objects SL-100 and SL-300 are very similar (see Fig. 5.15) and that the goal of the



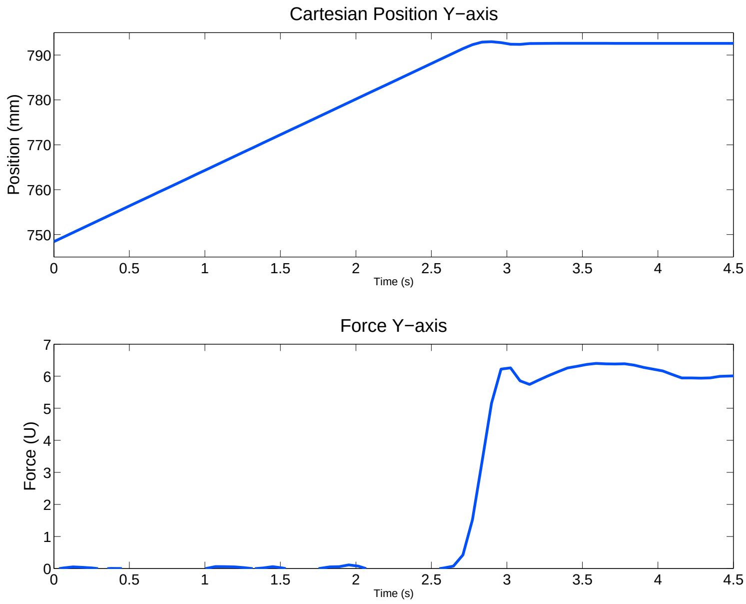


Figure 5.17: Environment estimation and adaptation of interaction forces using the proposed approach on the Mitsubishi PA-10 robot

estimation is to adapt the contact interaction to a given environment, a 'misclassification' in this scenario means that the robot will adapt to a false environment that from a practical point of view requires a very similar response. In other words, the responses required to deal with objects SL-100 and SL-300 are so similar that a misclassification will not affect the final result. More important is a complete adaptation when switching contacts between, for instance, one of the damping plates and one of the sponges. Moreover, it is worth mentioning that the results are rated as splendid given the fact that our custom-built sensor cannot provide the resolution and repeatability of commercial force sensors. It is in fact the great performance of the Bayesian estimation module that provides us with outstanding results thus proving that the method is effective when dealing with high uncertainty on the data.

An example of the compliance adaption using the identified environment is shown in Figure 5.17 using object SL-30. First, using a simulation of the PA-10 robot in contact with real data of the SL-30 environment, a set of parameters for the discrete impedance controller in Eq. (5.55) was selected to be $a_1 = 0.2766$, $a_2 = -0.0869$, $a_3 = -0.0507$, $b_1 = -0.4060$ and $b_2 = 0.2917$. Figure 5.17 shows the robot contacting the environment SL-30; in case of not having an interaction controller, the robot would continue penetrating into the object, applying as much force as necessary to obey the command dictated by the robot's position controller to continue straight. However, due to the action of the impedance controller, the current robot's position trajectory is modified and the robot remains at a fixed position 'inside' the object, determined by the maximum steady-state force that the impedance controller dictates.

5.7 Conclusions

This chapter presented a method to control the interaction forces when contacting an environment with a robotic manipulator. By estimating the most probable enviroment by means of a Bayesian inference model, the mechanical impedance of the robot and thus the forces exerted into the environment are controlled in order to accommodate for the current context. As humans do when identifying an unknown environment, a first contact interaction is used to confirm a hypothesis about its nature. Subsequent interactions are then 'tuned' to that specific scenario thus accounting for the proper contact forces to be exerted. The results enable a robotic manipulator to adapt its behaviour to multiple and heterogeneous environments far beyond the well-characterised and static workspace of an industrial robotic manipulator.

Chapter 6

EVOLUTIONARY TECHNIQUES FOR INTERACTION CONTROL

This chapter describes the use of evolutionary algorithms to find an optimal solution for the parameters of an impedance controller represented as an artificial neural network (ANN). Moreover, the impedance controller is enhanced with force tracking capabilities using evolutionary strategies in order to control the forces between a robotic manipulator and the environment.

The chapter concludes showing the results obtained with the evolved ANN-based impedance controller using a simulated two-link planar robotic arm.

6.1 Introduction

This chapter aims at describing the use of evolutionary techniques to control the interaction forces between a robot manipulator and the environment. More specifically, the chapter focuses on the design of an optimal and robust force-tracking impedance controller, the highlighted component in the architecture proposed in this thesis and shown in Fig. 6.1. As we have seen in previous chapters, current state-of-the-art approaches start usually the analysis and design of the properties of the impedance controller from a manually-given set of impedance parameters, since no well-defined methodology has been yet presented to obtain them. Neuroevolutionary methods are showing promising results as methods to solve learning tasks, especially those which are stochastic, partially observable, and noisy. Evolution strategies can be also used to perform efficient optimization, as it is the case in CMA-ES (Covariance Matrix Adaptation - Evolution Strategy) [110].

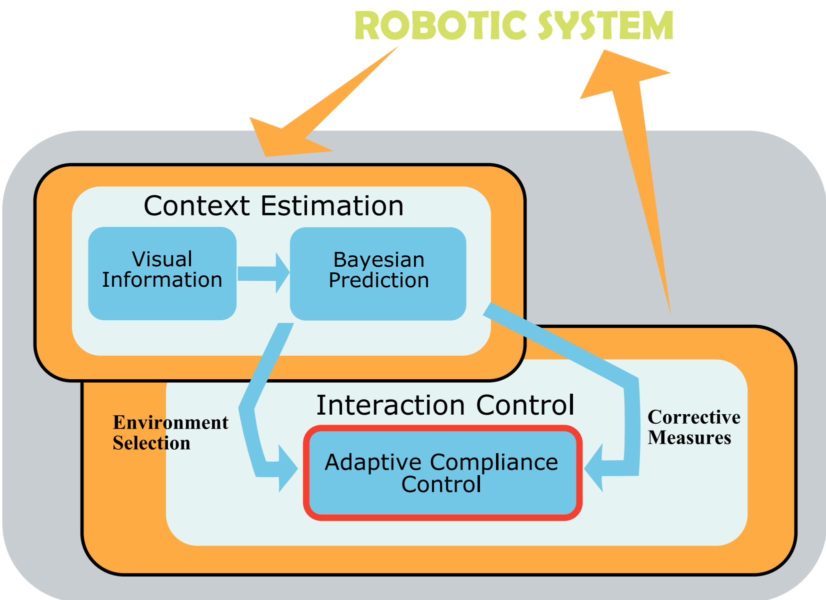


Figure 6.1: Proposed architecture for compliance control via context estimation. The component highlighted is discussed in this chapter

Neuroevolution is the combination of neural networks as structure for the controller and an evolutionary strategy which in the simplest case searches for the optimal weights of this neural network. The weights of this neural network represent the policy of the agent, in control engineering terms known as the control law. Consequently, the weights of this neural network bound the space of policies that the network can follow. In more complex strategies, the evolutionary strategy evolves both the weights and the topology of the neural network. In optimal control, one tries to find a controller that provides the best performance with respect to some measure. This measure can be for example the least amount of control signal energy that is necessary to bring the controller's output to zero. Whether in classical optimal control or in neuroevolutionary methods, there is an optimization process involved and we show in this chapter that neuroevolutionary methods can provide a good alternative to easily design optimal controllers. In this case study, an impedance controller represented as an artificial neural network (ANN) will be described, whose optimal parameters are obtained in a simple way by means of evolutionary techniques. The controller will regulate the contact forces between a robotic manipulator (a two-link planar arm) and the environment. Furthermore, it will be generalised and provided with force tracking capabilities through an on-line parameter estimator that will dynamically compute the weights of the ANN-based impedance controller based on the current force reference. The resulting controller presents robustness against uncertainties both on the robot and/or the environmental model. The performance of the controller has been evaluated on a range of experiments using a model of a two-link robotic arm and a non-linear model of the environment. The results evidenced a great performance on force-tracking tasks as well as particular robustness against parametric uncertainties. Finally, the controller was enhanced with a steady-state Kalman filter whose parameters were learned simultaneously with the weights of the ANN. That provided robustness against the measurement noise, especially important in the force measurements.

6.2 System Description

The system's control architecture (Fig. 6.2) used for the experiments and implemented under MATLAB is composed of the following submodules: Trajectory Generation module, Impedance Controller (neural network-based controller), Direct and Inverse Kinematics modules, Dynamical Model-Based Controller module, Two-link Arm Dynamical Model, and Environment model.

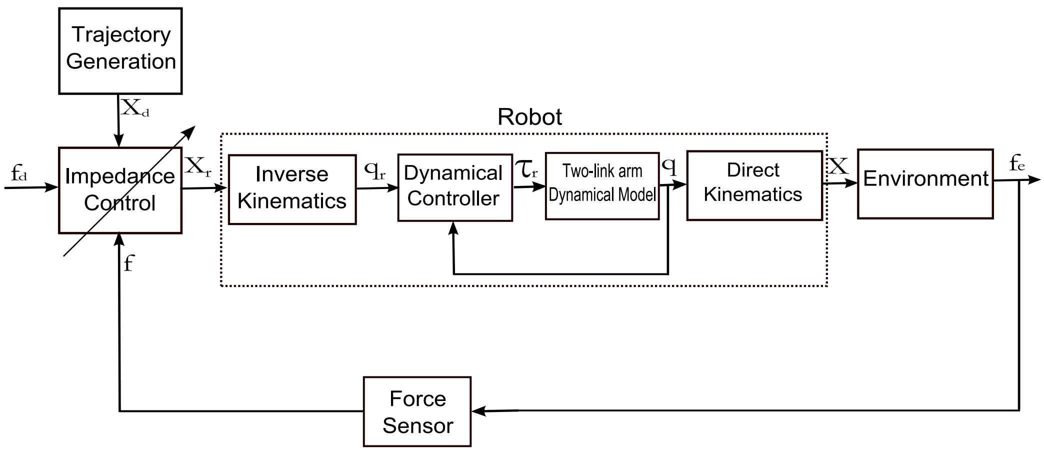


Figure 6.2: System's control architecture

6.2.1 Impedance controller

The classical impedance controller shown in Fig. 5.3 is described by the continuous-time Eq. (6.1). The mechanical impedance of a manipulator, in other words, the relation between the robot's end-effector position and force, can be expressed in the Laplace domain as:

$$H(s) = \frac{E(s)}{F(s)} = \frac{1}{M_T s^2 + D_T s + K_T} \tag{6.1}$$

where $E(s)$ and $F(s)$ stand for the relative Cartesian error position and Cartesian force of the robot's end-effector, respectively, and M_T, D_T and K_T represent the target inertia, damping, and stiffness of the contact. It is easy to draw similarities with the well-known canonical form of a second-order system:

$$H(s) = \frac{1}{s^2 + 2\xi\omega_n s + \omega_n} \tag{6.2}$$

where ξ is the damping factor and ω_n the system frequency. Equating Eq. (6.1) with Eq. (6.2), the following relation arises:

$$s^2 + \frac{D_T}{M_T}s + \frac{K_T}{M_T} = s^2 + 2\xi\omega_n s + \omega_n \tag{6.3}$$

from where we obtain the following relationships:

$$\omega_n = \sqrt{\frac{K_T}{M_T}} \tag{6.4}$$

$$\xi = \frac{D_T}{2\sqrt{K_T M_T}} \tag{6.5}$$

Using these relationships, we might be able to select a specific damping factor ξ and system frequency ω_n. Notice that the system frequency ω_n should be much lower than the frequency of the internal position controller in order to achieve a stable inner/outer control loop. Typical values for the damping factor ξ are well-known, being values below 1.0 the ones which would achieve a response without overshooting. Despite these equations, the process to obtain a stable and optimal impedance controller is not an easy endeavour. The purpose of this chapter is to ease the selection of those parameters by using neuroevolutionary methods in order to meet simultaneously different criteria on robustness, stability, and optimality. First step is to start from Eq. (6.1) in order to discretize the impedance controller for its implementation in a computer. Using the bilinear transformation, $H(z) = H(s)\mid_{s=\frac{2}{T}\frac{z-1}{z+1}}$, the discrete version of the impedance controller is obtained.

$$H(z) = \frac{E(z)}{F(z)} = \frac{T^2(z+1)^2}{w_1 z^2 + w_2 z + w_3} \tag{6.6}$$

where

$$w1 = 4M_T + 2D_T T + K_T T^2 \tag{6.7}$$
$$w2 = 2K_T T^2 - 8M_T \tag{6.8}$$
$$w3 = K_T T^2 + 4M_T - 2D_T T \tag{6.9}$$

From the discretized controller we can generate the difference equation which the filter will be implemented with in the computer:

$$\begin{aligned} E(n) = \frac{1}{w_1}(F(n)T^2 + 2T^2F(n-1) + T^2F(n-2)- \\ w_2E(n-1) - w_3E(n-2)) \end{aligned} \tag{6.10}$$

Following (6.10), it can be clearly seen that the impedance controller can be represented as a neural network as in Figure 6.3. That means that each classical impedance controller can be implemented as a one-neuron neural network with 5 inputs, 1 output, and only 3 weights.

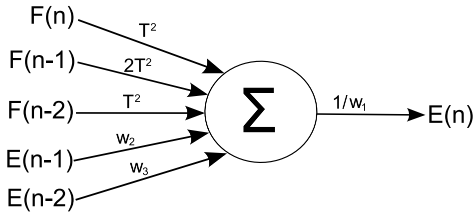


Figure 6.3: Neural network representation of the impedance controller

6.3 Evolving the ANN-based impedance controller

This section describes the process followed to evolve an ANN-based impedance controller. A first step is to create single-force impedance controllers, that is, force-tracking impedance controllers that track a specific force. Subsequently, these controllers are used as a basis for creating a generalised force-tracking controller which is able to track different force references, including those for which it was not trained for.

6.3.1 Single-force reference controller

The weights of the neural network in Fig. 6.3 are obtained by using the CMA-ES evolutionary technique. In order to do so, the closed-loop system shown in Fig. 6.4 is used. The ANN-based impedance controller modifies the desired Cartesian position trajectory for the robot (x_d) and creates a new reference trajectory (x_r) based on current sensed forces. The block named *Robot* includes the blocks enclosed under the dotted-line rectangular box in Fig. 6.2: a dynamical model-based controller that translates the Cartesian positions into the necessary torques for the robot, and forward/inverse kinematics formulations to translate from/to a Joint reference frame to/from a Cartesian frame. The contact forces exerted by the environment onto the robot (f) are fed back to the controller in order to regulate the robot-environment interaction.

The evolutionary algorithm searches for the optimal parameters M_T, D_T, and K_T, and the weights of the neural network are then computed using Eqs. (6.7), (6.8),

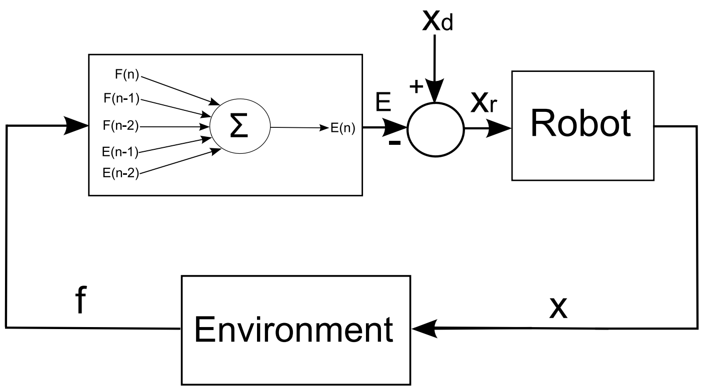


Figure 6.4: Close-loop system used to evolve the parameters of the ANN-based controller

and (6.9). A fitness function needs to be defined that drives the search and in this case was defined as to minimise the following force error criterion:

$$h = \frac{\sum_{k=1}^{N} |f_{ref} - f_k|}{N} \tag{6.11}$$

where f_{ref} is the force reference to be tracked, f_k is the actual force at time step k, and N is the number of samples. A first set of controllers were evolved using only this criterion. By doing that, a controller with fast response is obtained. On the other hand, there are situations where stability on the contact is of outmost priority. To include this additional measure on the evolution of the controller, the following criteria was used for a second series of controllers. The contact stability criterion described in [120] is applied on each individual in order to be selected as final solution. This criterion ensures that the contact with the environment is stable and no oscillations occur at the contact. A significantly overdamped impedance behaviour is required to ensure a stable contact with a stiff environment. If a relative damping coefficient is defined such as

$$\xi_T = \frac{D_T}{2\sqrt{M_T K_T}} \tag{6.12}$$

and the stiffness ratio is defined as

$$\kappa = \frac{K_E}{K_T} \tag{6.13}$$

where K_E is the stiffness of the environment, then to ensure contact stability we have to satisfy the following criterion:

$$\xi_T > 0.5(\sqrt{1 + 2\kappa} - 1) \tag{6.14}$$

CMA-ES was initialised to start the search at $[0.5, 0.5, 0.5]$, initial vector for M_T, D_T, and K_T, respectively. The initial global-step size for CMA-ES was set to $\sigma_{(0)} = 0.5$ and the system was evaluated 1000 times. The population size was chosen according to $\lambda = 4 + \lfloor 3\ln(n) \rfloor$, where n is the number of parameters to optimize and the parent number was chosen to be $\mu = \lfloor \lambda/4 \rfloor$. A series of single-force reference controllers were evolved under this setup. Each of these controllers obtained as a result of the evolutionary process the optimal weight values for a given force reference. Figure 6.5 shows the results of the controllers evolved without being strict on the contact stability, whereas Figure 6.6 shows the results where the controller has to obey the condition given by Eq. (6.14). Clearly, the latter offers a safer response at the price of making the system slower.

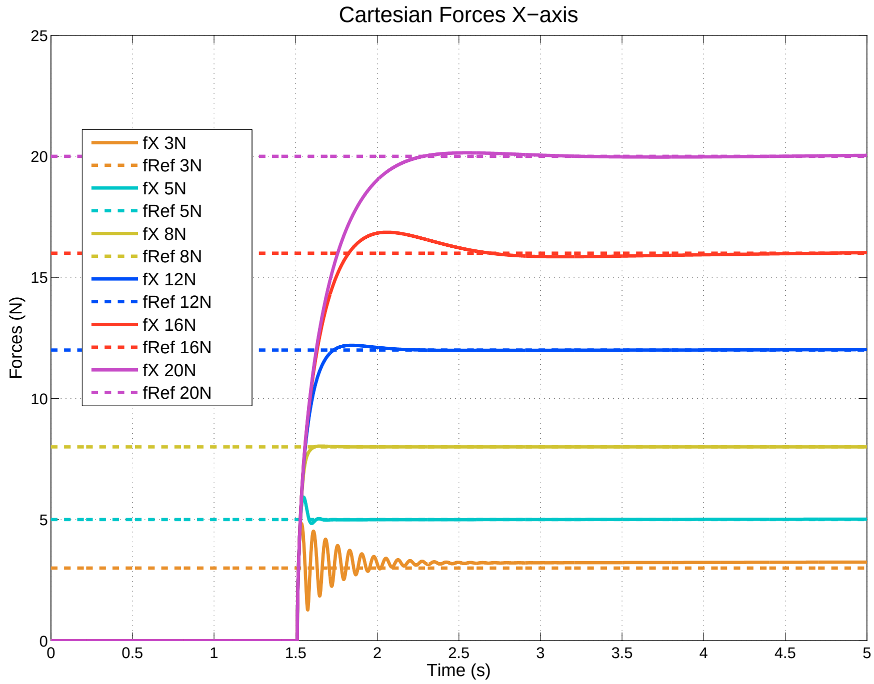


Figure 6.5: Responses of the single-force controllers evolved with CMA-ES for different force reference inputs without contact stability criterion

To summarise, each single-force controller possesses three weights and their optimal values are found for a particular reference force. In a given scenario, the evolved controller is able to control the interaction forces to the desired value and with the desired dynamical characteristics. Provided the current state-of-the-art on selecting the impedance parameters, this solution is a novelty in terms of providing a simple methodology to obtain the optimal impedance parameters for a given task.

6.3.2 Generalised force tracking controller

In this section, a more general force-tracking controller is designed that is able to adapt to different force references. To attain this goal, an additional block is added to the control scheme: the *Paramater Estimator*, a module that will generate estimations for $\widehat{M}_T$, $\widehat{D}_T$, and $\widehat{K}_T$ based on the current reference force. The complete control scheme

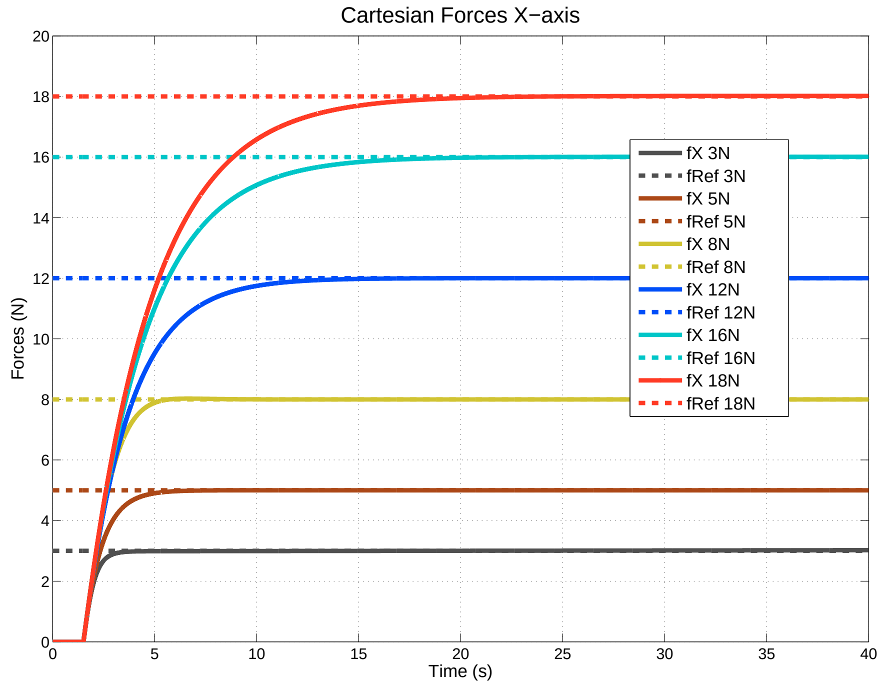


Figure 6.6: Responses of the single-force controllers evolved with CMA-ES for different force reference inputs with contact stability criterion

can be seen in Fig. 6.7. The force-to-weights data sets obtained in the previous section (Fig. 6.5) were used to generate a function that estimates the weights for the controller for any given force reference. By doing that, the input space of the force controller is generalised. Using a 6-th order polynomial as in Eq. (6.15) for each parameter (M_T, D_T, K_T), a function is generated that estimates the particular parameter for a given input force reference.

$$\widehat{y} = \sum_{i=0}^{n} a_i (f_{ref})^i \tag{6.15}$$

where $\widehat{y} = \{\widehat{y}_M, \widehat{y}_D, \widehat{y}_K\}$, are the estimation functions for each of the three parameters (M_T, D_T, K_T), respectively, and $n = 6$. The optimal coefficients a_i are again obtained using the CMA-ES evolutionary strategy.

The procedure is the following: CMA-ES is given the polynomial structure as in Eq. (6.15) and a set of force-weights training points. These points are the ones depicted in Fig. 6.8 for each of the parameters (inertia, damping, and stiffness) and relate an input force k with an output parameter. The vector k of input forces was $k = \{3, 5, 8, 12, 16, 20\}(N)$. Note that for the sake of clarity, damping and stiffness curves have been appropriately scaled in order to be shown on the same graph. The task for the CMA-ES algorithm is to find the parameters of the polynomial that best fit through the corresponding training points. The result is a function that estimates

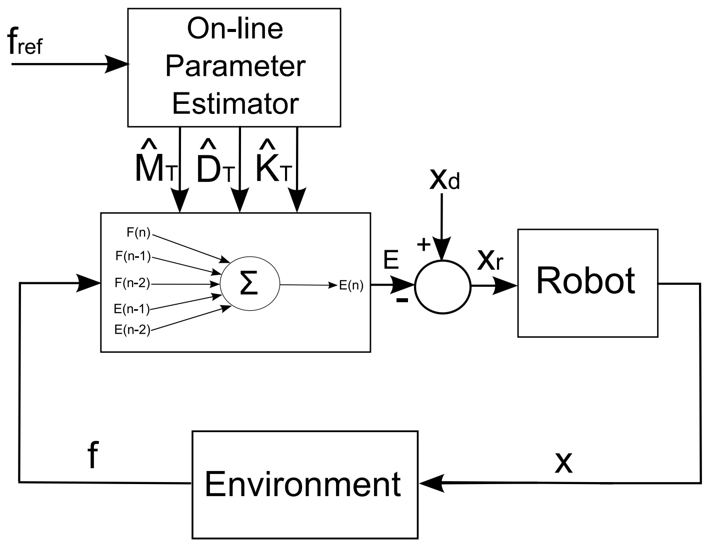


Figure 6.7: Structure of the complete control scheme

the inertia, damping, and stiffness coefficients for any given reference force. Thus the controller will adapt its weights dynamically as the force reference requirements change. As shown in Fig. 6.8, the estimated curves precisely pass through the training points (the *measured* force-to-weights relationships). CMA-ES was set to stop the search for the optimal a_i coefficients when the error between the training points and the values of the curves at force k was below $1 \cdot 10^{-10}$.

6.4 Experiments and Results

A series of experiments were conducted using a simulated two-link planar robotic arm (*Two-link Arm Dynamical Model* in Fig. 6.2) to test the performance of the ANN-based impedance controller. The robot's mechanical model is composed of two revolute joints and two bodies. The module receives torques as inputs and outputs joint angles. The masses are considered to be concentrated at the end of each link to simplify the modelling tasks. The lengths of the body links were set to $a_1 = a_2 = 0.2m$, and their masses to $m_1 = m_2 = 10kg$. A dynamic model-based controller (*Dynamical Controller* in Fig. 6.2) is used to cancel-out the non-linearities present on the dynamic model of the robot and to decouple the system. After this linearisation and decoupling process, a simple linear PD controller can be used to control the joint positions. The parameters K_p and K_v of the PD controller were set to $K_P = 10000N/m, K_V = 100Nms/rad$. The environment (*Environment* in Fig. 6.2) is modelled following a non-linear Hunt-Crossley relation ([34]) instead of the classical linear Kelvin-Voigt model (or spring-like model) since it achieves a better physical consistency and allows to describe the behaviour of both stiff and soft objects. Moreover, it is computationally simple to be computed on-line. The model obeys the following relation:

$$F(t) = kx^n(t) + \lambda x^n(t)\dot{x}(t), x \geq 0 \tag{6.16}$$

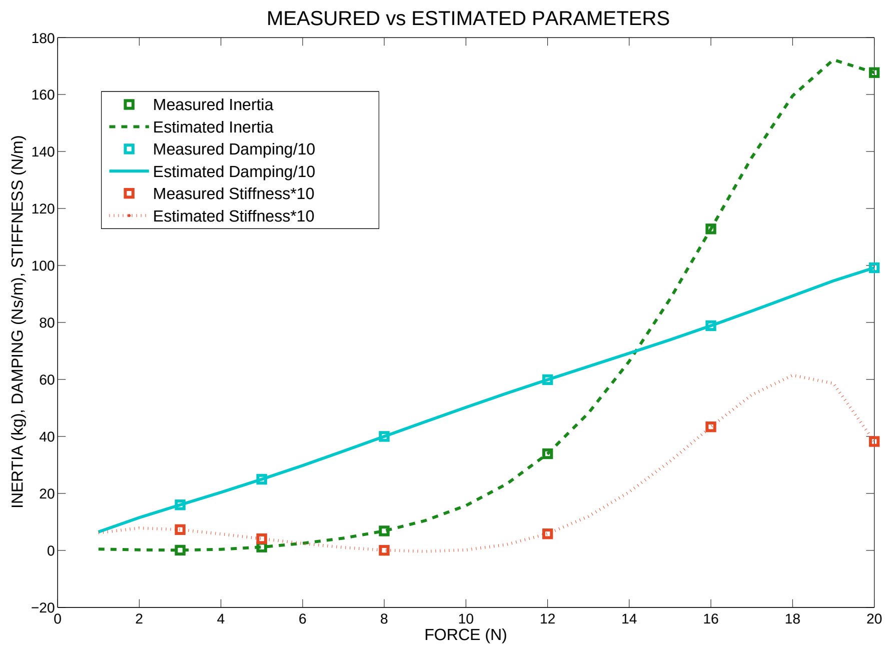


Figure 6.8: Estimation functions for each of the parameters of the impedance controller

where n is a real number that takes into account the geometry of the contact surfaces. For these experiments, the environmental parameters were set to $k = 250N/m$, $n = 0.5$, and $\lambda = 0.0072$. For all the experiments, the robot is commanded to follow a desired position trajectory in the Cartesian space: $x(t) = 0.02t + 0.2$. This desired trajectory will be eventually modified by the impedance controller to create a new reference trajectory that complies with the current force requirements.

6.4.1 Response to changes on force reference

To test the performance of the controller when dealing with force reference changes, two experiments were conducted. A first experiment presents multiple step changes on the reference force for the controller (Fig. 6.9). The upper part of the figure shows the Cartesian position on the X-axis for the tip of the robot. The robot moves along that axis until it contacts a wall, placed at $x_e = 0.23m$. The bottom part of the figure shows the robot's force responses. The reference force after contacting the environment is modified and set sequentially to $\{6.5, 10, 4, 14\}(N)$. Note that none of these values were used in the designing phase of the controller (Fig. 6.8). The robot is able to switch accurately between force references while keeping both a nearly-zero steady-state force error and a stable contact with the surface.

A second experiment was performed where the force reference is a sinusoidal signal (Fig. 6.10). In this case, a sinusoidal waveform of amplitude 2N is superimposed to the reference of 10N, i.e., the reference force to be tracked is $f_{ref} = 10 + 2sin(\pi t)$ (N). As it can be seen on the bottom part of the figure, the robot tracks the sinusoidal force

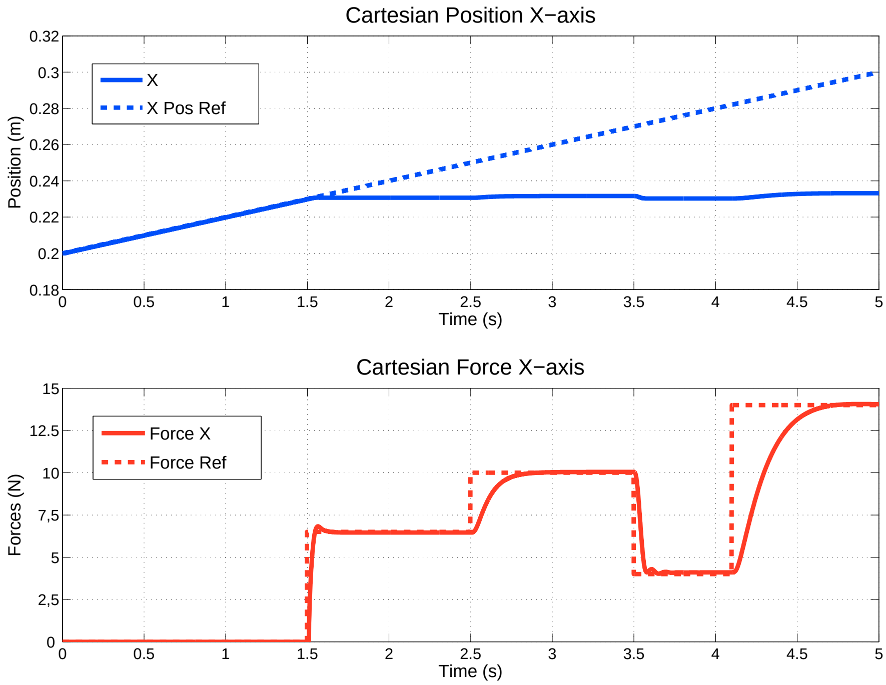


Figure 6.9: Robot's step response to changes on the reference force

reference accurately.

6.4.2 Robustness against uncertainties

The following experiments aimed at testing the robustness of the controller for changes on both the environment and the robot's model. A robust controller has to be able to cope with uncertainties, especially those related to uncertainties in the parameters of the models.

Variations on the environmental stiffness The first experiment modifies the stiffness of the environment during a stable contact. As previously stated, the stiffness of the environment in the Hunt-Crossley model was set to $k = 250N/m$. For this experiment, the stiffness is modified as $k_m = 250 \pm 10\%k$ (N/m). Figure 6.11 shows the behaviour of the controller in consequence of the changes on the environmental stiffness. The robot is able to recover and set back to the original force reference of 7N in a short time, despite of the fact that the stiffness is kept constant to a value below or over the nominal.

Variations on the robot's model A second experiment was conducted where the masses of the links of the robot were modified during a contact situation. As previously stated, the masses of the robot's links were set to $m = m_1 = m_2 = 10kg$. For this experiment, the estimated masses used on the dynamical model of the robot are

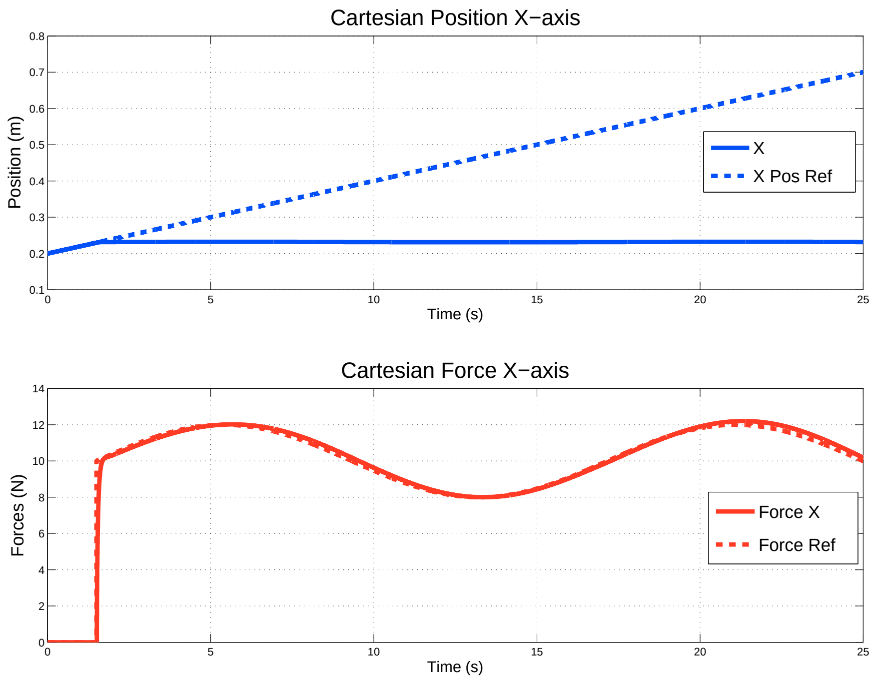


Figure 6.10: Robot's response to a sinusoidal reference force

modified as $m_m = 10 \pm 50\% m$ (kg). Figure 6.12 shows the behaviour of the controller to the changes on links' masses. The robot is again able to recover and set back to the original force reference in a short time, despite of the fact that the masses are kept constant to a value below or over the nomimal.

6.4.3 Robustness against noise

A final series of experiments aimed at testing the controller against the inherently-present measurement noise, especially important in the force measurement. The purpose of these experiments is twofold: on the one hand, to test whether the algorithm is able to find a solution using real-world noisy signals and, on the other hand, to enhance the evolved controller with a zero-delay noise filter using a Kalman filter. The filter is included on the evolution process in order to generate a one-step solution that takes into account noisy signals. In other words, the optimal parameters of the Kalman filter will be searched using the CMA-ES evolution strategy while simultaneously the controller's parameters are learned. The Kalman filter [67] estimates the state of a linear dynamical system that is perturbed by a gaussian noise. Formally, the filter addresses a general problem of estimating the true state $x \in \mathcal{R}^n$ of a discrete linear time system governed by

$$x_k = A_k x_{k-1} + B_k u_{k-1} + w_{k-1}, \tag{6.17}$$

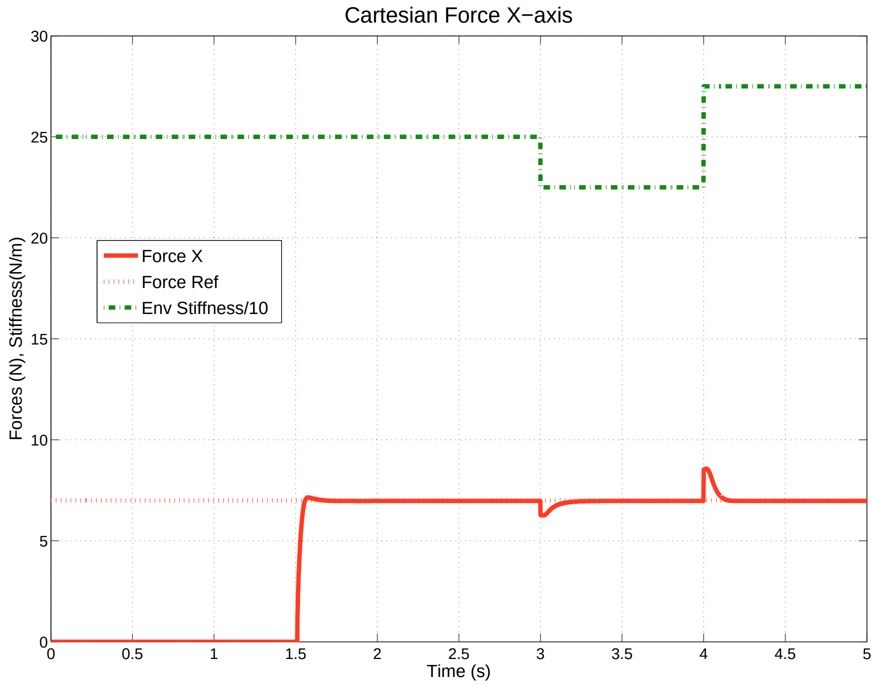


Figure 6.11: Robustness of the controller against changes on the stiffness of the environment

where A_k is an $n \times n$ state transition matrix, B_k is an $n \times m$ control input model matrix, $u_k \in \mathcal{R}^m$ is the control vector, and w_k is the process noise which is assumed to be drawn from a zero-mean multivariate normal distribution with covariance matrix Q_k of size $n \times n$. The measurment (observation) $z_k \in \mathcal{R}^l$ of the true state is modelled by

$$z_k = C_k x_k + v_k, \tag{6.18}$$

where C_k is an $l \times n$ matrix representing the measurement model and v_k is the measurment noise which is again assumed to be drawn from a zero mean multivariate normal distribution with covariance matrix R_k of size $l \times l$. The Kalman filter recursively estimates the current state based on the current measurement and the estimate from the previous state. The filter has basically two distinct phases: *predict* and *update*. Let $P_{k|k-1}$ and $\hat{x}_{k|k-1}$ be the *a priori* estimate of the error covariance matrix and the true state at timestep k, respectively, and $P_{k|k}$ and $\hat{x}_{k|k}$ be the *a posteriori* estimate of the error covariance matrix and the true state at timestep k, respectively. The filter starts with initial estimates for the true state $\hat{x}_{k-1|k-1}$ and the error covariance matrix $P_{k-1|k-1}$, and then repeatedly executes its *predict* and *update* phase routines. Refer to [130] for a more detailed introduction to the Kalman filter.

The Steady-state Kalman Filter with Constant Velocity Model The Kalman filter used in our implementation is a particular type of the general Kalman filter in

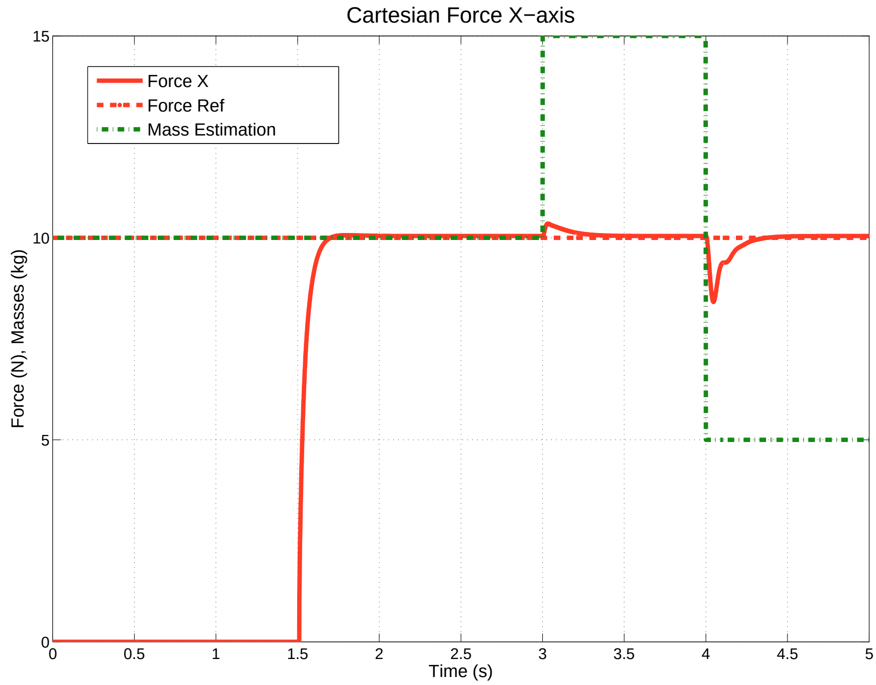


Figure 6.12: Robustness of the controller against changes on the mass of the robot's links

which a constant velocity model is assumed. The constant velocity model is usually used in tracking applications [66, 5, 99] and is also known as an $\alpha\beta$ filter. Since we assume that the system's velocity does not change dramatically, we are able to assume a constant velocity model. The steady-state version of the Kalman filter is used in cases where the time required to compute the algorithm is an important constraint. For a given system, one can let the Kalman filter run for several cycles and record the Kalman gains K in steady state. These will be constant, so the computation can easily be sped up by always using these constants instead of updating K each cycle (which requires a matrix inversion computation). The equations that describe the steady-state Kalman filter are:

$$\hat{x}_{k|k-1} = A \cdot \hat{x}_{k-1|k-1} \tag{6.19}$$
$$\tilde{y}_k = z_k - C \cdot \hat{x}_{k|k-1} \tag{6.20}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K \cdot \tilde{y}_k \tag{6.21}$$

where $\hat{x}_{k|k-1}$ represents the estimate of x at time k given observations up to and including time $k-1$. z_k is the measurement at time k, A is the state transition matrix, C is the output array and K is the steady-state Kalman gain. Expression (6.20) computes the innovation factor that allows the predictions to be updated after new measurements have been obtained. Given the assumption of a constant velocity model, the filter will choose two weighting coefficients (α and β) that will weight the differences

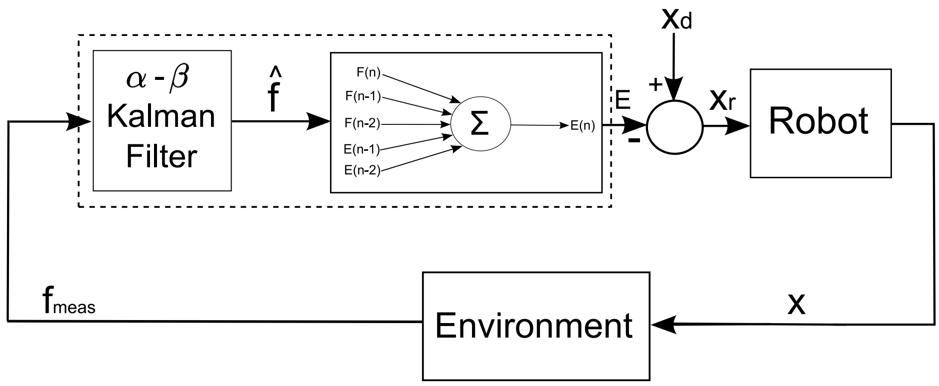


Figure 6.13: The $\alpha - \beta$ Kalman filter is used to estimate the sensor value $\hat{f}$ from the measured (noisy) value f_{meas}. The parameters of the blocks enclosed under the dotted lines are obtained using evolution strategies

between predictions and new measurements when updating the current prediction to find a new estimate. To better illuminate this, consider the classical tracking equations for the $\alpha\beta$ filter:

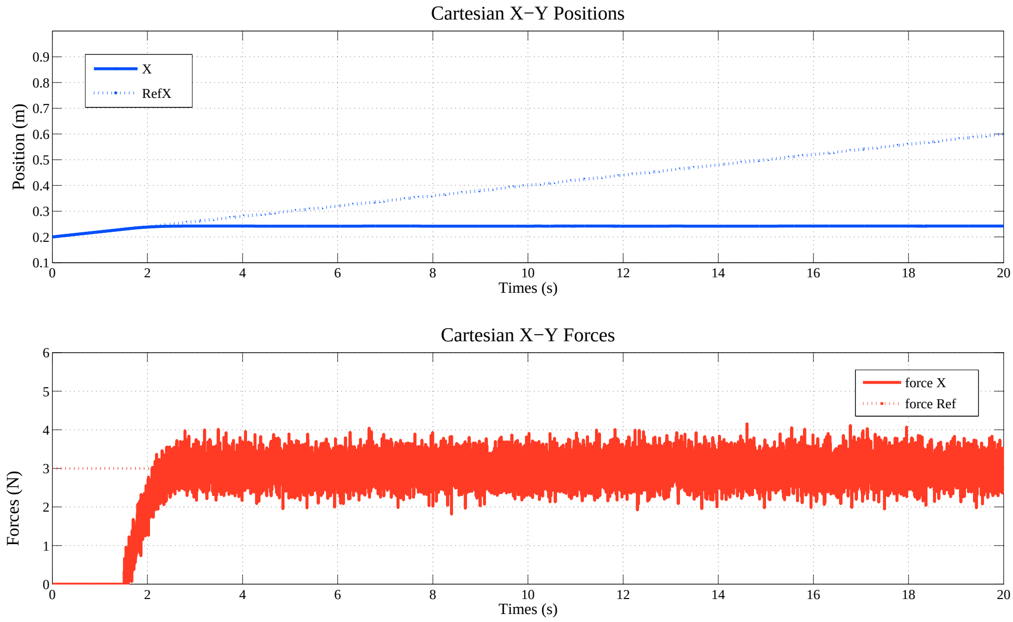
$$x_p(k) = x_s(k-1) + v_s(k-1)T \tag{6.22}$$
$$v_p(k) = v_s(k-1) \tag{6.23}$$
$$x_s(k) = x_p(k) + \alpha(z_k - x_p(k)) \tag{6.24}$$
$$v_s(k) = v_s(k-1) + (\beta/T)(z_k - x_p(k)) \tag{6.25}$$

where $x_p(k)$ and $v_p(k)$ are the predicted position and velocity at time k, $x_s(k)$ and $v_s(k)$ are the smoothed position and velocity at time k, T is the sampling time, and α and β are the weighting coefficients. After calculating $x_p(k)$ and $v_p(k)$ (eqns. 6.22 and 6.23), the calculation of the smoothed parameters only requires the proper selection of values for α and β. The optimal values for α and β have been derived by Kalata [66], and depend on the assumed variance of both measurement and process noises (σ_v and σ_w):

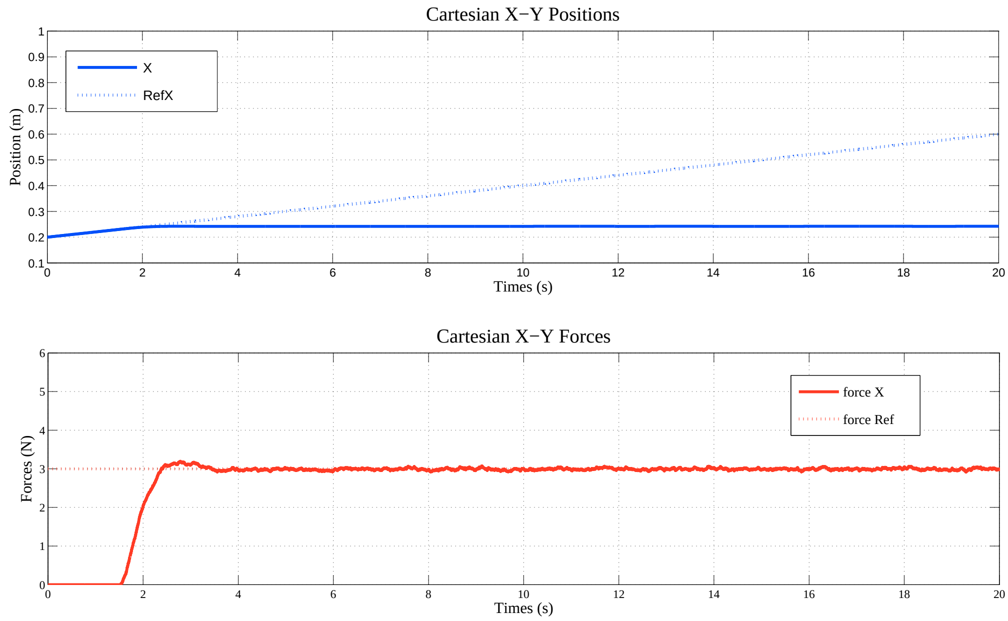
$$\gamma = \frac{T^2 \cdot \sigma_v}{\sigma_w} \tag{6.26}$$
$$r = \frac{4 + \gamma - \sqrt{8 \cdot \gamma + \gamma^2}}{4} \tag{6.27}$$
$$\alpha = 1 - r^2 \tag{6.28}$$
$$\beta = 2 \cdot (1-r)^2 \tag{6.29}$$
$$K = \begin{bmatrix} \alpha \\ \beta/T \end{bmatrix} \tag{6.30}$$

The state transition matrix A is initialized the with a constant velocity model:

$$A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{6.31}$$



(a)



(b)

Figure 6.14: Robustness against noise: (a) evolving the controller without a Kalman filter, (b) weights of the ANN-based controller and the Kalman filter evolved simultaneously

and the output array C is $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$, where the '1' in the first column indicates that we have measurements from the force, and the '0' in the second column indicates that we have no information about the force change with respect to time (derivative of the force).

In our experiment, we use one Kalman filter for the force measurement. The measurement noise that is introduced to the system is a Gaussian signal with zero mean and standard deviation $\sigma_v = 10^{-1}$. This noise is added to the input f_{meas} (the force measurement) of the Kalman filter depicted in Figure 6.13. In this experiment, the same neural network structure was used as in the previous experiments, i.e. the one-neuron feedforward neural network with 5 inputs, 1 output, and 3 weights.

Additionally, the optimization of the Kalman filter was incorporated into the evolutionary process, where optimal values for the parameters σ_v and σ_w were searched for using CMA-ES (along with the weights for the neural network). Because the problem is simulated, the standard deviation of the measurement noise we are introducing is known and thus the initial value for σ_v in the Kalman filter can be set to this value. In the case of a real system, however, a set of real measurements could have been collected, the mean and standard deviation of the data set calculated, and the standard deviation used as the value for σ_v. In the case of process noise, manual tuning is typically used due to the complexity of determining the value of the noise. The Kalman filter, however, usually performs well with only a rough estimate of σ_w. Figure 6.14 shows the results of the experiments with noisy signals.

Figure 6.14(a) depicts the case of learning to track a specific force reference with a highly-noisy force measurement. As it can be seen, the algorithm is able to, despite the noise, learn a proper solution in order to achieve the reference force. However, the controller would be useless in a practical scenario since the robot would oscillate at high frequency around the contact point. In order to provide a compact solution, the Kalman filter presented previously is included in the evolution process. By doing that, we obtain a solution in one step: both the weights of the neural network and the parameters of the Kalman filter are obtained simultaneously, without requiring of a pre-processing of the measurement data. Figure 6.14(b) shows the response obtained with the system depicted on Figure 6.13. The robot is able to reach the targetted reference force and, at the same time, imperceptible noise remains on the force response of the robot, i.e. no oscillations occur on the contact.

6.5 Conclusions

The chapter described the design of an ANN-based impedance controller by using evolutionary techniques. The impedance controller is first discretized and represented as a neural network. The use of evolutionary techniques provides a simple methodology to evolve the controller requiring only the definition of a proper performance criteria to be optimised. Currently, unclear or cumbersome methodologies are found to select impedance parameters. The proposed approach obtains optimal parameters given a task to perform. Besides, it is shown how the classical impedance controller can be described as a single-neuron neural network with 5 inputs, 3 weights, and 1 output. Since the weights of the neural network bound the policy space of the controller, and in this case they are only three, the space of the possible inputs is unique. To generalise the controller for any given force reference input, an on-line estimator has been designed that estimates the weights for the current force reference. Using the values of a series of single-force controllers, the parameters of a polynomial are obtained that estimate the proper neural network weights for the current scenario. The resulting controller is able to track a great range of force reference inputs, a quality that is not intrinsically present on a classical impedance controller.

Moreover, the robustness of the controller is demonstrated by modifying both the robot and the environmental model parameters. The controller is able to set back to the current reference force after abrupt changes on the environmental stiffness, even when it is constantly kept to values 10% below or over the nominal one. Similarly, abrupt changes on the estimated masses of the robot links of up to 50% of the nominal value are absorbed by the controller, which is able to keep track of the current reference force.

Finally, the controller is enhanced with a Kalman filter to improve the controller's robustness against the measurement noise. Both the controller and the parameters of the Kalman filter are evolved simultaneously, thus providing a one-step solution which does not require a pre-processing of the measurement data used to learn the solution.

Chapter 7

PREDICTIVE COMPLIANCE ON A DUAL-ARM ROBOT MANIPULATOR

In this chapter the last component of the presented architecture is introduced: the context-based prediction for modifying and/or updating the current compliance of the robot.

The chapter concludes with the results of two experiments using a dual-arm robotic platform. The experiments show the use of the predictive module in two different situations: reproducing the previously introduced 'Waiter Task' and using the context-based prediction during the lifting of look-alike objects

7.1 Introduction

In this final chapter, we aim at developing the highest hierarchy level of the PCAC architecture (Fig. 4.1) proposed in this thesis. Figure 7.1 highlights the components developed in this chapter: a Bayesian-based predictor to anticipate the consequences of self-generated actions and perform, if necessary, corrective measures via the impedance controller.

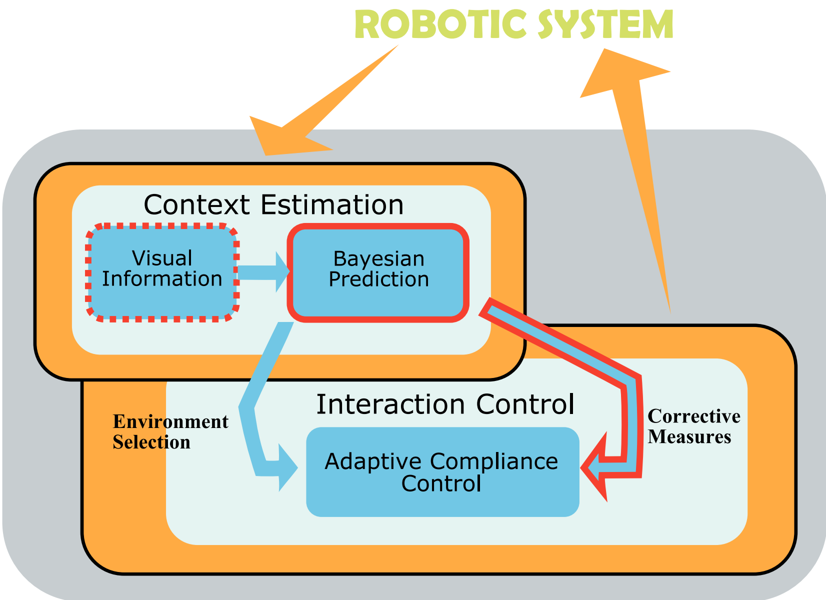


Figure 7.1: Proposed architecture for compliance control via context estimation. The components highlighted are discussed in this chapter

7.2 Methods

This section describes the methods and tools used for carrying out the two experiments which form the basis of this chapter. Moreover, this section also shows the results obtained by reproducing the experiments with human subjects. The first experiment is the so-called 'Waiter Task', in which the anticipation of self-generated actions is used for compliance regulation. The second experiment is the so-called 'Predictive Lifting', in which a context prediction is used to predict the sensory consequences of lifting a carton of milk. This prediction serves to select and pre-regulate the arm compliance before the task as well as to modify it during the execution of the task (in case of a wrongly-estimated context).

7.2.1 Anticipating upcoming actions for arm compliance regulation

Estimation has been shown to play an important role in biological systems. Recent neurophysiological studies in primates show how their brains predict the nature of different environments. These studies show for example that the Central Nervous System (CNS) models the expected feedback by using forward models [129][37]. As mentioned on Section 4.4, the Waiter Task [57][35] shows the predictive mechanisms controlling the arm's compliance adaptation to a changing context. This experiment allows to describe the predictive mechanisms controlling the unloading of objects from a tray with one hand (dominant) while the other hand (non-dominant) holds and keeps the tray and arm at a fixed posture. The results evidence that both the position of the tray and the non-dominant arm remain unchanged despite the changes on weight and forces during the unloading. Interestingly, the non-dominant arm (and thus the tray) just remain unchanged when it is the dominant arm the one that removes the objects from the tray. In case that an external person removes the objects (even when the person is informed about the removal action), the non-dominant arm (and tray) change its position. Further experiments also showed that the change of position diminishes after some trials, but there is always some perceivable change that the person cannot avoid as much as he/she wants to. From these observations, two conclusions about the control of the arm can be extracted: (a) first, the fact that the arm's compliance is regulated in order to adapt to the changing context, and (b) the predictive nature of the anticipatory actions to be performed. As it was mentioned in Section 4.4, the delay present on biological feedback systems would not allow a prompt reaction to overcome such a sudden load change if there were no prediction and preparation for the change. Figure 7.2 shows the results of the original experiments with human subjects reported in [35].

A similar experiment as described in the original paper [57] was reproduced during this work using a human subject. The purpose was to confirm these results more from a curious point of view rather than aiming at providing statistically significant results comparable to the original neurophysiological studies. In this case, the elbow of the subject was not tied to the chair (as in the original work) but instead the subject was told to leave it in contact with the armchair. That is, the only motion would be that of the elbow and wrist joints.

Figure 7.3 shows snapshots of the video recorded during the realisation of the experiment. The first row shows a self-unloading action. The subject had the opportunity to manipulate the object before the experiment took place. The fact of having previous experience with the object did not seem relevant since the subject would anyway sense the weight of the object as soon as it is placed on the tray. The third image on that row shows the moment where the object is completely lifted and, at this point, it can be observed that there is no noticeable change on the position of the tray and/or the non-dominant arm.

The second and third rows show two imposed unloadings. The subject is aware of the fact that the object will be removed (he can see and follow the action of the third person removing it). This time though, the third image on the corresponding

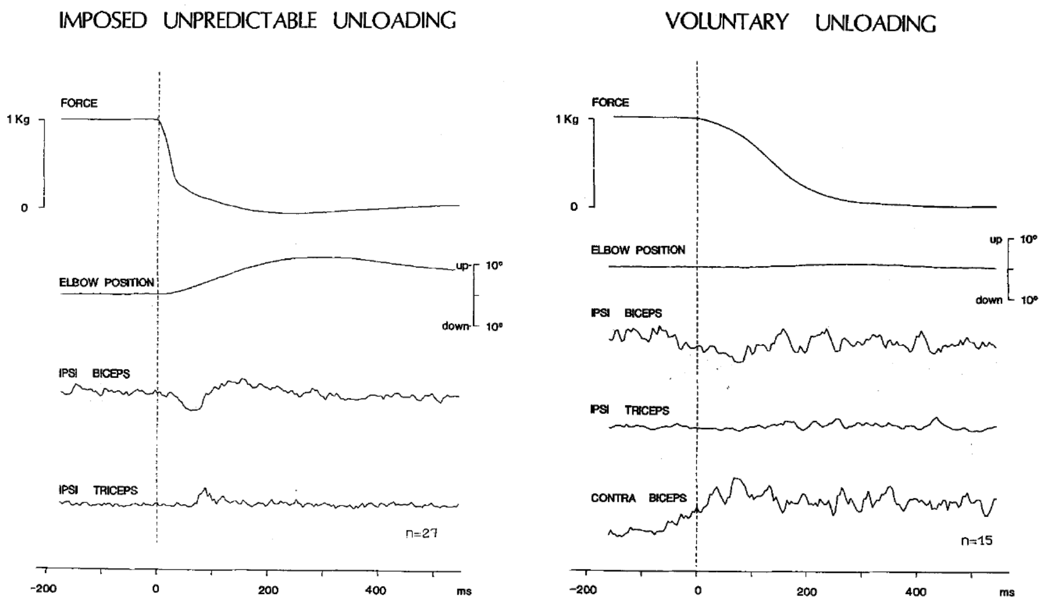


Figure 7.2: Original figure of a human arm performing the Waiter Task from [35]. Note the change in elbow position when the unloading is imposed in comparison to when it is voluntary

row shows a noticeable change on the position of the tray. No matter how strong is the subject's willingness to hold the non-dominant unchanged, there is an evident movement. However, as described in literature, the subject learns after some trials to reduce the movement. This fact can be also observed between the change on position observed in the second and third rows (third image). The third row shows a smaller change in position although subsequent trials will not be able to remove it completely or significantly.

7.2.2 Context prediction for arm compliance regulation

A second experiment involved the context prediction as described in Section 4.4. In this case, the experiment was also first reproduced with human subjects. The context prediction is exemplified by lifting a carton of milk and by intentionally causing false predictions with respect to its weight. For that purpose, the person is told that he/she has to lift a full carton of milk from point A to point B, lifting it to pass over the obstacle, and as fast as possible. Two apparently identical cartons of milks lie on a table and the person is told to move one carton after the other in a specified order. What the person does not know is that the second carton of milk is empty. Visually, both cartons appear to be full (the empty one was inflated with air to appear identical to the full one). The person lifts the first carton and his/her prediction about the weight of a carton of milk corresponds with what he/she senses. There are no surprises and the action is completed successfully. Then the person lifts the second object, predicting certain forces under the assumed context of a full carton of milk. The result

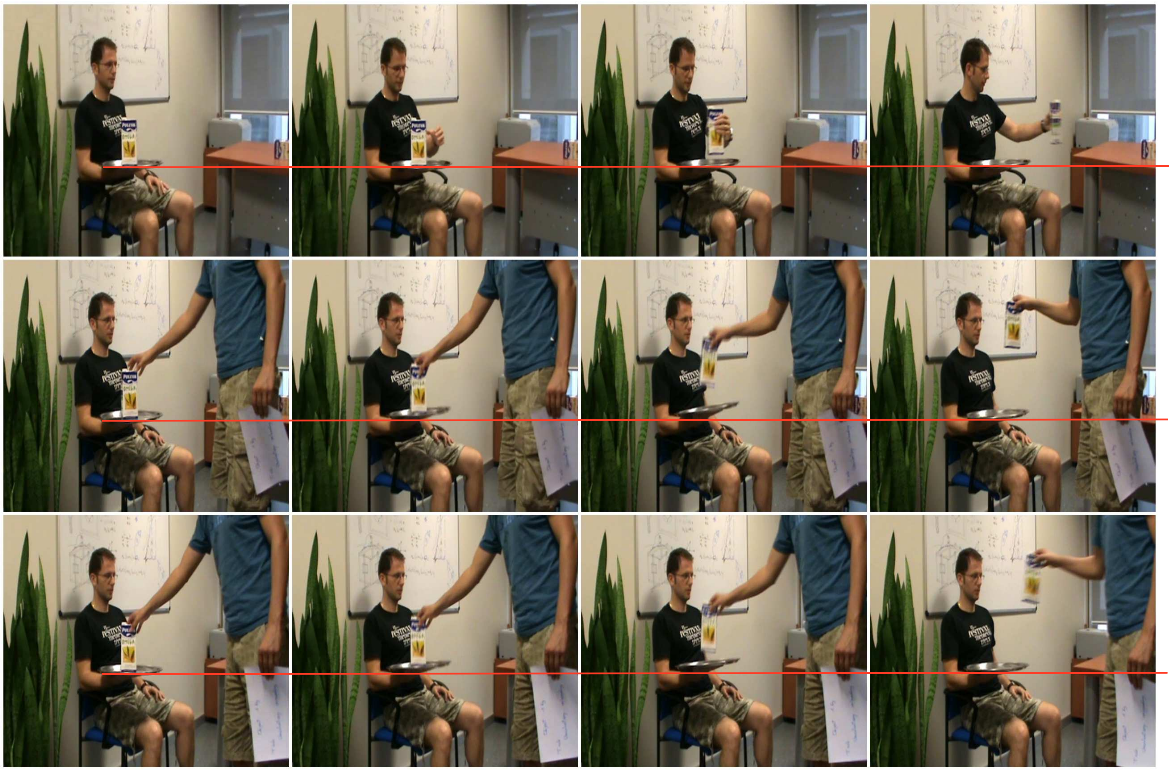


Figure 7.3: Human performing the Waiter Task. First row shows a self-unloading action, where no change is observed on the position of the non-dominant arm (third image). On the contrary, the imposed unloadings (second and third rows) show an evident change on the position of the non-dominant arm.

is clear to imagine: the person lifts the object more and faster than expected , becomes conciously aware of the false prediction, and immediately corrects the action in order to accomplish the task. The experiment was performed with ten people and, although all of them were tricked with the false assumptions, some reacted quicker and smoother than others. Figure 7.4 shows one of the subjects whose response was externally clearer to observe because of the overreaction caused by her false prediction.

7.2.3 Context estimation via Bayesian inference

As we have previously seen in Section 3.2.6, context estimation plays an important role on compliance adaptation by deciding for an initial arm compliance. Moreover, the use of a Bayesian model provides with the capability of updating initial predictions according to the incoming measurements. As previously mentioned, Bayesian inference is a statistical method which derives conclusions by collecting evidence about the trueness or falseness of a hypothesis. The probabilities of an event happening are continuously updated as new evidence is available. Basically, it is composed of a prior probability that represents the knowledge about the occurrence of an event before any evidence has been gathered and a likelihood function representing the probability that

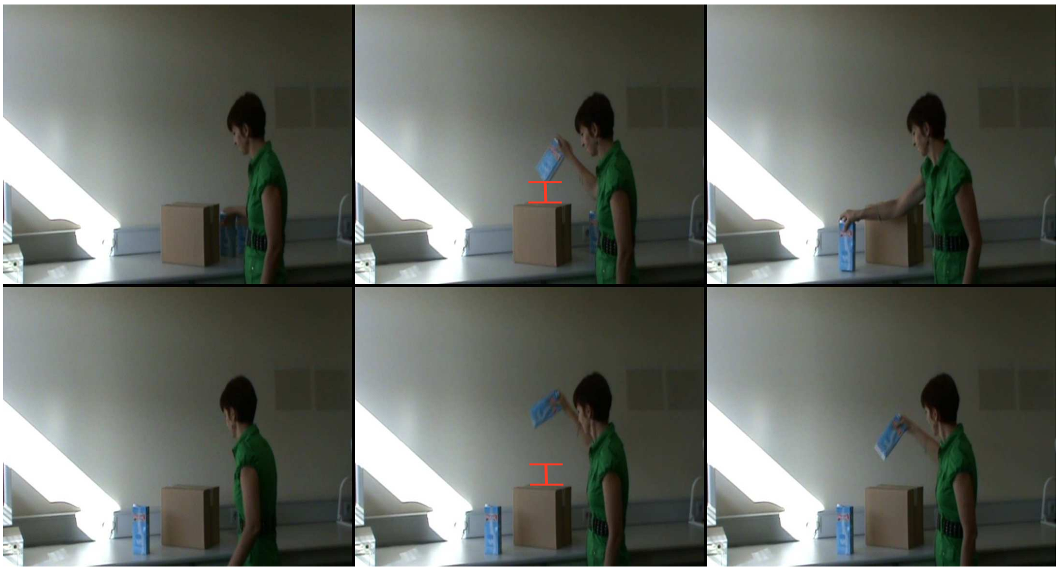


Figure 7.4: Human performing a lifting task. First row shows a lifting action of the full carton of milk. The second row show the lifting action of a carton of milk that the person thinks it is full but it is, in reality, empty. Notice the difference on the lifting height, due to the false prediction of the object's weight.

a particular evidence is observed given that a particular hypothesis is true. Bayesian inference gives as an outcome a new final probability after an evidence is observed, that updates the current probability. Using this statistical method, we analyse a priori our data and update our predictions as soon as new sensory data is available.

On our experiment there are a discrete set of different contexts $C = \{C_1, C_2, ..., C_N\}$, the probability of which is considered as $p(C_j) = \frac{1}{C}, C_j \in C$, that is, all contexts have the same a priori probability to be contacted. The likelihood function $p(x|C_j)$, the probability of sensory data vector x occurring given that the hypothesis of C_j is true is learned via training where the robot lifts the different objects and encounters the different contexts. Immediately after starting lifting an object, the algorithm uses the perceived sensory data vector x to compute the posterior probability of an environment $C_j, C_j \in C$. In this case, the sensory data vector x is formed by the readings of the position and force along the Cartesian axis Y at a fixed sampling time t while lifting the object, just after closing the gripper around the object. Assuming n time steps, the sensory data vector is in the form $x = [x_1, x_2]$ where $x_1 = [x_{pos_Y}(0), ..., x_{pos_Y}(n)]$ represents the Cartesian position along the Y axis and $x_2 = [x_{f_Y}(0), ..., x_{f_Y}(n)]$, stands for the Cartesian force along that axis.

7.2.3.1 Relevance vector machines

Generally, a complex learning problem requires a vast amount of training data and the training time is proportional to the amount of training data. That is why most of the time it is not feasible to be done on-line. In our implementation, the training is done using a Relevance Vector Machine (RVM) [122]. A RVM is a supervised learning method which applies to modelling nonlinear mappings of high-dimensional input feature space to a scalar-valued output space. RVM is functionally equivalent to a support vector machine but regarded as a sparse Bayesian classifier because of: (a) the sparsity of the stored model (thus providing fast and efficient classification) and (b) the output of the RVM is a probability function. Given a set of input vectors $\{X_n\}_{n=1}^{N}$ and corresponding targets $\{t_n\}_{n=1}^{N}$ (in our case, classification labels), we aim at learning the relationship between targets and inputs in order to classify future inputs to the proper class. The predictions are based on a function $y(x)$ over the inputs and the learning is the process of inferring this function. A set of candidates for $y(x)$ is that of the form:

$$y(x;w) = \sum_{i=1}^{M} \omega_i \psi_i(x) = w^T \phi(x) \tag{7.1}$$

where the $\phi(x)$ is a non-linear mapping (basis function). When trying to find w from training examples, we assume that each target t_n is representative of the true model y_n, but with the addition of a zero-mean Gaussian noise ϵ_n of variance σ^2, that is, $\epsilon_n\ N(0, \sigma^2)$.

7.2.4 Compliance Adaptation

The dynamic relation between the current end-effector position $X(t)$, the desired Cartesian trajectory ($X_d(t)$) and the end-effector force $F(t)$ can be described as

$$\begin{aligned} M(\ddot{X}_d(t) - \ddot{X}(t)) + B(\dot{X}_d(t) - \dot{X}(t)) + \\ K(X_d(t) - X(t)) = F(t) \end{aligned} \tag{7.2}$$

If the compliance controller is not present, $X_d(t)$ equals $X_r(t)$ in Eq. (7.2), thus the resulting robot tracks accurately position commands. As it has been seen in previous chapters, defining an impedance controller as $G(s) = M_T s^2 + D_T s + K_T$, the dynamical response of the robot can be easily controlled by modifying the reference trajectory $X_r(t)$ as:

$$\begin{aligned} M_T(\ddot{X}_d(t) - \ddot{X}_r(t)) + B_T(\dot{X}_d(t) - \dot{X}_r(t)) + \\ K_T(X_d(t) - X_r(t)) = F(t) \end{aligned} \tag{7.3}$$

where M_T, D_T and K_T are the inertia, damping and stiffness coefficients, respectively, that will define the dynamic behaviour of the robot. Assuming that our contact desired accelerations and velocities are zero ($\ddot{X}_d(t) = \dot{X}_d(t) = 0$) and we have ideal joint controllers that achieve $X_r(t) = X(t)$, the following equation describes the behaviour of the impedance controller:

$$M_T \ddot{X}(t) + B_T \dot{X}(t) + K_T(X(t) - X_d(t)) = -F(t) \tag{7.4}$$

If the interest lies only on the steady-state response of the system, the inertia and damping matrices can be obviated to build a so-called stiffness control, which

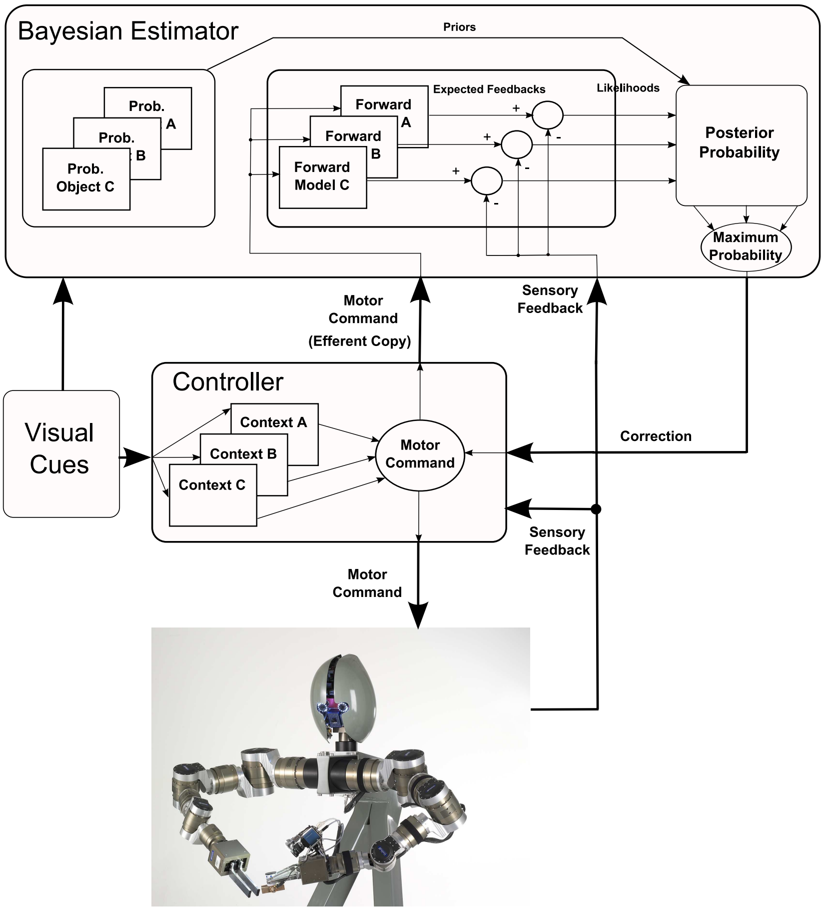


Figure 7.5: Bayesian Estimator for compensation of erroneous initial predictions about the nature of an object

will accordingly adapt the stiffness of the robot to the current situation. For the experiments of this chapter, which primarily focus on the decision required for changing the compliance of the arm and, secondarily, on the structure the compliance controller, a static stiffness controller has been used:

$$K_T(X(t) - X_d(t)) = -F(t) \tag{7.5}$$

7.3 Experiments

This section shows the experimental results obtained with 'Mr. SemProM' (the dual-arm robot manipulator) after performing the previously described tasks.

7.3.1 'Waiter Task' Experiment

The first experiment reproduces the classical 'Waiter Task' experiment. This experiment shows the benefits of anticipating consequences of self-generated (robot) actions.

As in the classical experiment, the robot holds with the left arm a tray with an object. The object can be removed from an external person or by the robot itself using its right arm. When the robot uses its right arm, the prediction module is enabled with a previously trained forward model, thus predicting a change on load and anticipating the necessary changes on compliance for the arm holding the tray. The moment the object is removed, the new compliance has been set for the left arm, thus we do not observe a postural change on that arm. On the other hand, if the object is removed from the tray by an external person, the robot cannot predict the change on load and thus the arm behaves as a normal spring, returning to its resting equilibrium point. Thus the compensation mechanisms come only into play for robot's self-generated actions, as seen in neuroscience results in the Waiter Task.

7.3.2 Results

Figure 7.6 shows the response of the system for imposed and unpredictable unloading (external person removes the object from the tray) and for voluntary unloading (robot's self-generated action). In the first case, the unloading causes a big displacement on the position of the robot's arm: the unloading was not expected and the compliance of the arm was not adapted to the new situation. In the second case, voluntary unloading, the robot knows beforehand the consequences of its unloading action, so the stiffness of the robot is adjusted just before the removal of the object. In the first case, the stiffness of the robot is set to $K = 176N/m$. For the second case, the initial stiffness K is the same as the previous, but as soon as the decision is taken to unload the object from the left arm, the stiffness is set to $K = 8kN/m$.

Figure 7.7 shows some snapshots of the video recorded during the realisation of the experiment. The first two rows show what happens when a person removes the object from the robot's left arm (imposed unloading situation). The left arm goes back to its equilibrium position dictated by its defined impedance. As the object is placed on the arm, the arm changes its position. If a person removes the object, the arm returns to its original equilibrium point.

The third row shows the voluntary unloading, that is, an unloading triggered and performed by the robot itself by using the right arm. In this case, its predictive modules are enabled in order to account for the upcoming change on load. The arm stiffness is going to be adequately adjusted and synchronised with the moment prior to the unloading. As it can be observed, there is no observable change on position of the left arm. Once the object is unloaded, the arm returns to its original soft stiffness and will set the current position as its new equilibrium position. The last row shows that upon a new imposed unloading the robot shows the behaviour observed in the first two cases, that is, its position is modified according to the changes on load.

7.3.3 'Predictive Lifting' Experiment

The second experiment deals with the context prediction. In this scenario, the robot should make use of some cues (in a practical scenario would primarily be of visual type) to generate a prior probability for the most probable object that lies in front

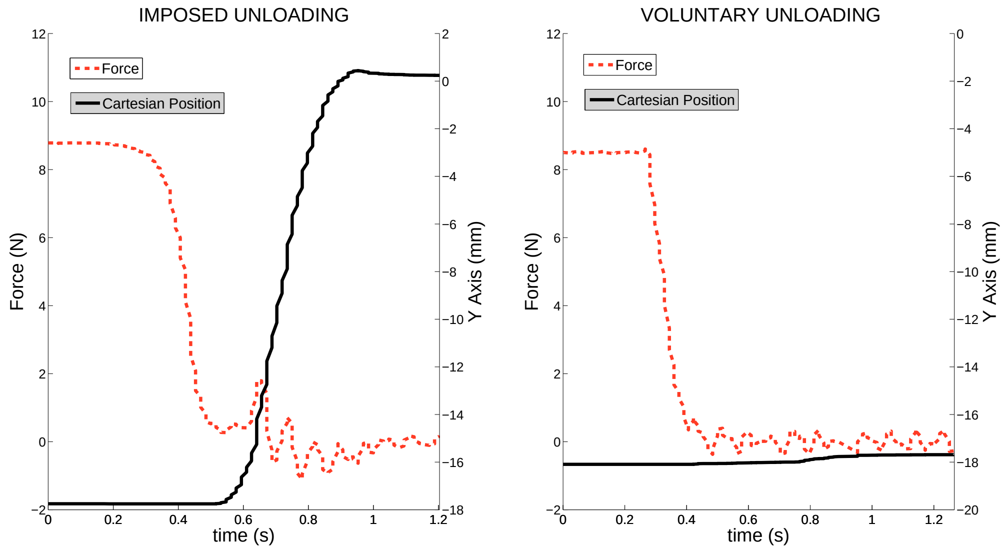


Figure 7.6: Results obtained on the Waiter Task using a dual-arm robotic manipulator. The left figure shows an imposed unloading where the change on arm's position can be clearly observed (black solid line). The right figure shows a voluntary unloading performed with the righ arm of the object held on the left arm. In this case, a minimum variation of the arm's position is observed due to the active change of the arm's stiffness prior to the unloading.

of the robot and that it is going to be lifted. This information sets the priors on the Bayesian model scheme shown in Fig. 7.5 and selects a suitable controller. Using learned forward models after a training phase, the robot will generate expectations for the sensory feedback to be received. If the expectations meet the current sensory feedback, no action needs to be taken. If otherwise, the Bayesian model indicates with a high probability that a different object as initially predicted is being lifted, correction actions are undertaken in order to change the controller's behaviour according to the most probable object.

The correction actions to be taken are related to changes on the stiffness of the arm. In order to reproduce a visually-dramatic effect similar to the one observed when human subjects performed a similar task (Figure 7.4), the robot had to lift, in this case, an empty carton of milk that, in reality, it is full. In this way, it is clearly observable that the stiffness of the arm is not sufficient to lift the object and the task fails repeatedly. Once the Bayesian prediction module and correction control paths are enabled (see Fig. 7.5), the robot will be able to sense and update the initial prediction in order to change the stiffness (increase it, in this case) and be able to successfully lift the object.

In these experiments, the Relevance Vector Machine has been previously trained with robot data from manipulating different objects. When the robot receives initial information about the object, it generates a prior probability about the most probable object and decides for a specific arm stiffness (higher for a heavier object, lower for a

Figure 7.7: Robot performing the 'Waiter Task'. The first two rows show imposed unloading situations; the first row shows a person placing the object on the robot's left arm and its subsequent movement. The second row shows how the robot's left arm moves back to its original equilibrium position when the object is unloaded. The third row shows the voluntary unloading, in which the right arm unloads the object from the left one. In this case, there is no visible change in position due to the prior change of compliance determined by the action to be executed. The last row shows that upon a new imposed unloading the robot shows again the same behaviour as on the first two rows, that is, its position is modified due to the change in load.)

lighter object), that is, chooses from a controller $\Gamma \in \{ContextA, ContextB, ContextC\}$. This, in turn, automatically generates expectations on the sensory feedback to be received. This is depicted in Figure 7.5 as the efferent copy pathway, in which a copy of the motor command is sent to the Bayesian estimator in order to be able to compare expected with real sensory feedback. As soon as the object begins to be lifted, the expected and the real measurements are compared in order to verify the trueness of the hypothesis. In case of a false hypothesis, the controller's context will change to the most probable object, thus dynamically and on-line changing the arm stiffness accordingly.

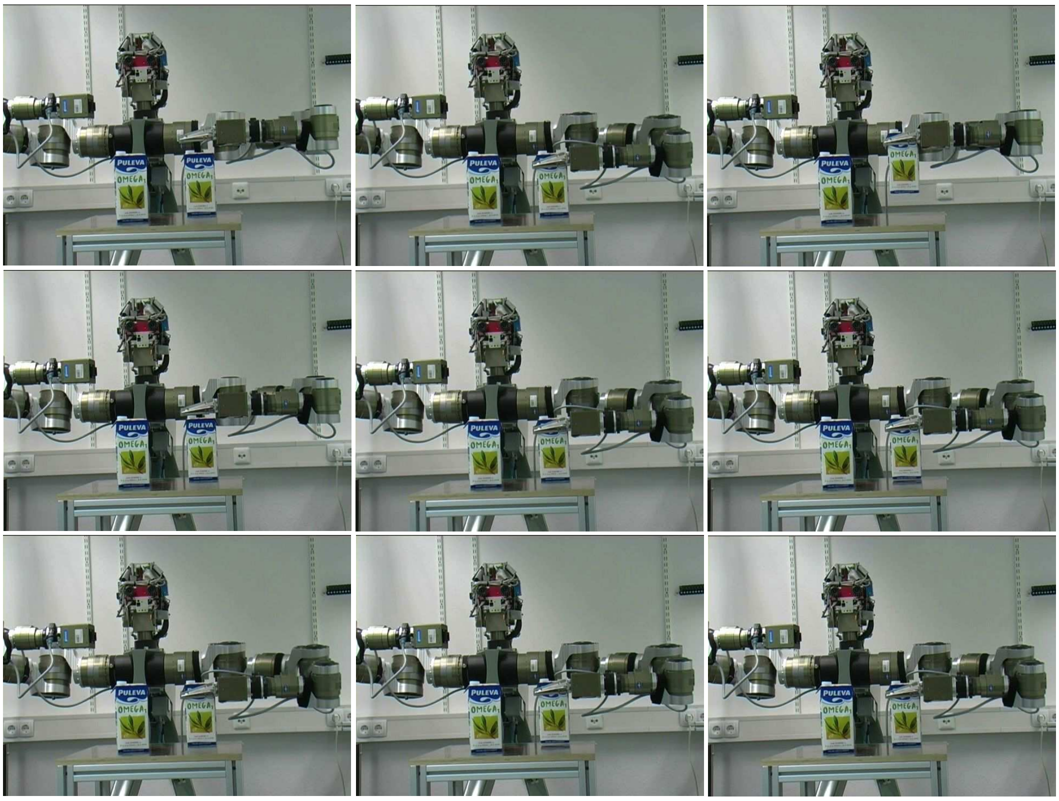


Figure 7.8: Robot performing the lifting experiment. The first row shows the case where the robot expects an empty object, finds a full one, but it is able to update the initial erroneous prediction and update the arm stiffness to lift the object. The second and third rows show the attempts of the robot to lift the full carton of milk when the predictor/correction modules are not enabled. The robot expects the empty object, the arm stiffness is set accordingly to a low value and the arm is not able to lift the weight more than a centimeter to fall down again.

Data processing Training data from lifting the different objects is used to train the Relevance Vector Machines and find a suitable decision function that can classify according to the examples given by our training data. The data is two dimensional including position and force information measured during the lifting process. Three different classifiers are created (one for each object). The first step of the pre-processing is to normalise the data by substracting their mean and dividing by the standard deviation. After that, and to avoid specific ordering of the data that would hinder a successful cross validation process, the data ordering is randomized. At that point the data is ready to be used for training. However, there is an additional step. Our RVMs use as kernel a Radial Basis Function (RBF), which includes a parameter that needs to be carefully chosen: the γ factor. The kernel of a RBF is defined as:

$$\phi = e^{-\gamma|x_i - x_j|^2}, \gamma > 0 \tag{7.6}$$

There are some heuristics that might be used to select an optimal γ, but the most

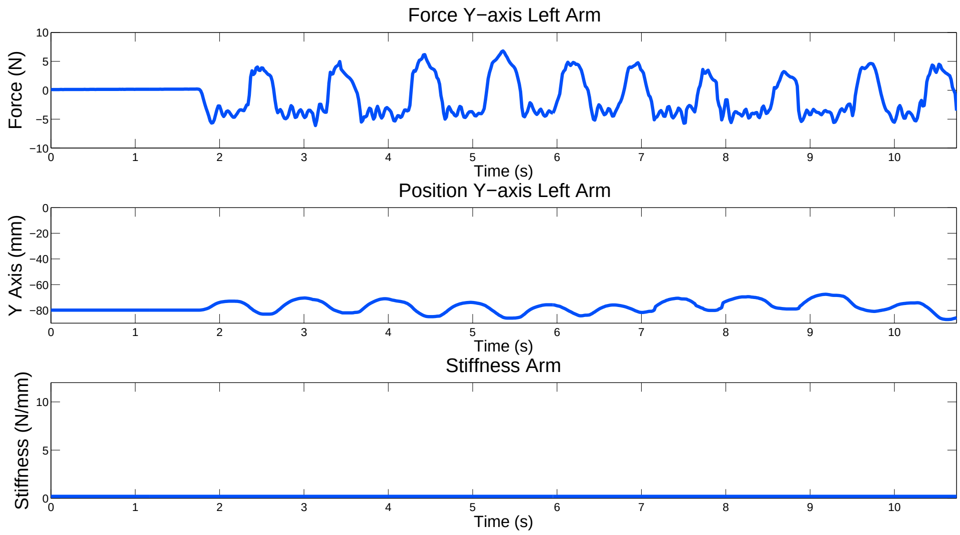


Figure 7.9: Robot response while trying to lift a full carton of milk. The robot expects an empty carton and, accordingly, the stiffness is set to a low value. In this case, the prediction and correction modules are not enabled. Consequently, the robot is not able to adapt to the unexpected situation and it tries unsuccessfully to lift the object but fails repeatedly.

common procedure is to perform a cross-validation process with the training data that is the procedure that was followed for these experiments. In our case, we performed a ten-fold cross validation on our training data for $\gamma \in [0, 1]$ using a step on gamma of 0.1. The best gamma value was accordingly selected to start the training with the full data set in order to find the optimal decision function.

The typical RVM (similarly to what occurs with a Support Vector Machine (SVM)) is a binary classifier, that is, it works with pairs of classes. Since we have more than two objects, i.e. more than two classes are to be identified, a multi-class RVM classifier $y \in \{1, ..., k\}$ was constructed. The two most common solutions to construct a multi-class classifier are either to construct a one-against-all classifier or to use a one-to-one classifier for all possible pairwise combinations. We chose to use the first approach, a one-against-all classifier. That is, the multi-class classifier is built by learning one binary RVM per class y_i, and assigning labels according to the following rule:

$$y_i = \begin{cases} +1 & \text{if } y = i \\ -1 & \text{else} \end{cases} \tag{7.7}$$

7.3.4 Results

The experiment setup did not include visual information to generate the proposed visual cues in Fig. 7.5. Instead, the prior initial prediction was given manually to the robot. This is due to the fact that the goal of the experiment was to prove the ability of

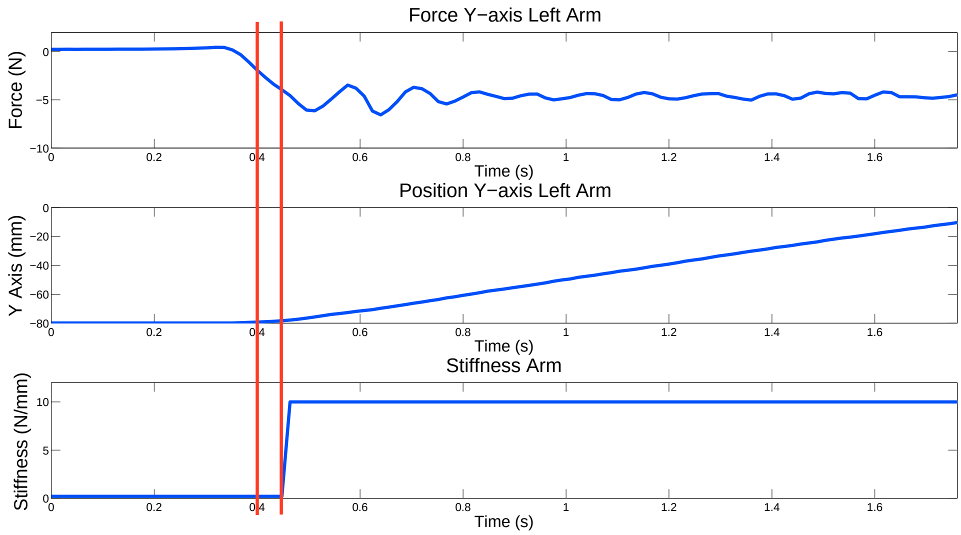


Figure 7.10: Robot response while trying to lift a full carton of milk. The robot expects an empty carton and, accordingly, the stiffness is set to a low value. In this case, prediction and correction modules are enabled. At the first stages of lifting, the robot realises that the initial prediction was wrong (time between the vertical red lines), updates the prediction, and generates a correction action that involves increasing the stiffness.

the algorithm to perform given a false prior prediction. The robot would receive a false initial prediction, in this case, that the object was empty when in reality it was full. This would force the robot to generate expectactions for the sensory feedback to be obtained from an empty object, and verify its accomplishment while lifting the object. Since the object is in reality full, the robot is required to recognise the discordance between expected and real sensory feedback and apply a suitable corrective measure. In this case, an increase of the stiffness of the arm that allows the robot to lift the full object. For our experiment, the low arm stiffness value is set to $K = 200N/m$. For the high stiffness case, the arm stiffness is set to $K = 10kN/m$.

Figure 7.8 shows the robot performing the 'Predictive Lifting' experiment, in which the first row shows the above described situation where the robot tries to lift an object expected to be empty when in reality it is full. Since in this case the Bayes predictor is enabled, the robot is able to sense the mismatch between the expected and real measurements and apply a corrective measure in the first miliseconds of the lifting movement. As a matter of fact, the correction is not observable during the real-time execution due to the prompt change of the arm stiffness. On the other side, the second and third rows in Figure 7.8 show the robot failing at lifting the object due to the fact that the prediction module is not enabled. Since the initial information tells the robot that it is an empty object, the robot is not able to perceive the wrong stiffness choice and keeps trying to lift the object indefinitely.

Figures 7.9, 7.10, and 7.11 plot the variables sensed by the robot (position and force) while attempting to lift an empty carton. Additionally, the stiffness of the arm during the experiment is also depicted. From these measurements and by comparing them with the expected ones from the output of the RVM predictors, the robot takes the decision to change or maintain the current stiffness. Figure 7.9 shows the case in which the predictors are not enabled. Consequently, the robot's stiffness is too low to lift the full object and it enters an oscillatory cycle, in which the robot manages to lift the object a few millimeters from the ground to fall again down and start over again. Figure 7.10 shows the case in which the Bayes predictor is enabled. On the first stages of the lifting the robot realises that the mismatch between expected and real measurements is too big and, accordingly, updates the prediction. In this case, the result is that the object with highest probability is the full one and the stiffness is immediately changed. Figure 7.11 shows the response of the robot while lifting the empty carton of milk. In this case, prediction and correction modules are enabled but this time the robot just confirms that that initial prediction is true, that is, that the object is empty and, accordingly, there is no need to modify the initial arm stiffness.

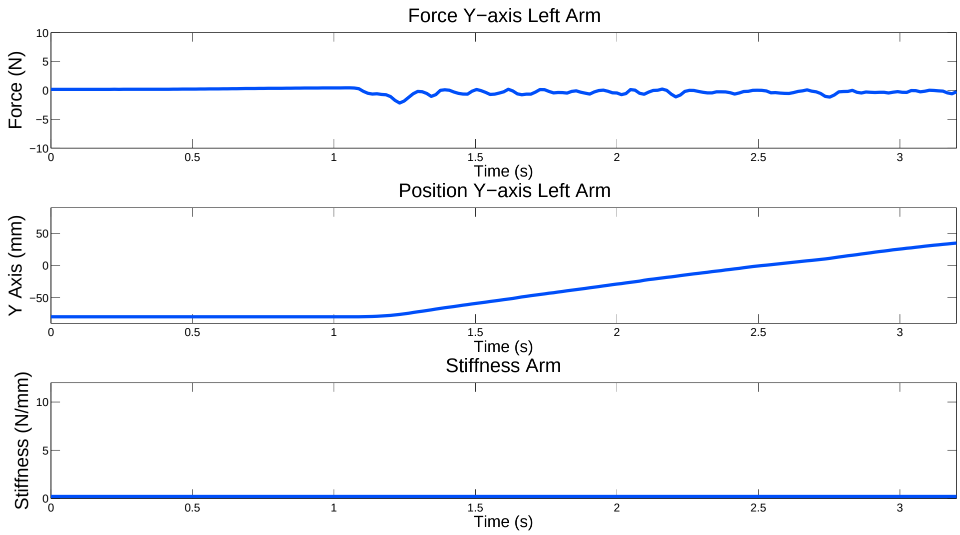


Figure 7.11: Robot response while trying to lift an empty carton of milk. The robot expects an empty carton and, accordingly, the stiffness is set to a low value. Prediction and correction modules are enabled. At the first stages of lifting, the robot confirms that the initial prediction is correct and thus needs no corrective action.

7.4 Conclusions

This chapter developed and tested on a real robot the context-based Bayesian prediction for pre-regulating and/or updating the current compliance of the robot. By using two different experiments, the utility of the context prediction was demonstrated. A first experiment ('Waiter Task') uses the knowledge about the actions to be performed and their expected outcomes in order to pre-regulate the compliance of the arm. This involves synchronisation with the predicted actions and their corresponding effects. A second experiment ('Predictive Lifting') uses the knowledge about different arm stiffness related to different contexts in order to deploy a context-based predictor that decides for the proper arm compliance. Moreover, the architecture is able as well to compensate in real time for wrong initial choices.

Chapter 8

CONCLUSIONS AND OUTLOOK

This chapter summarises the results of this thesis and gives an outlook on possible further experiments and improvements

8.1 Final Conclusions

This thesis aims at contributing to the state of the art in the control of the interaction between a robot and its environment. The first approaches of controlling interaction forces of the immense and stiff industrial robots started with basic force controllers more than thirty years ago. Yet limiting the maximum forces is not a solution for situations outside of the controlled environments present in factories. The upcoming robotic revolution that will deploy robots in close interaction with humans requires higher levels of intelligence to not only control and limit forces, but most importantly, to adapt to different, and possibly new, situations. Hence, the main focus of this thesis has been on providing new insights in the area of interaction control. The initial literature review revealed important deficiencies and non-solved issues that the thesis attempts to wrap into a sensible architecture. Most especially, a higher level of intelligence has been incorporated by including predictive mechanisms similar to the ones found in biological systems. Nevertheless, the proposed architecture needs to be understood as a picture of what blocks are required for the synthesis and high-level intelligent control of the interaction controller, but not as a software architecture. What follows is a summary of the conclusions and contributions of this work:

> The first problem that was observed is that the implementation of the increasingly popular impedance controllers (not so popular at the start of this thesis) strongly relies on the quality of the information that is available from the working environment. The first contribution of this thesis has been to propose an environment identification method. This approach increases the success rate of choosing the appropriate impedance controller when contacting previously known environments. It also enables the controller to extrapolate and apply actions to slightly different, but similar environments.
>
> The second problem which is investigated in this thesis is the definition and selection of the parameters that define an impedance controller instance. Literature dealing with appropriate methods to select the parameters of an optimal impedance controller for a given environment is scarce. This work aims at contributing to the state of the art by introducing neuroevolutionary techniques to provide a clear and defined methodology for the extraction and selection of optimal impedance parameters for the given cost function. A method has been demonstrated in which criteria can be easily incorporated to evolve an optimal impedance controller.
>
> Finally, although the use of impedance controllers is widespread, they lack the higher level of intelligence needed to adapt to new or unknown situations. The mechanisms used by biological systems to control the arm have inspired a control system architecture that utilises previous knowledge to generate predictions about the nature of the objects which are to be manipulated. These predictions are based on the current context in which the robot is situated. Firstly, these predictions allow a pre-regulation of the compliance of the robotic manipulator. Secondly, they might also be used to perform real-time corrections in case of wrongly-predicted environments by comparing real and predicted sensory feedback.

The formulation of these three components into the form of an architecture is the main proposal and contribution of this thesis. The architecture is named *Predictive Context-Based Adaptive Compliance* (PCAC) due to the fact that the regulation and adaptation of the robot's compliance is based on predictions about the context in which the robot is immersed. The context prediction is useful for: (1) anticipating actions to be taken 'before' proceeding with the task as it has been shown in the 'Waiter Task' experiment, and (2) for applying real-time corrective measures 'during' the execution of the task in order to ensure a successful operation, as demonstrated with the 'Predictive Lifting' experiment. Both experiments have been reproduced using the robot 'Mr. SemProM', which is a dual-arm robot manipulator designed and built at DFKI. The robot has been especially useful for the 'Waiter Task' experiment, in which a dual-arm asymmetric task is performed.

8.2 Outlook

The following quote from the German writer Johann Wolfgang von Goethe (1749-1832) is a good introductory passage for this section:

> *'Properly speaking, such work is never finished; one must declare it so when, according to time and circumstances, one has done one's best.'*
> (*'So eine Arbeit wird eigentlich nie fertig; man muss sie für fertig erklären, wenn man nach Zeit und Umstand das Möglichste getan hat.'*)
> from *Italian Journey (Italienische Reise)*

Although Goethe was not referring to that topic in his work, this quote can successfully be applied to a research thesis work. A research work is finished after one has done one's best given the time and circumstances under which it was performed. This means that there is always room for improvements, for more tests, and for further and deeper investigations in each of the topics.

In this case, there are a number of open topics, especially concerning the enhanced possibilities that a context predictor could deliver.

First, the proposed architecture might be transformed into a real software architecture that captures the idea of the proposed one and combines both 'off-line' and 'on-line' blocks. The architecture would include a deliberative layer that would decide which actions should be taken and find optimal impedance controller parameters for newly identified environments. Similarly, a reactive layer would make use of the comparison between expected and real sensory feedback in order to react as a kind of reflex system of the arm.

Secondly, further investigation can be carried out in the area of Bayesian-based techniques for generating context predictions. The proposed Relevance Vector Machines (RVM) showed a good performance. Nevertheless, a deeper comparison and investigation of other probabilistic approaches is necessary, for example, one that easily can handle a high number of different contexts. Currently, the option of creating a multi-class RVM that grows with the number of contexts is a cumbersome task.

Additionally, the current approach labels the different contexts manually. A certain block of training data is manually associated with its corresponding label (empty, full, etc ...). It can be envisaged that the robot itself generates, labels, and dynamically adds new classes (contexts) to its repertoire.

In terms of application, the use in our robot AILA of the possibilities offered by the predictive compliance architecture is a short-term goal for this work. AILA already exhibits capabilities to perceive and label the objects and environments in which it is moving and/or performing its work. This feature can be definitely combined with the prediction and regulation of the arm compliance to the most suitable one for the current scenario.

Part II

APPENDIXES

Appendix A

ROBOTIC PLATFORMS

This appendix describes the robotic platforms used for the experimental results presented within the scope of this thesis. Basically, the first experiments were performed on a commercial 7-DoF Mitsubishi PA-10 robot and the last experiments on a dual-arm manipulator designed at our group that is composed of two commercial Schunk arms. Currently, the results are being implemented on the android AILA designed at DFKI but the results are not included in this thesis document.

A.1 Mitsubishi PA-10

An industrial, lightweight, redundant robot with 7 degrees of freedom and with the following properties:

The robot possesses seven joints. The joint configuration from the base of the robot is: R - P - R - P - R - P - R (R: Rotary joint, P: Pivoting joint).

Weight: 35 kg.

Payload: 10 kg.

Max. Speed: 1550 mm/s.

Drives: AC Motors.

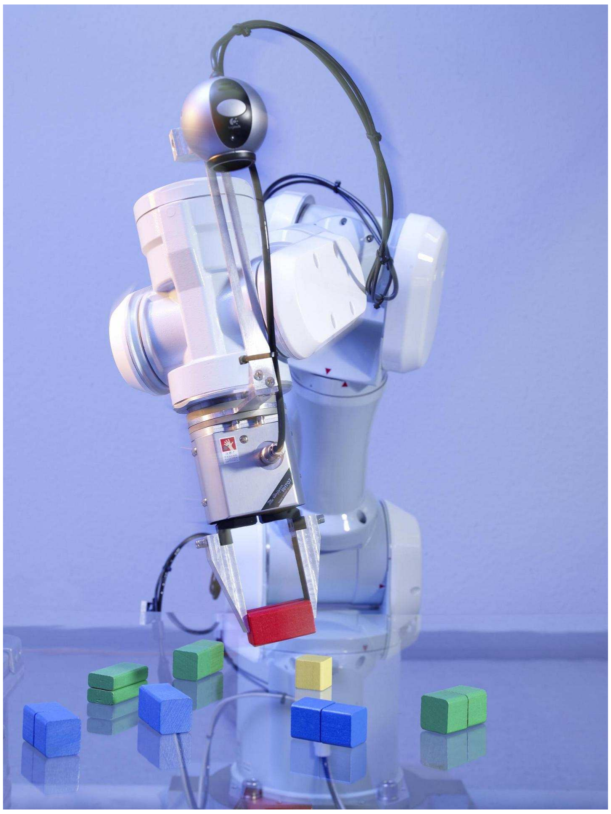


Figure A.1: Mitsubishi PA-10 robot at DFKI

The PA-10 has an open robot control architecture composed of different layers that allow an easy access to any of them.

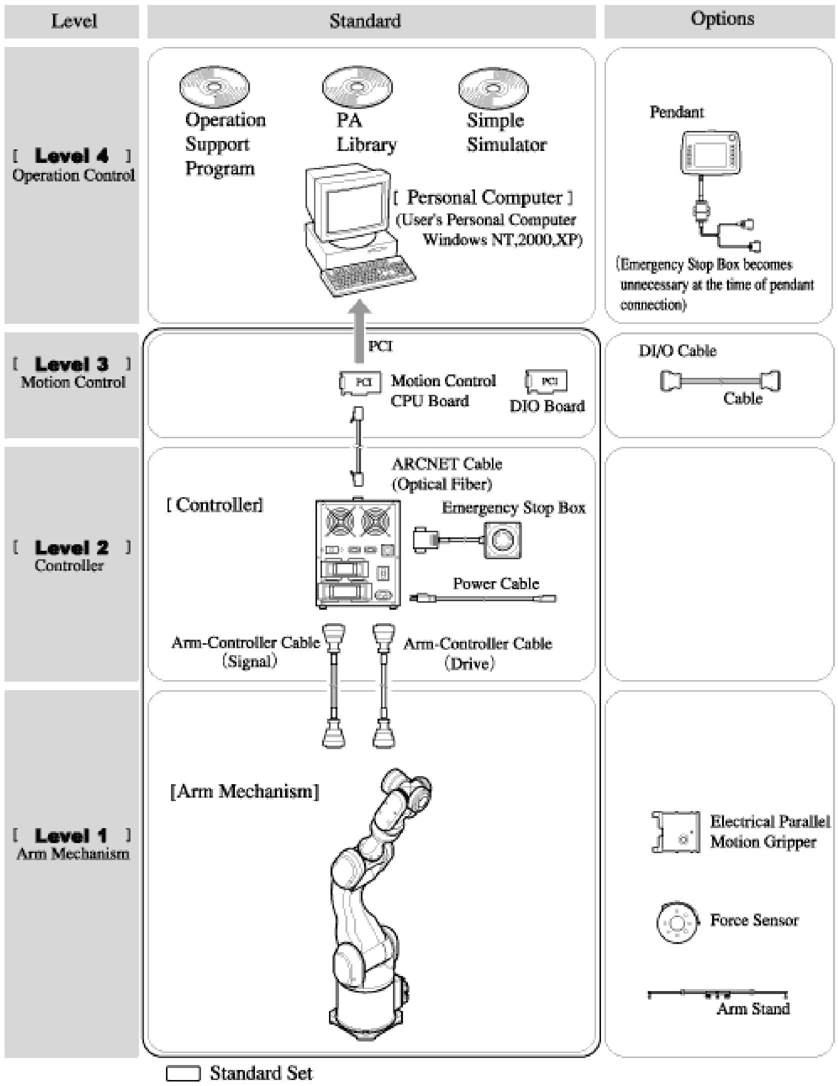


Figure A.2: Layer structure of the open architecture of the Mitsubishi PA-10

Hardware architecture The robot controller is composed of four layers as seen in Figure A.2. These layers are physically located in different devices, in order to provide easy access to them individually.

Layer 1: Mechanical Unit

It is composed of the mechanical robot arm.

Layer 2: Servo-Driver Unit

It is composed by the servo controlling the actuators of the PA-10. It can be accessed via ARCNET communication.

Layer 3: Movement Control Unit

It is composed by the ISA controller board. This board contains the kinematic and dynamic algorithms for the PA-10, and communicates in real-time with the servos in Layer 2. Similarly, since the board is connected to the user's PC (Layer 4), the board communicates with it by using a library provided with the robot (PAlib).

Layer 4: Control Unit

It is composed by the user's PC and by the C libraries that are used to communicate with the controller board.

Access to the different layers

Layer 1:

The access to Layer 1 is only through Layer 2 in order to guarantee that the PWM signals sent to the actuators are appropriate. These two levels communicate each other through a power cable and a signal cable and must be there in any system configuration.

Layer 2:

To access Layer 2, either the controller board on Layer 3 can be used or this can be replaced by an ARCNET communication board that sends the appropriate frames to the servos. This second option would require the development of the communication libraries as well as its implementation on a real-time system.

Layer 3:

To access Layer 3, the libraries provided by Mitsubishi have to be used in case of using the MHID6870 board. However, if a general communication board is used, a new set of libraries to control the robot need to be written that replace the libraries provided by Mitsubishi.

Layer 4:

Layer 4 is the user's PC that contains the PAlib provided by Mitsubishi or the one developed for a general communication.

In particular for this thesis, the control of the robot was done using a PC, which contained the MHID6870 controller board and making use of the PAlib.dll library provided by the manufacturer.

A.2 'Mr. SemProm' Robot

In the framework of the SemProM project funded by the German Federal Ministry of Education and Research (BMBF), a robotic dual-arm manipulator was designed using rotary joint modules from Schunk (Fig. A.3). The robot was deployed on a next-generation industrial automation facility, the $SmartFactory^{KL}$ [139], a factory whose components can be arbitrarily modified, and that autonomously reconfigures itself according to the current context. In this scenario, a robot should be able to deal with a mostly unknown, dynamically-changing environment as well as with variations on the properties and geometry of the goods to handle.

The requirements of the application scenario shaped the specifications of our robotic system. In terms of sensor equipment, the ability to react to enviromental changes is primarily provided by visual information. The robot needs to be able to visually scan the environment and recognize the current context. A stereo camera mounted on the head of the robot provides information about the objects in the environment as well as about their position. A high-speed camera on the robot's wrist guides the arm towards the object to grasp. On the other hand, the robot is required to be independent from the production line, i.e. it cannot rely on extra equipment mounted on the line. The reason stems mainly from the fact that a fault on the line can appear anywhere and it is not practicable to mount extra sensors/actuators all over it. That requirement led to the development of a dual-arm system that will combine the use of both arms to solve complex tasks. Figure A.3 shows our dual-arm robot manipulating objects from a simple $SmartFactory^{KL}$ module present at our laboratories.

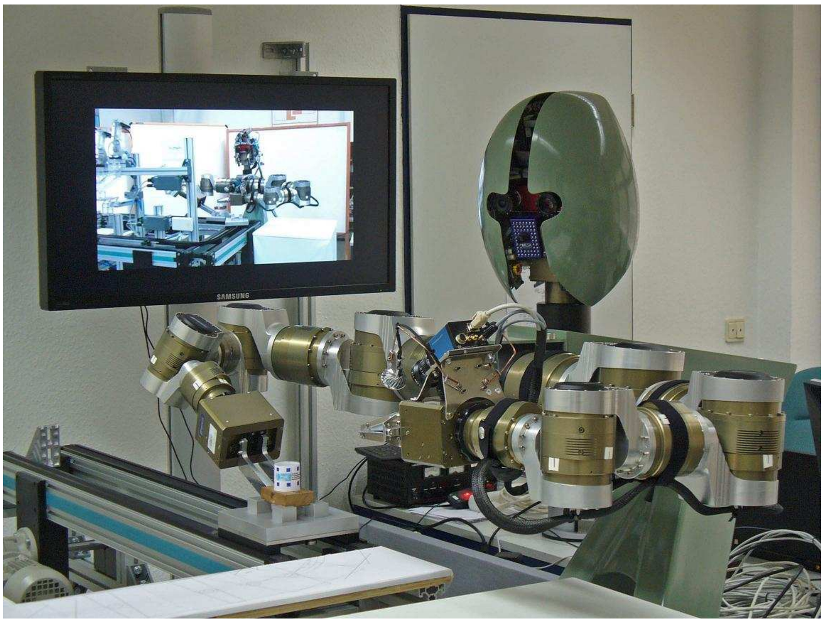


Figure A.3: 'Mr. SemProM', a dual-arm manipulator system built at DFKI using commercial modules from Schunk

Hardware This section aims at describing the hardware components used in our robotic platform, as well as giving a brief statement about its purpose for the system. Figure A.4 shows the main components of the system as well as the communication interfaces used between them.

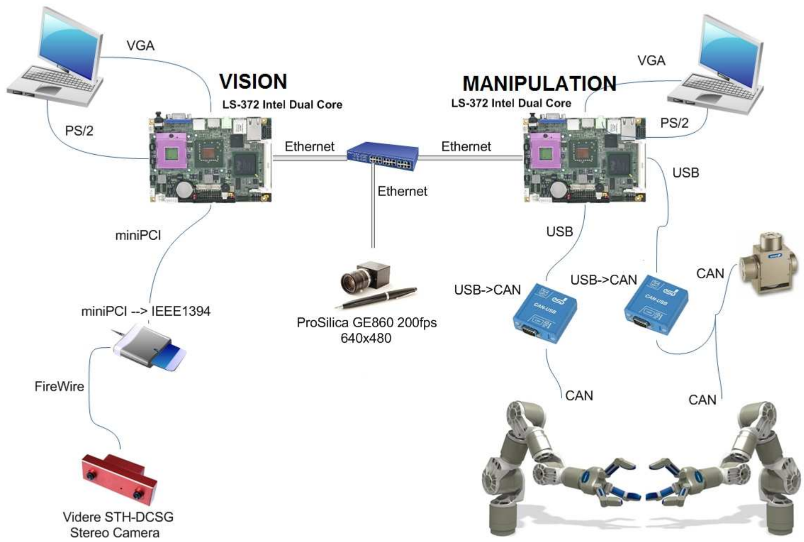


Figure A.4: 'Mr. SemProM' hardware components

Control PCs The *brain* of the robot system are two industrial 3.5" single-board computers (SBC) from COMMELL, model LS-372, with Intel Core 2 Duo Mobile T9300 processors at 2.5 GHz. Additionally, each board includes 1 GB RAM DDR2 memory, 1 Gigabit Ethernet interface, mini-PCI socket, two USB 2.0, two serial ports, and UltraATA33 IDE support for hard drives, among other interfaces.

> The *Manipulation Computer* is the main control board which controls the arms and requests, when necessary, camera information from the *Vision Computer* to guide the arm towards the objects.
>
> The *Vision Computer* is used for processing the data received from the two cameras: the stereo camera located on the head and the wrist camera. The former is used for object recognition, and the latter for visual servoing tasks.

Robot Arms The dual-arm system is based on modular joints from Schunk. Each arm is composed of seven modules, mixing four different module sizes (PRL120, PRL100, PRL080 and PRL060), with peak output torques ranging from 10 Nm to 372 Nm. The system uses two independent CAN bus lines, one line for controlling one arm plus the pan-tilt servo unit that controls the two degrees of the head, and the second line for controlling the second arm.

Cameras The robot is equipped with a set of cameras which provides the robot with valuable information about its dynamically-changing environment, thus allowing it to react according to that information. A stereo camera is used for the recognition of the objects to manipulate. This stereo camera is equipped with two different camera lenses: the left lens provides a wide angle view used to obtain a view of the whole scene, whereas the right lens provides a high-resolution and detailed view of a small area where the objects to manipulate are expected to be found. A second camera is located on the robot's wrist. This camera is used for visual servoing, i.e. to guide the arm towards the object by providing real-time information on the object's location. The camera is able to deliver 200 frames per second (fps) at a resolution of 640x480.

Control Software Figure A.5 shows the main processes running on both boards. Visual Servoing and Object Recognition modules running on the *Vision Computer* provide the *Manipulation Computer* with real-time information for guiding the arm towards an object, or to initiate proper actions according to the context. The Motion Generation module implements the CAN bus communication that interfaces with the arms, the direct and inverse kinematics algorithms, and controls the task execution as well as the communication between the two computers.

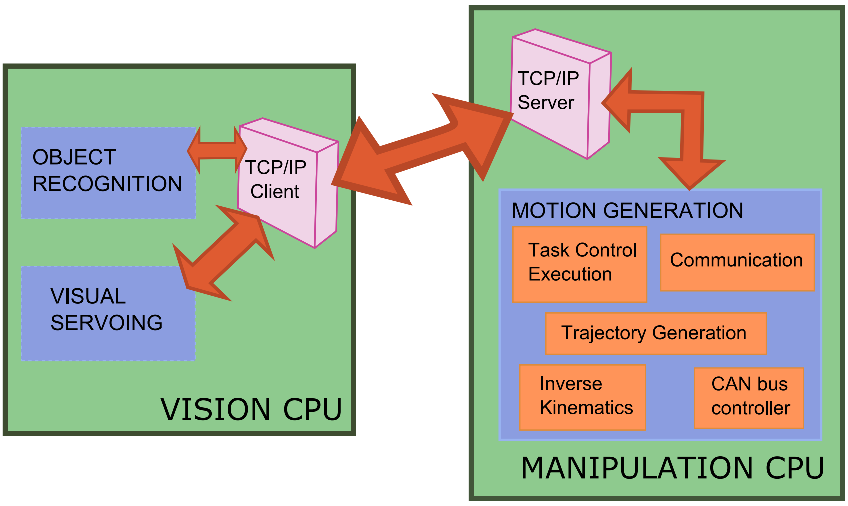


Figure A.5: 'Mr. SemProM' software architecture

A.3 AILA

The robot AILA (Fig. A.6), a mobile dual-arm robot system, was completely designed and built at DFKI as a research platform for investigating aspects of mobile manipulation. Its construction also occurred (as 'Mr. SemProM') in the framework of the project SemProM (Semantic Product Memory) funded by the German Federal Ministry of Education and Research (BMBF), as a second robotic platform that also includes mobility. AILA has 32 degrees of freedom, including 7-DOF arms, 4-DOF torso, 2-DOF head, and a mobile base equipped with six wheels, each of them with two degrees of freedom. The primary design goal was to achieve a lightweight arm construction with a payload-to-weight ratio greater than one. As a result, AILA's arms can lift 8kg and weigh 5.5kg, thus achieving a payload-to-weight ratio of 1.45.

Figure A.6: AILA, a mobile dual-arm manipulator built at DFKI

Hardware equipment The robot's hardware includes two Prosilica GC780C cameras that create a stereo system unit in the head which is mounted on a neck able to pan and tilt on an anthropomorphic path. A short-range Hokuyo URG Laser scanner in the chest and a Mesa SR-4000 3D Time-of-Flight (TOF) camera in the robot's stomach are combined for object and scene recognition, as well as for pose estimation. The mobile base carries two long-range Hokuyo UTM Laser scanners to provide a circumferential view required for mobility. Similarly to Mr. SemProM, AILA is equipped with three computers: two are 3,5-inches embedded PCs, one for motion control located in the head and one for navigation located in the mobile base. A mini-ITX board in combination with a dedicated graphics card for vision processing is located in the torso.

The communication network consists of five independent CAN-lines for controlling the two arms, the torso, and the wheel modules of the base. GigaEthernet communication is used to connect the head cameras, the three computers, and the outside world. Moreover, the wrists of the robot are equipped with six-axis force-torque sensors.

Electronics and Power Supply Within the mobile base, the current of the 48V battery is split to a 1 kW 24V power supply, a 0,5 kW 12V power supply, and direct supply of the wheel and arm drives. Fifteen different sub-circuits are separated by automatic circuit breakers. Within each sub-circuit all damageable components are protected with fast-reacting fuses according to their nominal power consumption. Voltages below 12V, that are needed for the chest laser-scanner and the mini ITX board, are converted by a 120W CPU power supply placed in the torso.

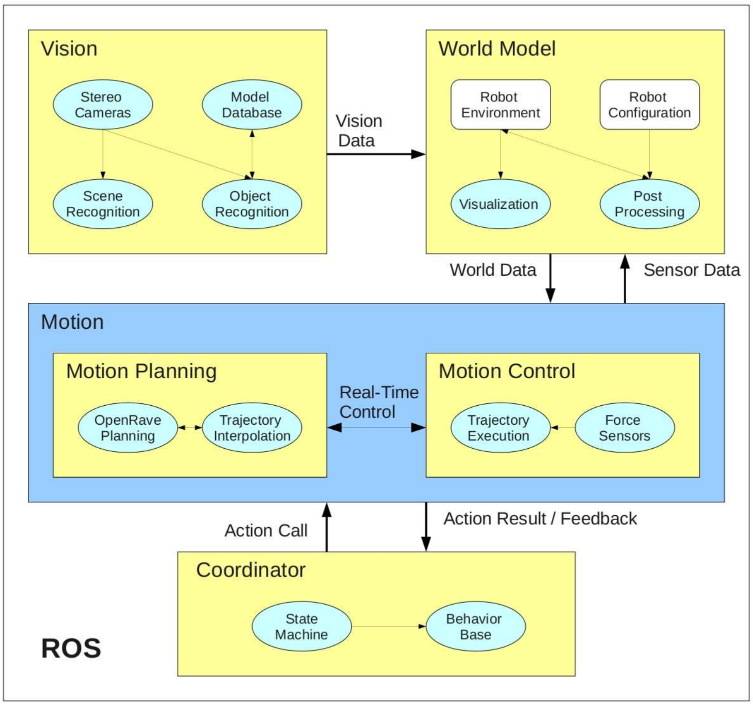


Figure A.7: Software architecture of the manipulation framework for the robot AILA

Control of the motor joints The torso joints as well as the vertical axes of the mobile platform use Faulhaber DC Motors which are controlled by an in-house developed power electronics board controlled by a STM32 microcontroller. The board is equipped with current, speed, and position sensors thus enabling local motor control. The high-level commands are transmitted from the embedded PCs via CAN messages. The arm joints and the horizontal axes of the mobile platform use brushless DC motors from Robodrive. A similar control approach to the one previously described has been used for these motors, which has already been successfully integrated in the Space-Climber robot [48][6]. In this case, the in-house developed motor electronics consists of a stack of three circular PCBs.

Control of the arms and torso The overall architecture of the manipulation software is shown in Figure A.7. The coordinator is implemented as a hierarchical state machine and controls the robot at task-level. It makes use of the behavior base, which represents a collection of basic robot functionalities (e.g. Plan Trajectory, Tilt Head, etc ...) that can be combined to achieve more complex behaviours. The behaviors themselves are implemented in different modules and can be triggered by action calls of the coordinator. Thereby, the motion planner entails the functionality for trajectory planning, whilst the motion controller contains routines for trajectory execution and other hardware-related features. The world model collects information about the robot environment (currently, mainly from the vision module) and the robot's current configuration. Moreover, upon request this module supplies this information to any other modules. For interprocess communication and as integrating software platform, we use the open-source framework ROS [104].

List of Figures

Bibliography

[1] Trends and challenges in industrial robot automation. *EURON White Paper, Fraunhofer IPA*, (DR-13.4), 2007.

[2] A. Albu-Schäffer and G. Hirzinger. Cartesian impedance control techniques for torque controlled light-weight robots. In *ICRA*, pages 657–663, 2002.

[3] D.E. Angelaki, A.G. Shaikh, A.M. Green, and J.D. Dickman. Neurons compute internal models of the physical laws of motion. *Nature*, 430:560–564, July 2004.

[4] T. Asfour, K. Berns, and R. Dillmann. The humanoid robot armar: Design and control. In *IEEE-RAS Int. Conf. On Humanoid Robots (HUMANOIDS 2000)*, pages 7–8, MIT, USA, Sept. 2000.

[5] Y. Bar-Shalom, X. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, New York, USA, 2001.

[6] S. Bartsch, T. Birnschein, F. Cordes, D. Kühn, P. Kampmann, J. Hilljegerdes, S. Planthaber, M. Römmermann, and F. Kirchner. SpaceClimber: Development of a six-legged climbing robot for space exploration. In *Proceedings for the Joint Conference of ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*. VDE Verlag GmbH, June 2010.

[7] M. Bazire and P. Brézillon. Understanding context before using it. In *CONTEXT*, pages 29–40, 2005.

[8] R. D. Beer, R. D. Quinn, H. J. Chiel, and R. E. Ritzmann. Biologically inspired approaches to robotics. *Communications of the ACM*, 40:30–38, 1997.

[9] C.M. Bennett and M.B. Miller. How reliable are the results from functional magnetic resonance imaging? *Annals of the New York Academy of Sciences*, 1191:133–155, 2010.

[10] R.G. Bonitz and T.C. Hsia. Robust internal-force based impedance control for coordinating manipulators-theory and experiments. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 622–628, 1996.

[11] R. A. Brooks. New approaches to robotics. *Science*, 253:1227–1232, 1991.

[12] R.A. Brooks, C. Breazeal, M. Marjanovic, B. Scassellati, and M.M. Williamson. The cog project: Building a humanoid robot. In *Lecture Notes in Computer Science*, pages 52–87. Springer-Verlag, 1999.

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

[14] D. Cattaert, A. Semjen, and J. J. Summers. Simulating a neural cross-talk model for between-hand interference during bimanual circle drawing. *Biological Cybernetics*, 81(4):343–358, 1999.

[15] D. Chakarov. Study of the antagonistic stiffness of parallel manipulators with actuation redundancy. *Mechanism and Machine Theory*, 39:538–601, 2004.

[16] C.C. Cheah and D. Wang. Learning impedance control for robotic manipulators. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 2150–2155, May 1995.

[17] S. Chiaverini and L. Sciavicco. The parallel approach to force/position control of robotic manipulators. *IEEE Transactions on Robotics and Automation*, 9:361–373, 1993.

[18] S. Chiaverini, B. Siciliano, and L. Villani. A survey of robot interaction control schemes with experimental comparison. *Mechatronics, IEEE/ASME Transactions on*, 4(3):273–285, Sep 1999.

[19] M. Cohen and T. Flash. Learning impedance parameters for robot control using an associative search network. *IEEE Transactions on Robotics and Automation*, 7(3):382–390, Jun 1991.

[20] J.V. Cohn, P. DiZio, and J.R. Lackner. Reaching During Virtual Rotation: Context Specific Compensations for Expected Coriolis Forces. *J Neurophysiol*, 83(6):3230–3240, 2000.

[21] R. Colbaugh, H. Seraji, and K. Glass. Direct adaptive impedance control of robot manipulators. *Journal of Robotic Sytems*, 10(2):217–248, 1993.

[22] M. Darainy, N. Malfait, P.L. Gribble, F. Towhidkhah, and D.J. Ostry. Learning to Control Arm Stiffness Under Static Conditions. *Journal of Neurophysiology*, 92(6):3344–3350, 2004.

[23] P.R. Davidson and D.M. Wolpert. Widespread access to predictive models in the motor system: a short review. *Journal of Neural Engineering*, 2(3):313–319, 2005.

[24] J. de Gea. The role of prediction in compliance adaptation. In *Robotics: Science and Systems, Workshop on Strategies and Evaluation for Mobile Manipulation in Household Environments (RSS2010)*, Zaragoza, Spain, 2010.

[25] J. de Gea, Y. Kassahun, and F. Kirchner. *Factory Automation*, chapter Control of Robot Interaction Forces Using Evolutionary Techniques, pages 445–462. In-Tech, 2009.

[26] J. de Gea, Y. Kassahun, and F. Kirchner. On evolving a robust force-tracking neural network-based impedance controller. In *In 40th International Symposium on Robotics (ISR'09)*, pages 127–132, 2009.

[27] J. de Gea and F. Kirchner. Contact impedance adaptation via environment identification. In *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE08)*, pages 1365–1370, Cambridge, UK, June 2008.

[28] J. de Gea and F. Kirchner. Modelling and simulation of robot arm interaction forces using impedance control. In *Proceedings of the 17th World Congress The International Federation of Automatic Control (IFAC)*, pages 15589–15594, Seoul, Korea, July 6-11 2008.

[29] J. de Gea and F. Kirchner. Using neuroevolution for optimal impedance control. In *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA-2008)*, pages 1063–1066, September 15-18 2008.

[30] J. de Gea, J. Lemburg, T.M. Roehr, M. Wirkus, I. Gurov, and F. Kirchner. Design and control of an intelligent dual-arm manipulator for fault-recovery in a production scenario. In *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA-2009)*, pages 1583–1587, September 22-26 2009.

[31] J. De Schutter, H. Bruyninckx, W-H. Zhu, and M.W. Spong. Force control: A bird's eye view. In *In B. Siciliano (Ed.), Control Problems in Robotics and Automation: Future Directions*, pages 1–17. Springer Verlag, 1997.

[32] M. Desmurget, D. Pelisson, Y. Rossetti, and C. Prablanc. From eye to hand: Planning goal-directed movements. *Neuroscience and Biobehavioral Reviews*, 22(6):761–788, 1998.

[33] A. K. Dey. Understanding and using context. *Personal Ubiquitous Comput.*, 5(1):4–7, 2001.

[34] N. Diolaiti, C. Melchiorri, and S. Stramigioli. Contact impedance estimation for robotic systemss. In *IEEE Transactions on Robotics*, pages 925–935, October 2005.

[35] M. Duffose, M. Hugon, and J. Massion. Postural forearm changes induced by predictable in time or voluntary triggered unloading in man. *Experimental Brain Research*, 60:330–334, 1985.

[36] D. Erol, V. Mallapragada, and N. Sarkar. Adaptable force control in robotic rehabilitation. In *IEEE Int. Workshop on Robots and Human Interactive Communication*, pages 649–654, 2005.

[37] E.N. Eskandar and J.A. Assad. Dissociation of visual, motor and predictive signals in parietal cortex during visual guidance. *Nature Neuroscience*, 2:88–93, 1999.

[38] R.S. Fishman. The Nobel Prize of 1906. *Archives of Ophthalmology*, 125(5):690–694, 2007.

[39] P.M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954.

[40] R.J. Flanagan, P. Vetter, R.S. Johansson, and D.M. Wolpert. Prediction precedes control in motor learning. *Current Biology*, 13(2):146–150, January 2003.

[41] T. Flash. The control of hand equilibrium trajectories in multi-joint arm movements. *Biological Cybernetics*, 57:257–274, 1987.

[42] T. Flash and N. Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of neuroscience*, 5:1688–1703, 1985.

[43] S.F. Giszter, F.A. Mussa-Ivaldi, and E. Bizzi. Convergent force fields organized in the frog's spinal cord. *Journal of Neuroscience*, 13(2):467–491, 1993.

[44] K. Gurney, T.J. Prescott, J.R. Wickens, and P. Redgrave. Computational models of the basal ganglia: from robots to membranes. *Trends in Neurosciences*, 27(8):453 – 459, 2004.

[45] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[46] C.M. Harris and D.M. Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394:780–784, 1998.

[47] M. Hersch and A. G. Billard. Reaching with multi-referential dynamical systems. *Autonomous Robots*, 25(1-2):71–83, 2008.

[48] J. Hilljegerdes, P. Kampmann, S. Bosse, and F. Kirchner. Development of an intelligent joint actuator prototype for climbing and walking robots. In *International Conference on Climbing and Walking Robots (CLAWAR-09)*, pages 942–949, 2009.

[49] M.R. Hinder and T.E. Milner. The case for an internal dynamics model versus equilibrium point control in human movement. *The Journal of Physiology*, 549:953–963, June 2003.

[50] N. Hogan. Impedance control-an approach to manipulation, part I- theory, part II- implementation, part III- application. *Journal of Dynamics Systems, Measurement, and Control-Transactions of the ASME*, 107(1):1–24, 1985.

[51] N. Hogan. The mechanics of multi-joint posture and movement control. *Biological Cybernetics*, 52(5):315–331, 1985.

[52] N. Hogan. Stable execution of contact tasks using impedance control. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1047–1054, 1987.

[53] N. Hogan. On the stability of manipulators performing contact tasks. *IEEE Journal of Robotics and Automation*, 4:677–686, 1988.

[54] J. Holland. *Adaptation in natural and artificial systems.* University of Michigan Press, 1975.

[55] J.M. Hollerbach and T. Flash. Dynamic interactions between limb segments during planar arm movements. *Biological Cybernetics*, 44(1):67–77, May 1982.

[56] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[57] M. Hugon, J. Massion, and M. Wiesendanger. Anticipatory postural changes induced by active unloading and comparison with passive unloading in man. *Pfluegers Archives*, 393:292–296, 1982.

[58] M. Ito. Neurophysiological aspects of the cerebellar motor control system. *Internation Journal of Neurology*, 7:162–176, 1970.

[59] R.S. Johansson. Sensory input and control of grip. *Sensory Guidance of Movement, Novartis Foundation Symposium*, 218:45–59, 1998.

[60] E.G. Jones and L.M. Mendell. Assessing the decade of the brain. *Science*, 284(5415):739, 1999.

[61] M.I. Jordan and D.E. Rumelhart. Forward models: Supervised learning with a distal teacher, 1992.

[62] S. Jung and T.C. Hsia. Neural network impedance force control of robot manipulator. In *IEEE Transactions on Industrial Electronics*, pages 451–461, Jun 1998.

[63] S. Jung and T.C. Hsia. Robust neural force control scheme under uncertainties in robot dynamics and unknown environment. *IEEE Transactions on Industrial Electronics*, 47(2):403–412, 2000.

[64] S. Jung, T.C. Hsia, and R.G. Bonitz. Force Tracking Impedance Control for Robot Manipulators with an Unknown Environment: Theory, Simulation, and Experiment. *The International Journal of Robotics Research*, 20(9):765–774, 2001.

[65] S. Jung, S.B. Yim, and T.C. Hsia. Experimental studies of neural network impedance force control for robot manipulators. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2001*, volume 4, pages 3453–3458, 2001.

[66] Paul R. Kalata. Alpha-beta target tracking systems: A survey. In *American Control Conference*, pages 832–836, 1992.

[67] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, Series D:35–45, 1960.

[68] M. Kawato. Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, 9(6):718–727, 1999.

[69] M. Kawato and H. Gomi. A computational model of four regions of the cerebellum based on feedback-error learning. *Biol. Cybernetics*, 68:95–103, 1999.

[70] H. Kazerooni. Contact instability of the direct drive robot when contrained by a rigid environment. *IEEE Transactions on Automation and Control*, 35:710–714, 1990.

[71] J.A.S. Kelso, D.L. Southard, and D. Goodman. On the coordination of two-handed movements. *Journal of Experimental Psychology: Human Perception and Performance*, 2:229–238, 1979.

[72] O. Khatib and B. Roth. New robot mechanisms for new robot capabilities. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 44–49, 1991.

[73] J. Kim and M.N. Shadlen. Neural correlates of a decision in the dorsolateral prefrontal cortex of the macaque. *Nature Neuroscience*, 2:176–185, 1999.

[74] K.P. Koerding, S. Ku, and D.M. Wolpert. Bayesian integration in force estimation. *Journal of Neurophysiology*, 92:3161–3165, 2004.

[75] N.I. Krouchev and J.F. Kalaska. Context-dependent anticipation of different task dynamics: Rapid recall of appropriate motor skills using visual cues. *J Neurophysiol*, 89(2):1165–1175, February 2003.

[76] A.D. Kuo. An optimal control model for analyzing human postural balance. *IEEE transactions on bio-medical engineering*, 42(1):87–101, January 1995.

[77] F. Lacquaniti, M. Carrozzo, and N. A. Borghese. Time-varying mechanical behavior of multijointed arm in man. *Journal of Neurophysiology*, 69(5):1443–1464, 1993.

[78] F. Lacquaniti, J.F. Soechting, and S.A. Terzuolo. Path constraints on point-to-point arm movements in three-dimensional space. *Neuroscience*, 17(2):313–324, 1986.

[79] F. Lacquaniti, C. Terzuolo, and P. Viviani. The law relating the kinematic and figural aspects of drawing movements. *Acta Psychologica*, 54(1-3):115–130, 1983.

[80] H. Lalazar and E. Vaadia. Neural basis of sensorimotor learning: modifying internal models. *Current opinion in neurobiology*, 18(6):573–81, 2008.

[81] C. Laschi, G. Asuni, E. Guglielmelli, G. Teti, R. Johansson, H. Konosu, Z. Wasik, M.C. Carrozza, and P. Dario. A bio-inspired predictive sensory-motor coordination scheme for robot reaching and preshaping. *Autonomous Robots*, 25(1-2):85–101, 2008.

[82] S. Laureys, M. Boly, and G. Tononi. *Functional neuroimaging, in book The Neurology of Consciousness*. S. Laureys, G. Tononi (eds), Academic Press, New York, USA, 2008.

[83] T. Lefebvre, J. Xiao, H. Bruyninckx, and G. De Gersem. Active compliant motion: a survey. *Advanced Robotics*, 19(5):479–499, 2005.

[84] F.L. Lewis, D.M. Dawson, and C.T. Abdallah. *Robot manipulator control: theory and practice*. 2004.

[85] S. Lin and H. Tsai. Impedance control with on-line neural network compensator for dual-arm robots. *Journal of Intelligent Robotics Systems*, 18(1):87–104, 1997.

[86] L.J. Love and W.J. Book. Environment estimation for enhanced impedance control. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA 1995)*, pages 1854–1859, 1995.

[87] A.A. Lu, Z. Goldenberg. Robust impedance control and force regulation: theory and experiments. *Internation Journal of Robotics Research*, 14:225–254, 1995.

[88] W.-S. Lu and Q.-H. Meng. Impedance control with adaptation for robotic manipulations. In *IEEE Transactions on Robotics and Automation*, pages 408 – 415, Jun 1991.

[89] D. Matko, R. Kamnik, and T. Badj. Adaptive impedance force control of an industrial manipulator. In *Proc. IEEE Int. Symposium on Industrial Electronics*, pages 129–133, 1999.

[90] W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. 5:115–133, 1943.

[91] B. Mehta and S. Schaal. Forward Models in Visuomotor Control. *J Neurophysiol*, 88(2):942–953, 2002.

[92] R. C. Miall, D. J. Weir, D. M. Wolpert, and J. F. Stein. Is the cerebellum a smith predictor? *Journal of Motor Behavior*, 25(3):203–216, 1993.

[93] T.E. Milner and D.W. Franklin. Impedance control and internal model use during the initial stage of adaptation to novel dynamics in humans. *Journal of Physiology*, 567:651–664, 2005.

[94] S.A.A. Moosavian and E. Papadopoulos. Multiple impedance control for object manipulation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 461–466, 1998.

[95] P. Morasso. Spatial control of arm movements. *Experimental Brain Research*, 42(2):223–227, 2000.

[96] W. L. Nelson. Physical principles for economies of skilled movements. *Biological Cybernetics*, 46(2):135–147, February 1983.

[97] I. Nevins and D.E. Whitney. The force vector assembler concept. In *Proceedings of first CSIM-IFToMM Symposium on Theory and Practice of Robots and Manipulators*, 1973.

[98] C. Ott, O. Eiberger, W. Friedl, B. Bauml, U. Hillenbrand, C. Borst, A. Albu-Schaffer, B. Brunner, H. Hirschmuller, S. Kielhofer, R. Konietschke, M. Suppa, T. Wimbock, F. Zacharias, and G. Hirzinger. A humanoid two-arm system for dexterous manipulation. In *6th IEEE-RAS International Conference on Humanoid Robots*, pages 276–283, Dec. 2006.

[99] C. Perez-Vidal, L. Gracia, N. Garcia, and E. Cervera. Visual control of robots with delayed images. *Advanced Robotics*, 23:725–745, 2009.

[100] A. Polit and E. Bizzi. Processes controlling arm movements in monkeys. *Science*, 201(4362):1235–1237, 1978.

[101] J. Pratt, P. Dilworth, and G. Pratt. Virtual model control of a bipedal walking robot. In *IEEE Conference on Robotics and Automation*, pages 193–198, 1997.

[102] J. Pratt, B. Krupp, and C. Morse. Series elastic actuators for high fidelity force control. *Industrial Robot: An Internation Journal*, 29:234–241, 2002.

[103] J.E. Pratt and G.A. Pratt. Exploiting natural dynamics in the control of a planar bipedal walking robot. In *In Proceedings of the 36th Annual Allerton Conference on Communication, Control and Computing*, pages 739–748, 1998.

[104] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source robot operating system. In *In Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[105] Osu R., Kamimura N., Iwasaki H., Nakano E., Harris C.M., Wada Y., and Kawato M. Optimal impedance control for task achievement in the presence of signal-dependent noise. *Journal of Neurophysiology*, 92:1199–1215, 2004.

[106] M.H. Raibert and J.J. Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 102:126–133, 1981.

[107] I. Rechenberg. *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog, Stuttgart, 1973.

[108] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. pages 318–362, 1986.

[109] J. Kenneth Salisbury. Active stiffness control of a manipulator in cartesian coordinates. volume 19, pages 95–100, Dec. 1980.

[110] Hans-Paul Paul Schwefel. *Evolution and Optimum Seeking: The Sixth Generation*. John Wiley & Sons, Inc., New York, NY, USA, 1993.

[111] H. Seraji and R. Colbaugh. Adaptive force-based impedance control. In *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1537 – 1544, Jul 1993.

[112] H. Seraji and R. Colbaugh. Force tracking in impedance control. *The International Journal of Robotics Research*, 16:97–117, 1997.

[113] R. Shadmehr and O.A. Mussa-ivaldi. Adaptive representation of dynamics during learning of a motor task. *Journal of Neuroscience*, 14:3208–3224, 1994.

[114] M. Shidara, K. Kawano, H. Gomi, and M. Kawato. Inverse-dynamics model eye movement control by purkinje cells in the cerebellum. *Nature*, 365:50–52, 1993.

[115] B. Siciliano and O. Khatib, editors. *Springer Handbook of Robotics*. Springer, Berlin, Heidelberg, 2008.

[116] B. Siciliano and L. Villani, editors. *Robot Force Control*. Kluwer Academic Publishers, Boston, 1999.

[117] S.K. Singh and D.O. Popa. An analysis of some fundamental problems in adaptive control of force and impedance behavior, theory and experiments. *IEEE Transactions on Robotics and Automation*, 11:223–228, 1995.

[118] P. Slaets, J. Rutgeerts, K. Gadeyne, T. Lefebvre, H. Bruyninckx, and J. De Schutter. Construction of a geometric 3-d model from sensor measurements collected during compliant motion. In *Proceedings of the International Symposium on Experimental Robotics*, 2004.

[119] M.A. Smith and R. Shadmehr. Intact ability to learn internal models of arm dynamics in huntington's disease but not cerebellar degeneration. *Journal of Neurophysiology*, 93:2809–2821, 2005.

[120] D. Surdilovic. Contact stability issues in position based impedance control: Theory and experiments. *Proc. IEEE ICRA96*, pages 1675–1680, 1996.

[121] S.P. Swinnen and N. Wenderoth. Two hands, one brain: Cognitive neuroscience of bimanual skill. *Trends in Cognitive Sciences*, 8:18–25, 2004.

[122] M.E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[123] E. Todorov. Optimality principles in sensorimotor control. *Nature neuroscience*, 7(9):907–915, September 2004.

[124] R. M. Turner. Context-mediated behavior for intelligent agents. *International Journal of Human-Computer Studies*, 48:307–330, 1997.

[125] Y. Uno, M. Kawato, and R. Suzuki. Formation and control of optimal trajectory in human multijoint arm movement. minimum torque-change model. *Biological cybernetics*, 61(2):89–101, 1989.

[126] M. Vukobratovic, D. Surdilovic, and Y. Ekalo. *Dynamics and Robust Control of Robot-environment Interaction.* World Scientific Publishing Company, River Edge, NJ, USA, 2009.

[127] E. Vul, C. Harris, P. Winkielman, and H. Pashler. Puzzlingly high correlations in fmri studies of emotion, personality, and social cognition. *Perspectives on Psychological Science*, 4(3):274–290, 2009.

[128] B. Webb. Can robots make good models of biological behaviour? *Behavioral and brain sciences*, 24:1033–1050, 2001.

[129] B. Webb. Neural mechanisms for prediction: do insects have forward models? *Trends in Neurosciences*, 27:278–282, 2004.

[130] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical Report TR95-041, University of North Carolina at Chapel Hill, Department of Computer Science, USA, 1995.

[131] D.E. Whitney. Force feedback control of manipulator fine motions. *ASME Journal of Dynamic Systems, Measurement, and Control*, 99:91–97, 1977.

[132] D.E. Whitney. Historical perspective and the state-of-the art in robot force control. *International Journal of Robotics Research*, 6:3–14, 1987.

[133] D.E. Whitney and J.L. Nevins. What is the remote center compliance and what can it do? In *Proceedings of 9th Int. Symposium and Exposition on Industrial Robots (ISIR)*, 1979.

[134] A. G. Witney. Internal models for bi-manual tasks. *Human Movement Science*, 23:747–770, 2004.

[135] D.M. Wolpert, K. Doya, and M. Kawato. A unifying computational framework for motor control and social interaction. *Philosophical Transactions of the Royal Society of London Series B-Biological Sciences*, 358:593–602, 2003.

[136] D.M. Wolpert and Z. Ghahramani. Computational principles of movement neuroscience. *Nature Neuroscience Supplement*, 3, 2000.

[137] D.M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11(7–8):1317–1329, 1998.

[138] T. Yabuta, T. Yamada, T. Tsujimura, and H. Sakata. Force control of servomechanism using adaptive control. *IEEE Transactions on Robotics and Automation*, 4:223–228, 1988.

[139] D. Zuehlke. SmartFactory: from vision to reality in factory technologies. In *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, July 2008.