POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione

Corso di Laurea Magistrale in
Biomedical Engineering

A SEMI-AUTONOMOUS TELEOPERATION FRAMEWORK FOR IMPROVED ADAPTABILITY TO EXTERNAL INTERACTIONS

Relatore: Prof. Elena De Momi
Dr. Arash Ajoudani

Tesi di Laurea di:

Adriano Scibilia Matr. 854154

Achnowledgements

I would like to thank the NearLab, especially professor Elena De Momi for giving me this wonderful opportunity in the first place. Furthermore, i would like to show my gratitude to all the colleagues of Human Robot Interfaces and Physical Interaction lab, in particular to Luka Peterel and Marco Laghi, and in general of ADVR department of IIT, who always helped me in my path through this field and taught me a lot.
Finally, thanks to Arash Ajoudani, that has been a guide for all this time.

Adriano

a te,
ovunque tu sia,
e qualunque percorso di vita tu abbia intrapreso.

Contents

List of Figures

List of Tables

Sommario

Questo lavoro presenta un metodo di controllo semi-autonomo, basato su un interfaccia uomo-robot per sistemi di teleoperazione, la Tele-impedenza, nel quale il robot ha la capacità di adattarsi in maniera dinamica quando interagisce fisicamente con l'ambiente. Il metodo proposto si basa su un sistema di controllo in feed-forward, in cui i riferimenti di posizione sono forniti dall'operatore umano, mentre localmente, il controllo adattativo qui proposto regola gli altri aspetti dell'interazione del robot con l'ambiente circostante. Questo metodo utilizza il feedback dei sensori di forza del robot per riconoscere e gestire autonomamanente aspetti dell'interazione con l'ambiente esterno a cui l' uomo non deve prestare attenzione, come la compensazione del peso di tutti gli strumenti usati durante l'esecuzione di un operazione, senza conoscerne alcuna informazione a priori, e mantenendo un comportamento sicuro con l'ambiente esterno sconosciuto, assicurato anche dall'uso di una bassa rigidezza. Grazie a questa continua compensazione, lo slave (ovvero il robot) riduce il disallineamento con l'operatore umano (master), durante l'esecuzione di complesse attività interattive, senza la necessità di usare un feedback di forza. Il controller locale del robot assicura inoltre la stabilità dell'intero sistema, basandosi sulla teoria della passività, tramite l'uso dell' energy-tank, per essere in grado di gestire contatti imprevisti o l'improvvisa perdita degli oggetti che stava manipolando, senza che questo causi movimenti inaspettati e non sicuri con il quale si potrebbero potenzialmente causare ferite o infortuni nel caso si stia operando in cooperazione con altri esseri umani.

La validità del nostro approccio è provata dai risultati sperimentali ottenuti eseguendo un task di manipolazione che richiede sia la capacità di manipolare gli ogetti, che un'opportuna interazione con l'ambiente, usando il braccio robotico Kuka LWR.

Abstract

This work presents a new semi-autonomous Teleoperation system, based on Teleimpedance (Ajoudani et al., 2012), where the robot has an adaptive capacity for physical interaction with the environment. In many application in which a robot is teleoperated, the surrounding area where it is acting is not known, therefore it is crucial in such situations to have a smooth motion and a non-rigid behavior when a collision with something occurs, in other words, it is important to be compliant.

While humans are able to adapt their endpoint force and viscoelasticity to compensate for the environment forces and instability (Tee et al., 2010), robots do not have this capability, being, in general, not conceived to deal with unstable interactions. The concept of active control of a manipulator's interactive behavior is formally treated as an aspect of impedance control (Hogan, 1984). The impedance control provides a compliant behavior during the interaction and regulates the dynamic response of the robot end-effector to interaction forces by establishing a suitable virtual mass-spring-damper system on the end-effector. Excessive contact force between the manipulator and the environment should be prevented.

In order to do that, humans control the force exerted on an object by adapting their arm stiffness, in a similar way the robot should be able to change the stiffness of its arm while performing an interaction task, the idea to transfer the human stiffness references to the robot has led to the development of the "Teleimpedance" by (Ajoudani et al., 2012). The work here presented, has the purpose to extend the Teleimpedance framework providing additional capabilities of the robot of autonomously manage interaction with external environment. The main considered application was the robotic surgery, in this scenario, the focus has to be on the reproduction of surgeon-like behavior while performing a surgical operation.
In (Ferraguti et al., 2013), a puncturing task (i.e. act of penetrating a biological tissue with a needle) was considered as example. Since different human tissues present different impedance characteristics (Hernandez et al., 2001), the surgeon has to adapt the stiffness of his arm while performing the puncturing and inserting the needle, depending on the physical characteristics of the tissue or organ the needle is penetrating (Barbé et al., 2006). In order to imitate the surgeon's behavior, an impedance control with variable target stiffness has been proposed.

On the other hand, if considering different situations apart from the puncturing task, a varying stiffness term can be quite dangerous, being the robot still unable to react with the same ability of human beings to unexpected situations. Consider, for example, the case in which the surgeon has to lift an organ or a piece of tissue, in order to compensate its weight, the stiffness will be increased, and if the grasped organ is accidentally lost, after a contact with an obstacle like another

organ in the surrounding area this will cause a large and dangerous motion and can cause damages. Therefore, an alternative approach is to command the lowest possible stiffness, in order to interact in a compliant way with any obstacle in the surrounding area, and leave the object compensation responsibility to the local robot controller: thus in this work an hybrid force/impedance controller with commanded low stiffness was preferred to manage safely this situation on the robot side.

From the human side, the main objective in robotic tele-operation is to enable the human operator to effectively and intuitively control the robot's operations to accomplish a remote task, the degree with whom the human operator can intuitively interact with the external environment trough the robot is defined as "transparency" of the system. To improve the transparency of a teleoeration framework, approaches using a force-feedback from the robot has been used in the Teleimpedance (Laghi et al., 2017). Also if this solution is useful for improving the degree of accuracy with which the operator interact with the external world, it still have some drawbacks: in order to send these huge amount of informations between human operator and robot, a four channel teleoperation architecture (described in chapter 1) must be used, therefore, an eventual presence of delays in the communication channel can cause instable behavior.

Also, in order to provide force feedback to the master side of the teleoperation system, an additional device should be used, causing discomfort to the human operator. Therefore, in situations in which a good visual feedback can be provided to the operator, such as the surgery case mentioned above, force feedback is not necessary and should be avoided, while a feed-forward approach can improve system robustness to communication delays. For this reason this solution was used in this work, in which the lack of a feedback to the user is compensated by the improvement of the capability of the robot to autonomously manage most important aspects of the interaction with the external world mentioned above.

The proposed method is based on a feed-forward control scheme, where the position reference in the world frame coordinates, is provided by the human operator, while the physical interactions with the environment are performed semi-autonomously though the proposed adaptive local control method. This method uses the feedback from robot's force sensors to recognize and compensate aspects of the interaction to which the human does not need to pay attention to: the dynamic compensation of the weight of any used tools, without any a priori information, while maintaining a safe and compliant behaviour with the unknown external environment. Thanks to this on-line compensation, the slave reduces the misalignment with the human operator, while performing complex interactive tasks.

While the robot execute these tasks in cooperation with humans, unexpected events like a collision with an obstacle or a sudden loss of the manipulated tool can occur, without a proper additional controlling action, this can cause dangerous motion and potentially hurt or injury someone, therefore safety should be a parameter of primary importance in such teleoperation framework. To prevent the robot from doing unsafe actions, our local controller also includes a stability controlling part based on the energy tank passivity paradigm (that will be describe in detail in chapter 4).

The energy-tank passivity controller dynamics were tested with a simulation of a contact loss situation using Gazebo and the XBotCore software architecture, and

then real experiments of object grasping with successive contact loss were provided. Once proved the validity of the stability control part, the whole teleoperation system was tested in an experimental setup involving both object manipulation and environment interaction, using Kuka LWR arm. All the results were measured in terms of position and force errors with the aim of proving how they were reduced by the proposed controller, showing (as expected) its best performances with low commanded stiffness. **Keywords:** Tele-impedance, Safe interactions, Energy tank,

Introduction

The role of robotics in modern industry has increased for decades, from the mining to the automotive sector, robots are performing tasks that humans either don't want or can't do, and will doubtless continue to grow as quality and productivity demands become more stringent. In the past, these machines were generally large, with inflexible programming, and physically separated, often with a cage, from their human operators for safety reasons. Alternately, they were meant to operate in remote or dangerous locations where human presence is inconvenient or inadvisable. These kind of robots were able to operate with a complete knowledge of the environment and a detailed planning of the task it had to do. Models were used to describe every aspect of the motions and of the eventual expected interactions.

Most recently developed technologies have led robots to be used in different fields beside the previously described heavy-industrial environments. As the most recent applications in healthcare and surgery may suggest, robots are being increasingly present in our homes, in hospitals or disaster scenarios, helping people and constantly interacting with them in ways that can't be planned, while being in an unknown (and potentially changing) environment. Therefore, robots are expected to smartly adapt itself to all these uncertainties during a task execution without loosing performance, in terms of precision and speed, and safety when interacting with the external environment. For these reasons, required tasks started to became more and more complex, and autonomous control is not able to provide all this capability at the moment, despite all the achievements done in this field. Artificial intelligence and machine-learning, for example, which had a steady improvement in the last years, has not developed enough yet to allow robot's decision making to be comparable to the human capability to adapt to unpredicted situations, and to enhance complex tasks.

To overcome all these issues, the best solution is to exploit human superior cognitive abilities, and let to the robot all the physical effort. There is a pressing need to engineer systems that can have a close interaction with humans, gather and integrate data on their intention, and respond to the operator's needs in a smart and collaborative way. Teleoperation systems meet this requirement, and their usage has spread in the last decades for a wide range of applications, from aerospace industry to surgery, robots are able to do tasks with high efficiency and precision while following operator's references, sometimes also outperforming them (e.g. the Da Vinci robot, mostly used nowadays in operating rooms). As such technologies develop and become more pervasive, with more general applications, the problem of how to teach a task to the robot in order to let him autonomously execute it in the future, or transmit its knowledge to others has raised. In this direction, an alternative of pure machine learning techniques, as reinforcement learning, was found (Peternel et al., 2014) in "Teaching by demonstration" (or imitation learning). Learning by demonstration provides a shortcut to allow transfer of some of the demonstrator's task knowledge to the robot. The transfer can be crafted by designers so that the transferred skill promptly generates the desired robot performance which can be played back on the robot. Alternatively, it can be used as initial point for the self-learning of the robot or utilize the operator's sensorimotor learning to tune the motor performance of the robot to obtain an autonomous controller. Although the initial approach was to do this through direct physical contact, it can have several drawbacks such as: the inability of human to teach the robot remotely in cases of operation in hazardous environments; difficulty to simultaneously control the impedance of the robot; and demonstrator induced dynamics during the teaching stage, which may alter the dynamical conditions between the demonstration and the autonomous stages. Teleoperation is again the natural solution of this problems: master operator is able to remotely transfer its motion to the robot without being physically coupled to it.

Several control strategies have been studied to achieve this goal, mostly including force feedbacks or haptic interfaces in order to insert the human operator in the robot control loop. However, successful results were found also without any kind of feedback given to the operator about the forces interaction between the robot and the environment (Ajoudani et al., 2012) (Peternel et al., 2014). This approach is still more simple and reliable, moreover, the control action is semi-autonomous: the human operator doesn't have to control every aspect of the robot-environment interaction and has to worry only about sending reference commands, letting to the local feedback on the slave side the other control aspects.
For these reasons, an unilateral architecture, without any feedback apart from the visual one, was chosen in this work.

A body-machine interface, the Teleimpedance (Ajoudani et al., 2012), was used to send human arm position references to a robotic arm (Kuka LWR in particular was used for experiments).
My contribution has been to enrich this framework with a local hybrid force/impedance controller, which has to autonomously manage the interaction with unknown external environment, maintaining a compliant and smooth behavior when unexpected contacts occur. The force control does a simple object-gravity compensation action in a grasping task, when an object is detected in the robot end-effector, distinguishing its weight-force from the other contacts.
Apart from the plain control action, stability of the whole system during was ensured implementing a passivity controller, based on the Energy-Tank paradigm. The energy tank regulates the action of the force controller of the robot when an unexpected contact loss occurs during the task execution, but is able to distinguish it from other external interactions, possibly with humans, that does not require any change in the control action. Therefore, objective of this work is to improve the stability of a teleoperated robot when it interacts with an unknown environment, allowing it to be used in complex cooperative tasks involving humans, without the risk that an unsafe motion could accidentally hurt them or cause injuries.

Outline The thesis is structured as following:

- Chapter 1: some examples of Human-Machine interfaces and Teleoperation systems are described.
- Chapter 2: description of the complete Teleimpedance framework which couples human operator's arm and the robot, and on the experimental setup used to calibrate it
- Chapter 3: general analysis of robot impedance regulation and specific description of the designed force/impedance controller
- Chapter 4: overview about passivity theory and on its main applications and description of the Energy-Tank paradigm.
- Chapter 5: description of the software architecture used to control our robot model (XBotCore), providing both kinematic and dynamic simulation. General characteristics of the chosen simulator: Gazebo.
- Chapter 6: parameter estimation and experimental results are presented and analyzed.
- Chapter 7: all the work done within this thesis is briefly summarized and some possible future extensions and improvements are presented.

Chapter 1

State of the Art

1.1 Teleoperation systems

Master-Slave teleoperation systems for unstructured and hostile environments have been studied and applied for a long time, its early establishments date back even before the emergence of robotics (Goertz et al., 1961). Earlier prototypes using position measurements from the human operator arm to be replicated by a rigid manipulator soon showed their limitations in dealing with interaction tasks in uncertain environments, due to high forces generating at contacts. The second generation of teleoperation systems therefore included means of feeding back to the operator information on the interaction forces between the slave robot and the remote environment.

1.1.1 Bilateral teleoperation

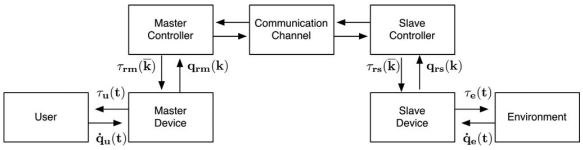
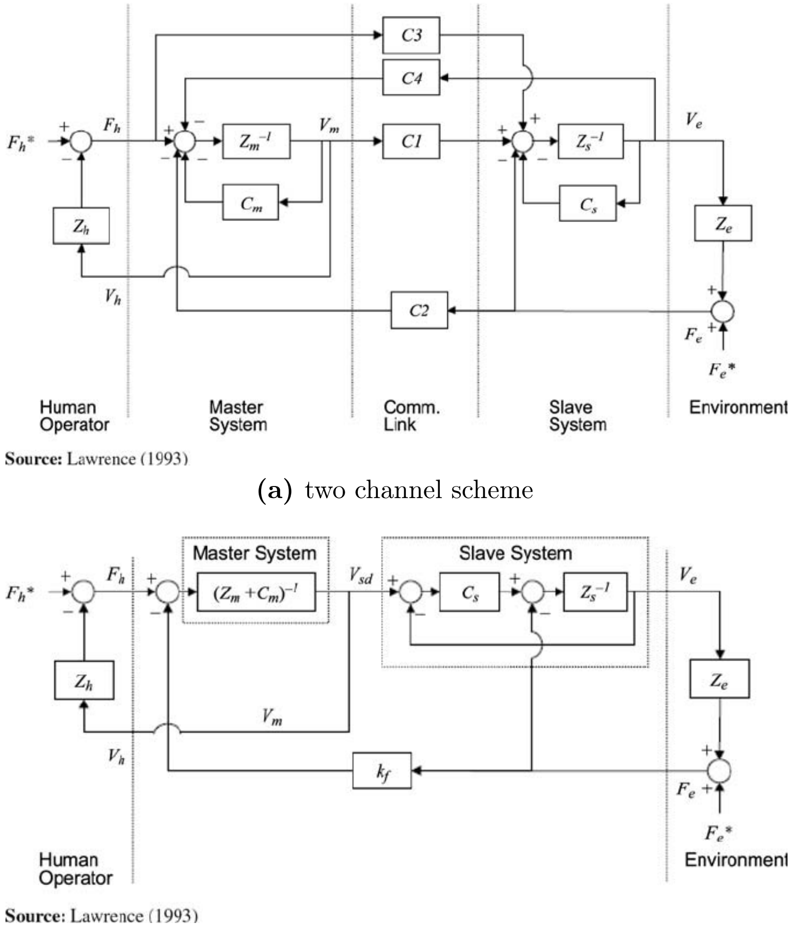

Master Controller
Communication Channel
Slave Controller
$\tau_{\mathbf{rm}}(\overline{\mathbf{k}})$
$\mathbf{q_{rm}}(\mathbf{k})$
$\tau_{\mathbf{rs}}(\overline{\mathbf{k}})$
$\mathbf{q_{rs}}(\mathbf{k})$
$\tau_{\mathbf{u}}(\mathbf{t})$
User
Master Device
$\dot{\mathbf{q}}_{\mathbf{u}}(\mathbf{t})$
$\tau_{\mathbf{e}}(\mathbf{t})$
Slave Device
Environment
$\dot{\mathbf{q}}_{\mathbf{e}}(\mathbf{t})$

Figure 1.1: general bilateral teleoperation scheme. (Lawrence, 1993)

A bilateral teleoperation system consists of the master and slave mechanical systems, often with some degree of "self control," i.e., control loops closed separately around master and slave. As described in (Lawrence, 1993), in the position-position architecture master and slave each have position control loops to enable tracking of position commands. Other control loops are constructed by establishing communication links between master and slave. In general, both positions and forces can be communicated bilaterally, as well as various filtered versions of positions and forces. Figure 1.1 shows a block diagram of the entire teleoperation system, including master, slave, bilateral communication, operator, and task (environment) dynamics. This architecture represents all teleoperation structures to appear, by suitable specialization of the subsystem dynamics. The external forces coming from the user and the environment are independent of teleoperator system behavior.

The nomenclature of the subsystem blocks is listed in figure 1.2, while in figure 1.3 typical values of various blocks for the two most common teleoperator architectures are shown.



Source: Lawrence (1993)

(a) two channel scheme

Source: Lawrence (1993)

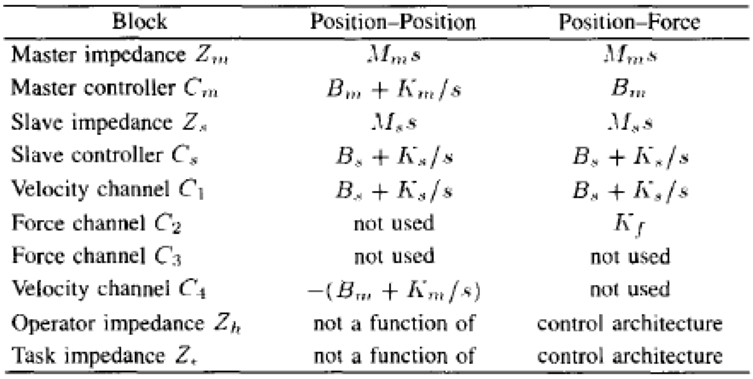
(b) four channel scheme

Figure 1.2: difference between teleoperation communication architectures using two or four channels (Lawrence, 1993)

The general structure of figure 1.2b includes various architectures recently seen in literature utilizing all four communication paths, as generalized position-position form, modified position-force strategies, and force-force architectures.

Arguments have been made for preferring one architecture over another,relating to the operator information-processing capacity and its performance studies. Although this type of functional information is of great importance to users, it is difficult to correlate to specific design choices and unambiguous measurements of teleoperator system properties needed by system designers.

This general architecture is used to analyze and quantitatively compare various teleoperation schemes in terms of transparency performance and stability. This systematic approach reveals that all four information channels between master and slave are necessary to achieve good transparency. It is also shown that transparency and robust stability (passivity) are conflicting objectives, and a trade-off must be made in practical applications (Franken et al., 2011), a more detailed description



Block	Position–Position	Position–Force
Master impedance Z_m	$M_m s$	$M_m s$
Master controller C_m	$B_m + K_m/s$	B_m
Slave impedance Z_s	$M_s s$	$M_s s$
Slave controller C_s	$B_s + K_s/s$	$B_s + K_s/s$
Velocity channel C_1	$B_s + K_s/s$	$B_s + K_s/s$
Force channel C_2	not used	K_f
Force channel C_3	not used	not used
Velocity channel C_4	$-(B_m + K_m/s)$	not used
Operator impedance Z_h	not a function of	control architecture
Task impedance Z_e	not a function of	control architecture

Figure 1.3: Nomenclature and typical values of subsystems in the general teleoperator architecture.

will be provided in chapter 4. Figure 1.3 provides a simplified description of the subsystem blocks in the general architectures described in figure 1.2. These typical forms for subsystem dynamics are intended to capture only the dominant effects, and will be used as concrete examples to accompany the analysis results. More complex descriptions of the blocks can be used to provide more realistic measures of teleoperation performance as needed for a given application. Note that velocities are often used in all dynamic descriptions rather than positions. Stability and transparency are unaffected by this convention, although communicating between master and slave positions. Since position indexing is common in bilateral teleoperation (to overcome workspace limitations of the master), and visual feedback is relied upon, this presents no difficulty. If transitions from teleoperated to autonomous operation are desired, initial positions can be communicated between controllers to obtain the desired absolute position registration.

Although such teleoperation systems can outperform pure position–controlled ones, they require an active device able to display force on the master side (for example an actuated exoskeleton), which imposes extra costs and discomfort for the operator. Furthermore, delays in the communication channel between the master and slave robot may generate insufficient transparency or even stability issues in the bilateral teleoperation system (Hannaford and Anderson, 1988, Imaida et al., 2004, Niemeyer and Slotine, 2004a). Despite the results and continuous improvements in the mechatronics and control of bilateral teleoperation systems (Leung et al., 1995, Eusebi and Melchiorri, 1998, Hannaford and Anderson, 1988, Sheridan, 1993), there are still many tasks in which stability and reduced transparency, if not mere cost of sensing and actuating reflected forces, prevents application of bidirectional teleoperation.

1.1.2 Unilateral teleoperation

In unilateral teleoperation, the information flow is in one direction, or in short, unidirectional. Master system that is driven by the human operator sends the necessary inputs (e.g. position, and/or velocity) through the communications line to drive the slave system. No information is sent back to the master system or the human operator during this type of manipulation. Instead, in most of the cases, the slave system has a local closed-loop control system which uses the feedback signals within this control system. Although the human operator sends the command signals to the slave system, actually all the monitoring is accomplished in the control system of the slave. This kind of architecture is particularly useful in semi-autonomous teleoperation systems where force feedback is not required. The main reason that make this architecture preferable is the significantly lower time delay, due to the fact that the control action is completely provided by the slave, and reference commands take half the time to be processed, leading to a more stable behavior.

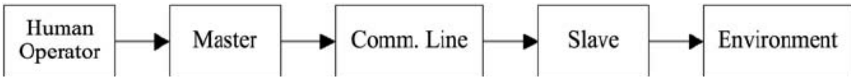


Figure 1.4: simple scheme of an unilateral teleoperation system

Tasks which are normally performed by humans without difficulty such as drilling, reaming, chipping and many others with large uncertainly in the environment constraints, cannot yet be easily conducted. This is not only due to the stability and transparency issues but also in many cases due to inadequate or low quality sensory information (such as position, force, velocity) which defines the mechanical work exchanged during the interaction of the teleoperated tool and the remote environment. Part of these problems are certainly related to the slave side, where rigid manipulator arms are used to replace the sophisticated, adaptively compliant structure of the human arm. But on the other hand, to change in this direction, not only the control action should be improved, but also the way in which human skills are transfered and coupled to machines should be, and have been, a matter of study.

1.2 Human interaction types

To develop effective modes of human-robot interaction (HRI), suitable interaction techniques are needed, and work in this field has experienced considerable growth in the past decade. Most HRI systems today are based on manual interaction techniques. Since the very beginning of teleoperation technologies, the first tipe of human-machine interaction studied was through a computer interface, at the beginning through the classical mouse and keyboard. However, other interaction paradigms, such as eye-tracking-based interaction, have emerged, which could lead to more natural interaction and teleoperation. Eye-tracking technology has evolved considerably since early implementations. Aulikki Hyrskykari explained the notion of gaze-attentive interfaces, examining the benefits and limitations of using eye movements in human-computer interaction (HCI).(Hyrskykari et al., 2005) The conclusion was that such systems have the potential to be more pleasing and effective than traditional interfaces. The advantages in terms of performance should be evaluated by studying human responses in a 2D environment and, generally, in human-computer and human-machine interfaces.

1.2.1 Eye tracking

Two conventional control interfaces that have been traditionally used to interface humans with computers are mouse and touchscreen devices, the human response performance in terms of speed has been compared with the one obtained by eye-tracking-based interfaces in (Hansen et al., 2004). Mainly two control techniques can be used in these kind of systems. One eye-tracking technique employed dwell time for object selection, and the other used a manual mouse-button click. Dwell time seems to be the more intuitive choice for selection when using an eye tracker, and it has the advantage of using just one device and leaving the user's hands free. However, in a pick-and-place task, users might run the risk of involuntarily selecting objects just by glancing at them. People normally use their eyes to gather information from the environment and not to issue commands. Using dwell time as an interaction paradigm can introduce a Midas touch problem, because it can be difficult to determine whether users are simply looking at an object or if they want to issue a command.

To avoid this dilemma, a second eye tracking technique was described in Roel Vertegaal's work, where results using a separate, distinct manual input as selection criteria (for example pressing a mouse button) were compared to the previous dwell time technique, showing a strong decrease in error rate, despite the slower performance (Vertegaal, 2008).

Multimodal approaches that combine gaze with other interaction modalities have also been explored, for instance, Darius Miniotas and his colleagues have shown how to improve the accuracy of gaze interaction by using speech. (Miniotas et al., 2006) Despite the attractiveness of using gaze control for HCI and HRI, clear advantages of this next-generation control mechanism are not always apparent. Such systems are often evaluated while users perform simple tasks, for which they may prefer conventional control techniques that are familiar and already efficient. More complex tasks that require the use of various attentional modalities, however, could yield different results. Cognitive psychology offers insight on the impact of task modality (for example, visual versus manual). Specifically, Multiple Resource Theory (MRT) provides predictions regarding performance on tasks that employ disparate cognitive resources, such as vision and hearing, as compared with multiple vision-based tasks (Wickens, 2002). This has implications for HRI systems design. Specifically, if the demands of a given operation require the application of overlapping attentional or cognitive resources, performance could be expected to decline for one or both of the tasks. This notion is particularly relevant to the development of new HRI techniques. Therefore alternative ways to send reference commands to machines have been explored, using biosignals like EMG and EEG.

1.2.2 Body-Machine and Brain-Computer Interfaces

The applications of robotic arms in assistive domains have demonstrated a high potential in improving the patient's quality of life by reducing their degree of dependence on caregivers, aiding them in activities of daily living (ADLs), such as controlling assistive devices like wheelchairs, tools, or computers. Patients who can benefit from this technology include those with upper body disabilities, as traumatic spinal cord injuries (SCI), paraplegic and tetraplegic patients (Casadio et al., 2011). To make robotic arms suitable for the execution of a large class of daily tasks, intuitive and user-friendly human-robot interfaces must be designed to associate the residual capabilities of the users with disabilities to appropriate robot functions.

There are many studies aiming at improving the usability of commercially available robotic arms in assistive scenarios, by designing control interfaces able to reduce user's cognitive burden and time required to accomplish a task. Such interfaces often rely on teleoperation or shared control paradigms. Typically the partially or fully autonomous robot behavior is adapted to the user's high level inputs, based on his/her residual limb motor capabilities, to enhance the task execution performance. An example of implementation is shown in (Artemiadis and Kyriakopoulos, 2010), where upper-limb movements are detected and mapped into high level control signals for a robotic arm. Here a shared-control framework for assistive manipulation is built on the concept of autonomous piecewise trajectory segments. In a similar vein in (Jiang et al., 2013), a haptic 3D joystick was adapted to directly control a robotic arm movements, enhancing the performance of traditional arm joystick control. In (Jia et al., 2007), a hand gesture recognition interface was combined with robot embedded object tracking and face recognition to control a wheelchair-mounted robotic manipulator.

However, for people suffering from more severe forms of motor disabilities, the above mentioned Body-Machine Interface examples cannot be exploited to drive a robot function.

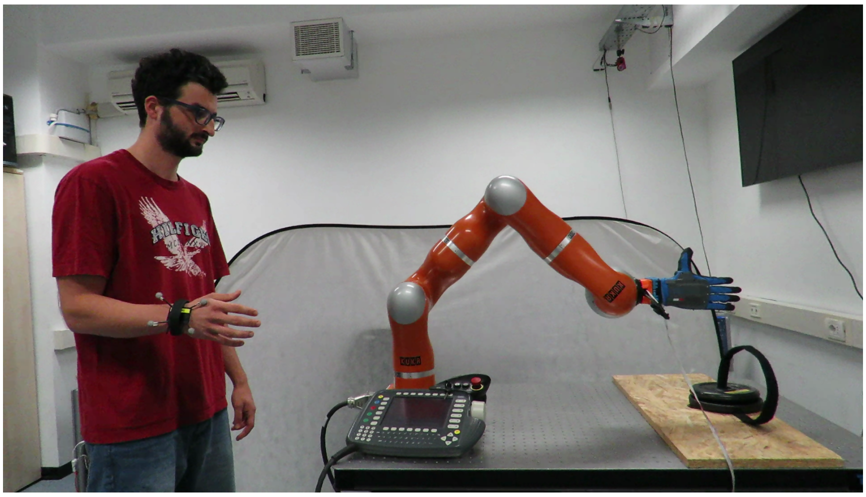


Figure 1.5: The proposed simple body-machine interface

To address this issue, former studies explored the use of brain signals, through non-invasive EEG-based Brain-Computers Interfaces (BCI) (Cincotti et al., 2008), to produce robot control by means of non-muscular channels voluntarily modulating EEG sensorimotor rhythms recorded on the scalp; this skill was learnt even though the subjects have not had control over their limbs for a long time. Robotic systems driven by brain signals, also known as Brain-Machine Interfaces (BMI), typically exploit P300 (Hoffmann et al., 2008), steady-state visual evoked potentials (SSVEP), or motor imagery paradigms, to generate high level commands to control a robot. However, despite the continuous improvements, BCI systems are still far behind those based on Body-Machine Interfaces in terms of performance and reliability. This is due to well-known issues, such as the same previously mentioned high cognitive loads issues on the user, especially in continuous control schemes, low performance in high degrees of freedom control, and lack of flexibility given that user choices are mostly predefined.

To overcome the limitations of pure BCI systems, hybrid interfaces have been proposed that exploit a combined use of EEG and another biosignal (Pfurtscheller et al., 2010), as those based on BCI and gaze, to enhance communication or robot control. It is worth mentioning that most of the existing solutions focus on aspects related to the ease-of-use and intuitiveness of the user interface rather than on the control of physical interaction capabilities of the robot. The latter feature can contribute to enhanced interactions in the human-robot-environment loop, and the underlying safety. When developing a robotic assistive tool, a safe behaviour must be guaranteed e.g., for eating/drinking or personal hygiene tasks, as well as in case of accidental arm collisions with the environment. A well-known strategy to ensure safety in a robotic system is the active impedance control, which regulates the interaction between the robot and its environment, including a virtual compliance at the joint level between the motor and the output, which causes the human operator to perceive a softer arm. This control action will be later discussed and used in our work, besides a supplementary energy-tank based stability controller, used to provide a safe and compliant interaction in our system.

Chapter 2

Teleimpedance

From all the existing techniques illustrated in section 1.2.2, a Body-Machine interface was chosen as master-side controller. This particular framework, called Teleimpedance, was described for the first time by (Ajoudani et al., 2012). For our purposes, position references should be set by the human operator, while commanded low stiffness provide the compliant and safe behaviour we look for, while the stability issues that may occur on the slave side (e.g. the force term, as explained in chapter 4) are autonomously solved. However, on its most complete version, Teleimpedance is meant to provide also stiffness references. To achieve this, aside the signals coming from passive markers (used for positions), EMG sensor are used, and a proper model able to find a correspondence between human arm and robot impedance should be found. In the following sections the techniques used to describe the complete model are shown, before understanding how to extend this framework with a proper stability controller in the next chapters.

2.1 Human Arm Impedance Modeling in 3D

Contact reliability during tasks performed by humans in environments with stochastic uncertainties or while being exposed to mechanical disturbances can be ensured by a stable mechanical interface between human arm endpoint and the environment (Osu and Gomi, 1999, Burdet et al., 2001, Perreault et al., 2004). Quantification of this stability could be done by measurements of visco-elastic profile of human arm endpoint. Early studies conducted by Mussa-Ivaldi et al. (Mussa-Ivaldi et al., 1985) demonstrated spring like properties of human muscles. At around the same time, human arm endpoint stiffness profile was estimated by means of imposed displacements and resulting steady state force responses. Other studies analyzing the overall human arm impedance have also been carried out (Dolan et al., 1993, Tsuji et al., 2004, 1995, Gomi and Osu, 1998).

Perturbation based impedance estimation techniques are currently the most accurate and reliable, due to direct measurements of the applied force–displacement profiles. However, their application can be problematic when real-time estimation of the human arm impedance is required during the execution of the task. As a consequence, current work is devoted to application of continuous and adaptive biosignals (e.g. EMGs) which correlate with human impedance modifications (Osu

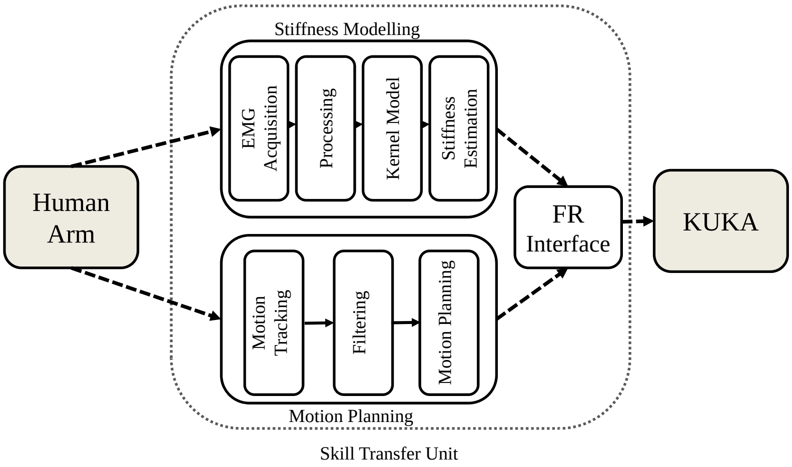


Figure 2.1: The overall block diagram of tele-impedance, (Ajoudani et al., 2012)

and Gomi, 1999). This work proposes a new method for the estimation of human arm endpoint stiffness, based on extraction of stiffness-related EMG features in real-time.

Task-oriented modifications of the endpoint arm stiffness along with endpoint force variations in humans are shown to have strong correlation with the patterns of activations in the involved muscles (Osu and Gomi, 1999, Selen et al., 2005, Franklin et al., 2003). Agonist-antagonist muscle co-contractions affect and modify selectively the overall stiffness and viscous profiles of the arm endpoint. Resulting modifications in force and impedance can be regarded as the effects of internal force regulation exerted by group of extensor and flexor muscles. It should also be noted that, when muscles act on multiple limb segments or about multiple axes of rotation (as is the case at the shoulder joint), the separation of muscles into agonist and antagonist can be oversimplifying, and a more general concept of muscle groupings or synergies should be considered (Turvey, 2007).

Based on this and the observed approximate linear dependency of force/stiffness to muscular activation/coactivations (Selen et al., 2005, Franklin et al., 2003), the overall mapping between muscular activities and resulting arm endpoint force (T_F) and stiffness (T_σ) in Cartesian coordination and around equilibrium position (close to isometric condition) can be compactly described by

$$\begin{bmatrix} F \\ \sigma \end{bmatrix} = \begin{bmatrix} T_F \\ T_\sigma \end{bmatrix} P + \begin{bmatrix} 0 \\ \sigma_0 \end{bmatrix}, \tag{2.1}$$

where $F, \sigma \in \mathbb{R}^3$ are the endpoint force and stiffness vectors, respectively, σ_0 is the intrinsic stiffness of the arm in relaxed conditions, and $P \in \mathbb{R}^n$ is the vector of muscular activities of the n considered muscles, as obtained from preprocessing EMG signals from electrodes applied on each muscle (details on the preprocessing are given in Section (2.2).

Ideally, a model for the estimation of endpoint force and stiffness values (in the vicinity of a specific arm configuration) from measured muscular activities could be obtained by identification of the elements of T_F and T_σ. To obtain an accurate experimental identification, two ingredients are necessary. Firstly, precise measurements of 3D endpoint force F and stiffness σ are needed. Secondly, a rich and varied enough set of data samples should be generated by the subject. As a result of both these issues, while identification of the EMG–to–force map T_F is relatively easily done, to accurately identify the EMG–to–stiffness map T_σ is more difficult.

Because of such limitations in the estimation of T_σ, R. Osu and H. Gomi (Osu and Gomi, 1999) proposed an algorithm for estimating the human arm stiffness which relies on identifying the map between muscular activities (as coded by EMGs) and joint torques (in formulas, $\tau = T_\tau P$, where T_τ is the EMG–to–joint torque map). In this work, authors estimate T_τ by combining shoulder and elbow torques in the horizontal plane with measured activities of six involved muscles. The identification of $T_\tau \in \mathbb{R}^{2\times 6}$ is based on an assumed structure of the type

$$T_\tau = \begin{bmatrix} c_1 & -c_2 & 0 & 0 & c_5 & -c_6 \\ 0 & 0 & c_3 & -c_4 & c_7 & -c_8 \end{bmatrix}, \quad c_i > 0,\ i = 1, \dots, 8 \tag{2.2}$$

which corresponds to the anatomical arrangement of the six considered muscles. Here, c_i are constant gains which correspond to the mapping from muscular activities to the joint torques, identified by experiments. Authors then propose an index of co-contraction around the joint (IMCJ, denoted here as L) that is evaluated as

$$L = |T_\tau|\ P,$$

where the absolute value of matrix T_τ is meant elementwise. Each element of $L \in \mathbb{R}^2$ is considered to be linearly related to the corresponding joint stiffness, i.e.

$$\sigma_i = m_i L_i + \sigma_{0,i}, \ \ i = 1, 2$$

where the constants $m_i, \sigma_{0,i}$ are experimentally identified on the basis of stiffness measurements (Osu and Gomi, 1999, Osu et al., 2002).

It should be noted that the method relies upon the availability of an accurate estimate of joint torques, which can not be measured directly, and have to be computed from measured endpoint forces through the arm Jacobian as $\tau = J^T F$. Results based on the above method have demonstrated good performance for 2D joint stiffness estimation even during small movements in the vicinity of the posture.

However, application of this method to our problem poses some challenges. In our system, we want to estimate the 3D cartesian endpoint stiffness using measurements of the activation of $n = 8$ involved muscles acting at the shoulder and elbow. Firstly, as observed in (Weiss, 1941), anatomy does not warrant the separation in agonists and antagonists at a ball joint, as “any one of the muscles of the shoulder can be engaged in phase or antiphase with any other muscle” (Turvey, 2007). Furthermore, the use in (Osu and Gomi, 1999) of an estimate of the arm Jacobian has the disadvantage of introducing errors due to imperfect adherence of the double shoulder and elbow joints to simple lower pair kinematic models (Shin et al., 2009), and a strong dependency on the subject’s limb proportions. In conclusion, the assumption of a structure of the type (2.2) for the EMG–to–force map $T_F \in \mathbb{R}^{3\times 8}$ is not justified in our case. By the same token, no simple relation between the EMG–to–stiffness map $T_S \in \mathbb{R}^{3\times 8}$ and $|T_F|$ can be expected to hold.

To overcome the limitations of applicability of the method of (Osu and Gomi, 1999) in our setting, we exploit the observation that in our experimental conditions the generation of forces can be considered to be decoupled from the regulation of stiffness. It is well known that in general end-point impedance has three components, depending on posture, force, and co-contraction, respectively. While the first two components may be large and even dominating (Milner, 2002) in a large enough range of variations, an ample literature reports the existence and independence of co-contraction contribution to stiffness (Akazawa et al., 1983, De Serres and Milner, 1991, Milner et al., 1995). Because in our experiments we only use small motions about a reference configuration for the arm and small end-point force variations dependent from the master reference command (large force variations due to the object compensation force are completely autonomous from the teleoperation system, and are treated by the controller as an additional gravity compensation term, as described in section 3), it can be safely assumed that the control of forces and stiffnesses are decoupled.

Accordingly, for the first-order linearized model (2.1), we consider a decomposition of the space of muscular activations $\mathcal{P} \ni P$ as the direct sum of a force–generating subspace $\mathcal{P}_F$ and the force-map null space $\mathcal{P}_k = \mathsf{ker}\ T_F$, i.e.

$$\mathcal{P} = \mathcal{P}_F \oplus \mathcal{P}_k.$$

By choosing a right-inverse T_F^R of T_F, i.e. any $n \times 3$ matrix[1] such that $T_F T_F^R = I$, we also affix a system of coordinates to these subspaces. In these coordinates, we can decompose the vector of muscular activations P in a force–generating component P_F and a null–space component P_k as

$$P = T_F^R T_F\, P + \left(I - T_F^R T_F\right)\, P \overset{def}{=} P_F + P_k.$$

The null–space component P_k contains information on the co-contraction component of stiffness generation.

It is convenient to give an alternative description of P_k as follows. Let N_F denote a basis matrix for the kernel of T_F, and let $\lambda = N_F^+ P_k = QP$ be the coordinates in that basis of P_k, where $Q \overset{def}{=} N_F^+ \left(I - T_F^R T_F\right)$. Hence the model of cartesian stiffness regulation through co-contraction is written as

$$\sigma - \sigma_0 = M_\sigma\, Q\, P \tag{2.3}$$

where $M_\sigma \in \mathbb{R}^{3\times 5}$ is a mapping from the kernel of T_F (the set of muscle activations that do not change endpoint force, in the selected coordinate frame) to stiffness variations. The map M_σ can then be identified and calibrated once, based on direct measurements of human arm end point stiffness, at different coactivation levels as described in the following section. It is worth to explicitly note that the choice of the right-inverse T_F^R and that of the kernel basis N_F are both arbitrary, and

[1] The existence of a right inverse is guaranteed by the fact that in nonsingular configurations T_F is full row-rank. Because $n > 3$, there exist infinite right-inverses: a particular choice is for instance $T_F^+ = T_F^T (T_F T_F^T)^{-1}$, i.e. the pseudoinverse of T_F.

imply an arbitrary description of the co-contraction component, for which only the calibration matrix M_σ is relevant. In other terms, there is no “natural” choice of coordinates for $\mathcal{P}_k$, but, once one is given, this has to be used consistently for system calibration and operations.

Several caveats apply to the method for 3D stiffness estimation above, among which the fact that extrapolation of identified values of the maps to the whole workspace of the arm might not be valid, due to configuration dependency of human endpoint impedance and nonlinear dependency of the muscular activity and muscle tensions during movements. Also, large force variations would bring about stiffness changes of the same order or larger than co-contraction, thus violating the validity of the linearized model. However, as far as the tasks to be performed by the master arm remain in a vicinity of the static posture in which the parameters are estimated, and the human arm is not applying external forces, the approach provides a reasonable approximation of the endpoint stiffness profile in real time.

2.2 Stiffness Model Calibration-Identification

Experiments were also done by (Ajoudani et al., 2012) in order to identify and calibrate the model.

Materials and Methods One healthy subject (male; age 27) stood upright with the feet side–by–side in front of a robotic arm.

The robot arm was equipped at the endpoint with a handle connected to a 6–axis force and torque (F/T) sensor (ATI Mini-45). An idle spherical joint was used to connect the center of the handle to the F/T sensor, so as to avoid exertion of torques by the subject (see fig. 2.2). Displacements of the human arm at its endpoint (considered as the wrist center) were tracked by an Optitrack system (Natural Point, Inc.), with nominal resolution is 0.02 mm. Optical markers were also placed at the shoulder and elbow of the subject’s arm. Both force and position measurements were acquired at a sampling frequency of 200 Hz, and filtered by a low-pass Butterworth filter with cutoff frequency 15 Hz to eliminate noise. In all experiments, surface electromyograms (EMGs) were used to trace muscular activities. Eight dominant muscles acting on elbow and shoulder joints (see Table 2.1 and fig. 2.3) were chosen as the sources of EMG recordings. Analogue EMG signals were measured and amplified with a Delsys-Bangoli 16 (Delsys Inc.) apparatus. Acquired signals were band-pass filtered in the frequency range $[20, 450]$ Hz. Resulting EMG signals were sampled at 2 kHz (PCI-6220, National Instruments) and full rectified for further processing. A digital non-causal FIR linear phase low-pass filter was used for the detection of the envelope of the signal, which approximately corresponds to muscle activity. The robotic arm is a 7 degrees-of-freedom KUKA LWR with DLR’s Fast Research (FR) Interface (Schreiber et al., 2010). The data acquisition and synchronization interface between the KUKA controller, the EMG acquisition board, the Optitrack position streaming data, the six axes F/T sensor were developed in C++.

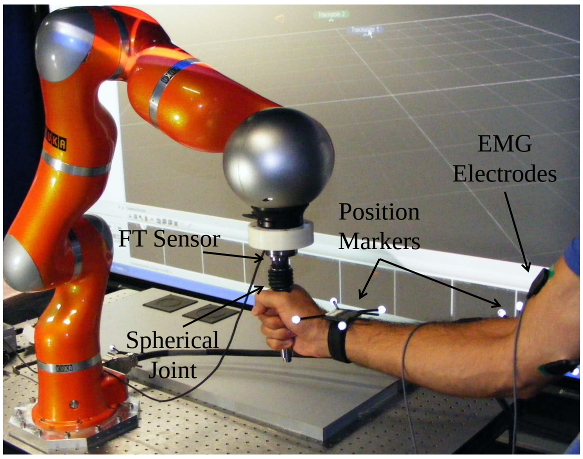


Figure 2.2: Experimental setup used for the first calibration experiments. Subject applies constant forces in 6 directions while holding the handle attached to an idle spherical joint (Ajoudani et al., 2012).

2.2.1 Identification of the EMG–to–Force Map

In a first set of experiments, aimed at estimating the force–generating map T_F, the robot was only used passively, with motors inactive and brakes turned on, serving only as a support structure for the F/T transducer mounted at endpoint of the arm. The subject was asked to apply constant forces of ±5N, ±10N and ±20N, respectively, along 6 directions ($\pm x$, $\pm y$ and $\pm z$) while holding the handle (isometric conditions). Force components applied by the subject were measured and shown in real time to the subject himself in the form of a graph with three colored bars on a screen. In each trial, the subject was instructed to apply forces so that bars reached and maintained given targets (corresponding in turns to the three different intensities in the 3 different directions). The subject was also instructed to use minimum muscular activity (minimum effort). Each trial was 60 seconds long. Data form the first 10 seconds were discarded to eliminate transient force fluctuations. For each direction and force level, 4 trials were executed and recorded (for an overall number of $4 \times 3 \times 2 \times 3$ trials) in EMG–to–force map identification experiments. In a subsequent postprocessing phase, all components of (T_F) were evaluated by means of a least–squared–error algorithm, and a basis of its nullspace

Table 2.1: Muscles used for EMG measurements

Flexors		Extensors	
Monoarticular	Biarticular	Monoarticular	Biarticular
Deltoid clavicular part (DELC)	Biceps long head (BILH)	Deltoid scapular part (DELS)	Triceps long head (TRIO)
Pectoralis major clavicular part (PMJC)		Triceps lateral head (TRIA)	
Brachioradialis (BRAD)		Triceps medial head (TRIM)	

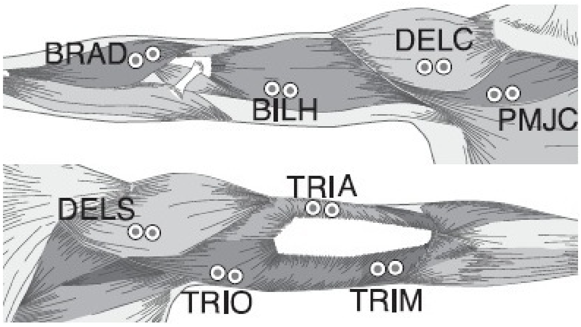


Figure 2.3: Electrode positions in EMG measurements.(Ajoudani et al., 2012)

and the projector matrix Q used in (2.3) were computed.

2.2.2 Identification of the EMG–to–Stiffness Map

In a second set of experiments, aimed at estimating the stiffness–regulating map M_σ, similar equipment and arrangements were used. In this case, however, the robot was actively controlled and its Cartesian stiffness was set to the highest value possible (3 kN/m in all directions). To minimize voluntary stiffening behavior of the subjects arm, following Perreault et al. (Perreault et al., 2004) we applied continuous stochastic perturbations to the subject's hand through the handle in x, y and z directions. The amplitude of the applied perturbations had the peak-to-peak value of 20 millimeters in each direction. Figure 2.4 shows a typical applied endpoint displacements and restoring forces along x, y and z directions. Frequency spectrum of the perturbations were flat while decaying at the rate of 40 dB/Hz in frequencies higher than 4 Hz (Figure 2.5). This perturbation profile and corresponding forces in response, ensure adequacy of data for the identification of endpoint dynamics (Mann et al., 1989).

The experiments for EMG–to–stiffness map identification were performed by measuring the subject's force response to these random perturbations while the subject was instructed to set his arm's stiffness at a given level. Restoring forces exerted by the subject's arm were monitored by means of the F/T sensor, positions of the subject's wrist center were measured via the optical tracking system, and EMGs were acquired simultaneously and synchronously.

A rough stiffness indicator was graphically shown consisting of a bar of length proportional to the norm $|P|$ of the vector of muscle activations. Four different stiffness reference levels were provided in different trials. The first set of random position perturbations was applied to the subject's hand at minimum muscle activity level (relaxed arm). The corresponding data was used to estimate the intrinsic stiffness σ_0 based on the identification protocol described below. Also, the endpoint inertia parameters in the vicinity of the predefined posture were identified in relaxed conditions, to be used later as a constant (under the simplifying assumption of negligible influence of muscle mass distribution on endpoint inertia) for the estimation of endpoint visco-elastic parameters in higher cocontraction trials. After such experiments in relaxed conditions, the subject was asked to stiffen his

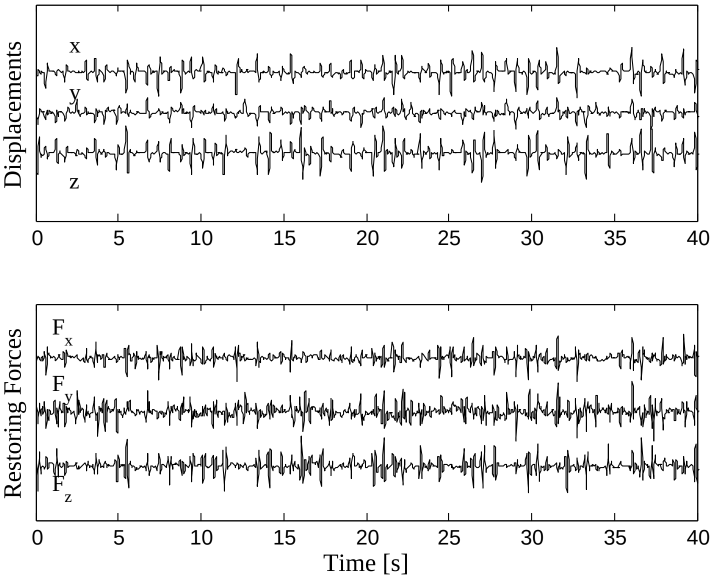


Figure 2.4: Typical applied endpoint displacements and restoring forces in x,y and z directions. Each trial was 40 s long which the last 35 s of the rials were used for the further processings.(Ajoudani et al., 2012)

arm as hard as possible. The three reference levels of stiffness were hence defined by subdividing in four equal parts the measured interval of $|P|$ between its relaxed and maximum values.

Instructions given to the subject in the EMG–to–stiffness map identification phase included keeping the visualized index $|P|$ close to the set value during the experiment. Trials where the deviation from the preset level of $|P|$ were larger than 10% were discarded. Five successful trials were recorded for each level of arm stiffness. Each perturbation trial lasted 40 seconds. The first 5 seconds were used for allowing the subject to adapt to required stiffness level. The force on the handle caused by arm weight was removed as bias, being constant for small deviations from the equilibrium configuration.

The estimate of endpoint stiffness in different trials was performed based on measurement of corresponding pairs of forces and positions at the subject's wrist, following standard methods (see e.g. (Perreault et al., 2004)). For this reason, multiple–input, multiple–output (MIMO) dynamics of the endpoint impedance was decomposed into the linear subsystems associating each input to each output. Based on this assumption, and indicating with $F_x(f)$, $F_y(f)$ and $F_z(f)$ the Fourier transforms of the endpoint force along the axes of the cartesian reference frame, with $x(f)$, $y(f)$ and $z(f)$ the transforms of the human endpoint displacements, the dynamic relation between the displacements and force variations can be described by

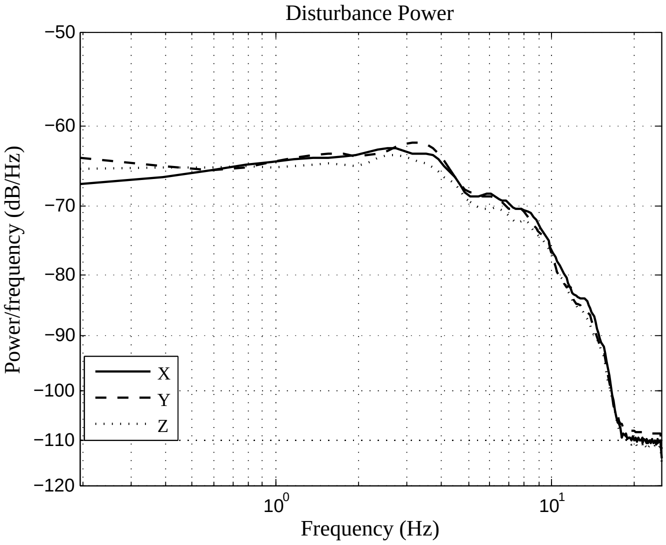


Figure 2.5: Power spectrum of the applied disturbances.(Ajoudani et al., 2012)

$$\begin{bmatrix} F_x(f) \\ F_y(f) \\ F_z(f) \end{bmatrix} = \begin{bmatrix} G_{xx}(f) & G_{xy}(f) & G_{xz}(f) \\ G_{yx}(f) & G_{yy}(f) & G_{yz}(f) \\ G_{zx}(f) & G_{zy}(f) & G_{zz}(f) \end{bmatrix} \begin{bmatrix} x(f) \\ y(f) \\ z(f) \end{bmatrix} \tag{2.4}$$

A non-parametric algorithm was adopted to identify the empirical transfer function of each of the SISO subsystems described above in frequency domain (MATLAB, The MathWorks Inc.). The smoothed spectral estimates of input and outputs (using windowing techniques) were fed to this algorithm in order to identify the each SISO transfer function. Consequently, we adopted a parametric, second order, linear model of each impedance transfer function of the type

$$G_{ij}(s) = I_{ij}\mathrm{s}^2 + B_{ij}\mathrm{s} + K_{ij}, \mathrm{s} = 2\pi f\sqrt{-1} \tag{2.5}$$

where I, B and K denote the endpoint inertia, viscosity and stiffness matrices, respectively. The parameters of the second order linear model were identified based on least squares algorithm in frequency range from 0 to 10Hz. Although simple, this model has been shown to be adequate to the representation of the endpoint impedance of the human arm in a large class of tasks (Perreault et al., 2004, Tsuji et al., 2004).

Finally, in the post-processing phase, experimental EMG vectors P were mapped in the EMG–to–force map nullspace through the previously computed projector matrix Q. The elements of the stiffness matrix K were used as estimates for the components of σ, and the map M_σ was estimated by applying a least–squared–error method to (2.3).

2.2.3 Identification Results

The strength of linear dependency between measured force signals and estimates via the least-squared-error identification of the components of T_F was evaluated by

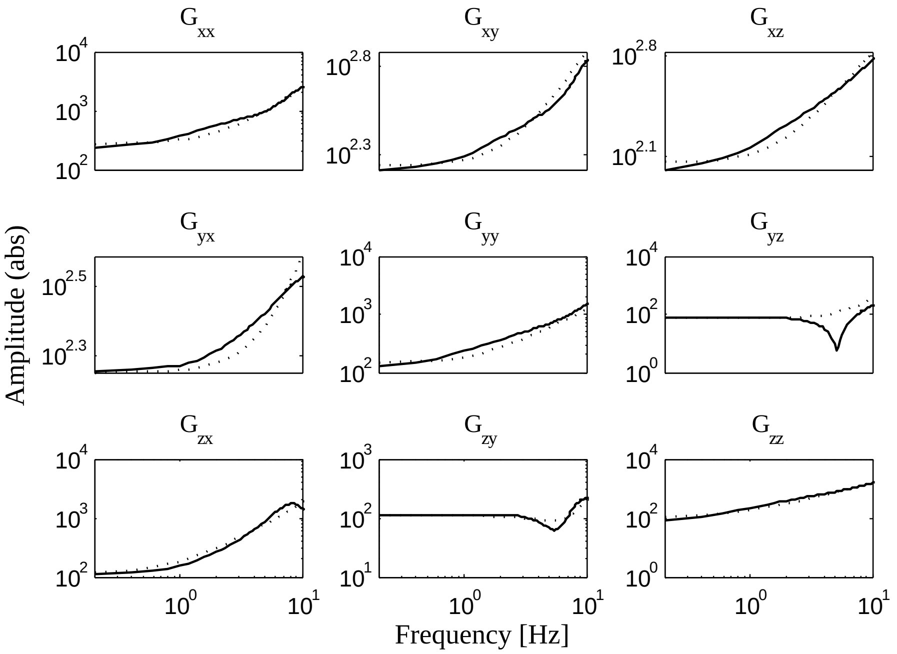


Figure 2.6: Non–parametric (solid lines) and parametric second order (dotted lines) transfer functions of SISO impedance subsystems obtained from stochastic perturbations.(Ajoudani et al., 2012)

Pearson's product-moment correlation coefficient. The coefficient is defined as

$$R_k = \frac{\sum \hat{f}_k f_k - \frac{\sum \hat{f}_k \sum f_k}{N}}{\sqrt{(\sum {f_k}^2 - \frac{(\sum f_k)^2}{N})(\sum \hat{f}_k^2 - \frac{(\sum \hat{f}_k)^2}{N})}}, \quad k = x, y, z \tag{2.6}$$

where f_k and $\hat{f}_k$ are measured and estimated values of force in the Cartesian directions, and N is the number of pairs of data. The fit was consistently good in the three directions, resulting in average $R^2 = 81\%$. Figure(2.6) demonstrates the results of non–parametric and second order model identification of the hand impedance transfer functions in the frequency range from 0 to 10Hz, according to methods described above. The second order parametric impedance models presented (69.7%) of the data variance across all directions in minimum muscular activity trials in the frequency range of 0 to 10 Hz. Values of estimated endpoint inertia, damping and stiffness along the Cartesian directions are presented in table (2.2). Estimated values are in good accordance with the former results of impedance estimation in 2D (Dolan et al., 1993, Tsuji et al., 1995) and 3D (Trumbower et al., 2009).

Extracting the symmetric and antisymmetric parts ($K_s = (K + K^T)/2$ and $K_a = (K - K^T)/2$, respectively) of the estimated stiffness matrix, it can be observed that the estimate is rather strongly symmetric ($\|K_a\|/\|K_s\| \approx 0.04$).

Table 2.2: Estimated hand impedance parameters.

Stiffness (N/m)			Viscosity (Ns/m)			Inertia (Kg)		
K			B			I		
240.58	163.99	102.35	21.33	7.09	6.96	1.02	0.32	0.15
181.81	128.97	78.78	3.77	17.72	3.29	0.17	0.78	0.09
113.36	107.18	81.01	23.12	1.90	26.06	0.19	0.12	0.87

Chapter 3

Controller design

Now that the system setup on the master side has been described, let's switch to the controller which has been designed in the slave side. Here, as said, an hybrid force/impedance controller was implemented, with the main goal of maintaining a safe and compliant behavior while following master's references, therefore compliance/stiffness regulation in interaction with external environment has to be analyzed in the first place.

3.1 Robot compliance regulation

In contrast to pure position control which rejects disturbance forces in order to track a given reference motion trajectory, (Albu-Schaffer and Hirzinger, 2002), three control techniques can be used to obtain the desired compliant robot behavior in Cartesian space.

Stiffness control In all the position-based control systems, the main assumption to make in order to grant a smooth and stable behavior is that the gains in the overall control loop in the Cartesian space, are substantially lower than those in the local joint controllers of the robot. This leads to the idea of converting the desired Cartesian stiffness $\mathbf{K_d}$ and damping $\mathbf{D_d}$ to corresponding matrices for the robot local joint stiffness $\mathbf{K_j}$ and damping $\mathbf{D_j}$. Assuming that the desired Cartesian matrices are changing rather slowly, there is no need for very high Cartesian sampling rates. If the inverse kinematics of the robot $K^{-1}(x)$ is known, the joint impedance controller S_R can then use the resulting joint positions and the local joint stiffness and damping to generate the desired motor torque $\boldsymbol{\tau_m}$:

$$\{\mathbf{K_j}, \mathbf{D_j}\} = T(\mathbf{K_d}, \mathbf{D_d}) \to \boldsymbol{\tau_m} = S_R\{K^{-1}(\mathbf{x_d}), \mathbf{K_j}, \mathbf{D_j}\} \tag{3.1}$$

where T maps the matrices $\mathbf{K_k}$, $\mathbf{D_k}$ from Cartesian to joint space ($\mathbf{K_j}$,$\mathbf{D_j}$). The defined joint stiffness and damping give a local relationship between the joint torque and the joint positions or velocities. The idea of mapping the Cartesian stiffness to the joint stiffness is biologically motivated. The human arm is able to change its stiffness by contracting antagonistic muscle pairs. In case of robotic manipulators, this can be accomplished only by means of controlling, the input for the robot is always the motor torque, thus a more detailed coupling of the two, as presented in chapter 2, is described in detail in (Ajoudani et al., 2012).

Admittance control A proper measure of effectiveness of the compliant motion control, as defined in (Seraji, 1994), is the mechanical admittance Y, defined as

$$Y = \frac{v_f}{F} \tag{3.2}$$

where v_f is the end-effector velocity and F is the contact force, both at the point of interaction. A large admittance corresponds to a rapid motion induced by applied forces; while a small admittance represents a slow reaction to contact forces, therefore, the admittance $Y(s)$ relates the force error $e_f = F_r - F$ to the end-effector velocity perturbation $v_f(s)$. Consequently, the force compensator of the robot $K_F(s)$ attempts to comply with the environmental interaction and react quickly to contact forces by rapidly modifying the reference motion trajectory, its transfer-function is expressed as

$$\mathbf{K_F}(s) = \frac{1}{s}\mathbf{Y(s)} \tag{3.3}$$

For a known environmental stiffness, an admittance $Y(s)$ can be constructed to achieve a desirable force response with small or zero error, low overshoot, and rapid rise time.

The Cartesian force at the end-effector is usually measured, by a 6DOF force-torque sensor. The force vector is used to generate a desired Cartesian position x_d. Using the inverse kinematics K^{-l}, this displacement is converted to desired joint positions q_d. The joint position controller P_r then generates the motor torques τ_m:

$$\mathbf{x_d}(s) = \mathbf{x_0}(s) - \frac{\mathbf{\Delta f}(s)}{\mathbf{K_d} + \mathbf{D_d}s} \rightarrow \mathbf{q_d} = K^{-1}(\mathbf{x_d}) \rightarrow \boldsymbol{\tau_m} = P_r(\mathbf{q_d}) \tag{3.4}$$

where K_d and D_d are respectively the desired stiffness and damping. This method is the most commonly used one, since most robots have only a position interface. The advantages are that the high gain position controllers can compensate for the friction in the joints and that, for the implementation of high stiffness, low gains are needed in the Cartesian control loop. Hence it is clear that stability problems will appear for low desired stiffness and damping, for which the bandwidth of the Cartesian control loop approaches the joint bandwidth. This problem is even more noticeable for flexible joint robots, since in that case the bandwidth of joint control is more critical. Further problems arise in the vicinity of singularities, where Cartesian position control can typically lead to fast, destabilizing movements. For all these reasons, in our framework, in which the desired stiffness is low, impedance control was chosen.

3.1.1 Impedance Control

Overview In this approach, the actual Cartesian position is computed from the joint position using direct kinematics $\mathbf{x} = K(\mathbf{q})$, therefore, using the transposed Jacobian $\mathbf{J}^T(\mathbf{q})$, the Cartesian force is transformed into desired joint torques. The generated joint torque command $\boldsymbol{\tau_i}$ will be in the form:

$$\mathbf{F_i} = \mathbf{K_d}\boldsymbol{\Delta}\mathbf{x} + \mathbf{D_d}\boldsymbol{\Delta}\dot{\mathbf{x}} \rightarrow \boldsymbol{\tau_i} = \mathbf{J^T}(\mathbf{q})\mathbf{F_i} \tag{3.5}$$

The impedance controller is, in principle, complementary to the admittance controller. It is well suited for low stiffness and damping, which now require low gains in the Cartesian loop, while the bandwidth of the torque controller is optimally exploited. The stability problems will appear correspondingly for high Cartesian stiffness, witch is not the purpose of this work. The behavior at singularities is also different from that of the admittance controller: components of the force which act in singular directions, are not mapped to the joint space. The movement in points close to singularities will be stable and smooth, but on the other hand the stiffness matrix will be distorted, this aspect will have a marginal effect on our controller, in witch low stiffness is commanded.

Implementation In order to understand how the impedance controller was implemented in our system, the dynamics of our slave robot should be analyzed, in order to understand how it can be controlled by human position references given in the Cartesian space. It is known that the rigid body dynamics, with n degrees of freedom, can be formally described by the Euler-Lagrange model:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau_m} + \boldsymbol{\tau_{ext}} \tag{3.6}$$

where $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$ are respectively the joint position, velocity and acceleration vectors, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ denotes the mass matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \in \mathbb{R}^n$ is the Coriolis and centrifugal vector, and $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^n$ is the gravity term. The input of the system is represented by the controlled motor torque vector $\boldsymbol{\tau_m} \in \mathbb{R}^n$, while the $\boldsymbol{\tau_{ext}} \in \mathbb{R}^n$ vector comprises all the externally applied torques. The position of the end-effector in the Cartesian space, can be described by a set of local coordinates $\mathbf{x} \in \mathbb{R}^m$; if the forward kinematics $\mathbf{x} = K(\mathbf{q})$ is known, velocity and acceleration in Cartesian space can be computed via the Jacobian $\mathbf{J}(\mathbf{q}) = \frac{\partial f_k(\mathbf{q})}{\partial \mathbf{q}}$ as

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{3.7}$$

$$\ddot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} \tag{3.8}$$

External wrenches $\mathbf{F_{ext}}$ can also be mapped in the joint space through the Jacobian matrix of the manipulator $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{m \times n}$, so that $\tau_{ext} = \mathbf{J^T}(\mathbf{q})\mathbf{F_{ext}}$. Considering this, the control action of our force-impedance controller in the Cartesian space is expressed by the force

$$\mathbf{F}_m = \mathbf{J}(\mathbf{q})(\mathbf{F_i} + \mathbf{F_c})$$

where $\mathbf{F_c}$ is the reference compensation force as will be later defined in (3.13), and F_i the impedance force, which can be written in the same form explained in 3.5, as:

$$\mathbf{F_i} = \mathbf{K_d}\tilde{\mathbf{x}} + \mathbf{D_d}\dot{\tilde{\mathbf{x}}} \tag{3.9}$$

where, $\mathbf{x_d}(t) \in \mathbb{R}^m$ is the desired position, $\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x_d}(t)$ represents the position error, $\mathbf{K_d} \in \mathbb{R}^{6\times 6}$ and $\mathbf{D_d} \in \mathbb{R}^{6\times 6}$ are respectively the desired stiffness and damping, which are provided by the master as described in chapter 2.
It is also possible to define the inertia matrix $\boldsymbol{\Omega}(\mathbf{x})$ and the Coriolis/centrifugal matrix $\boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}})$ with respect to the Cartesian space coordinates x as

$$\boldsymbol{\Omega}(\mathbf{x}) = (\mathbf{J}(\mathbf{q})\mathbf{M}(\mathbf{q})\mathbf{J}(\mathbf{q})^T)^{-1} \tag{3.10a}$$

$$\boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \mathbf{J}(\mathbf{q})^{-T}[\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{M}(\mathbf{q})\mathbf{J}(\mathbf{q})\dot{\mathbf{J}}(\mathbf{q})]\mathbf{J}(\mathbf{q})^{-1} \tag{3.10b}$$

therefore, as proposed in (Ott, 2008) it is possible to express the impedance control law of the system in task space, setting $\Omega_d = \Omega(x)$ in order to avoid the direct feedback of F_{ext}, witch lead us to the following equation

$$\boldsymbol{\Omega}(\mathbf{x})\ddot{\tilde{\mathbf{x}}} + ((\mathbf{D_d} + \boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\tilde{\mathbf{x}}}) + \mathbf{K_d}\tilde{\mathbf{x}} = \mathbf{F_{ext}} \tag{3.11}$$

However, impedance control, also if outperforms pure position control, is not enough to regulate all the aspects of the interaction with environment, especially when a low stiffness setup does not allow to precisely control position error. In some tasks, it would be almost impossible, like in heavy object grasping, or if for example there are obstacles nearby the robot end-effector. That would introduce a constraint on manipulator position, so not a complete positional freedom would cause inevitable errors that affect the performance of our controller, therefore force control is required

3.2 Force control

Pure force control, on the other hand, is not suitable for all situations, since the opposite case can occur, with no constraints on the manipulator position and no force freedom, because there is no possible source of the required reactive force. It appears that pure position and pure force control are dual concepts, and that the historical emphasis on position control is the natural result of applications which involve very little physical contact, but in real-world applications, especially for a teleoperated robot, witch act in unknown environments, both approaches are useful and should be used. Intermediate between the extremes of solid space and free space can be modeled, as done in (Mason, 1981), and are surfaces in configuration space-C-surfaces. Loosely speaking, a C-surface is a task configuration which allows only partial positional freedom. Freedom of motion occurs along C-surface tangents, while freedom of force occurs along C-surface normals. Neither pure position or pure force control is appropriate in this case, but rather a hybrid mode of control, which gives control of effector force along the C-surface normal and control of effector position along the C-surface tangent. In general, without doing any assumption on the structure of the environment in witch our robot is acting, a criteria to switch between force and position control must be found.

The hybrid controller approach distinguishes one or more degrees of freedom as being force-controlled rather than position-controlled. The simplest implementation of this approach is the free joint method. This method is easily understood by considering a task with the property that each force or velocity constraint happens to be aligned with a manipulator joint. In that case, the force axes can be servoed on force and the position axes on position in an independent fashion. Even if the joints are only approximately aligned with the desired constraints, good results may still be obtained. For each constraint, a joint is identified which is most sensitive in the direction of the constraint. The corresponding actuator torque is held constant without regard for any normal forces contributed by the other manipulator joints. Similarly, the position axes are servoed without regard for tangential motions contributed by free joints. The resulting errors limit the usefulness of the system.

Alternatively, priority should be given to one of the two task, while the other would be implemented in its nullspace. However, in our force controller, there is no desired wrench to be achieved that could potentially collide with the position reference, our force compensation it's a constant term, which respect to the impedance controller is only an additional gravity compensation. Therefore no issues of this type are present in our system.

When an object is grasped, we want the slave side to recognize it and autonomously compensate its weight, without the operator to feel any difference. The main issue is to distinguish, using the robot force sensors, between an external force produced by an accidental contact (in witch case no control action is requested) and by the weight of a grasped object.

To achieve this goal, the controller was implemented following a state-machine logic, as shown in figure 3.1

- When the controller is initialized, the force part is disabled (state 0)
- If the sensed external force $\mathbf{F_{ext}}$ has a z component smaller than a certain threshold th_0 (set along negative z-axis direction), a timer counter starts and the weight is saved (contact)
- If the sensed weight remains constant for two seconds (or vary around a reasonably small range), an object has been grasped (grasp). Otherwise the controller is disabled and return to the initial state

If the grasp condition is verified, the current external force vector is considered as the object weight to compensate

$$\mathbf{F_{ob}}(t) = -\mathbf{F_{ext}}(t) \tag{3.12}$$

in order to avoid a jump in the time instant when the object weight is compensated, a simple exponential filter has been implemented so that the actual filtered compensation force is

$$\mathbf{F_c} = s\mathbf{F_{ob}} + (1-s)\mathbf{F_c} \tag{3.13}$$

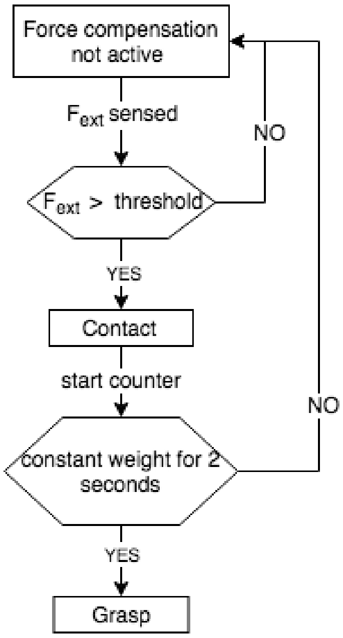

Force compensation not active
F_{ext} sensed
F_{ext} > threshold
NO
YES
Contact
start counter
constant weight for 2 seconds
NO
YES
Grasp

Figure 3.1: force compensation state-machine logic and all the possible phases after that an external force along vertical direction (z axis) is detected

where "s" is the smoothing factor.

When the force controller is active, the threshold must be updated adding the sensed object weight along negative z direction F_{ob}^{z} to the initial value, so that $th_1 = th_0 + F_{ob}^{z}$. By doing this, unwanted external contacts are not considered and additional weight can be carried. If the contact with the object is lost or the task is finished, and the controller is deactivated, also the threshold should be reset to the initial value.

Chapter 4

Stability Analysis

The best and most simple way to assure the stability of a teleoperation system, is definitely through the passivity theory, thanks to its properties is the easiest (and most sure) solution for practical implementations. After a general introduction about passivity, an overview of passivity controllers that have been used in the last years is provided, before describing the approach which have been chosen in this work: the energy tank.

4.1 Passive systems

A system is said to be passive if the total amount of energy that can be extracted from it is upper bounded by the sum of the injected and the initially stored energy (Franken et al., 2011). Any proper combination of passive subsystems will lead to an overall passive system (vd Schaft and Schaft, 1999). Independently from anything else, including the goal of the system or the control type, an energy balance of the telemanipulation system can be composed of the energy present in all of its components. The total energy $H_T(t)$ present in the control system at instant t is

$$H_T(t) = H_M(t) + H_C(t) + H_S(t) \tag{4.1}$$

where $H_M(t)$, $H_S(t)$, and $H_C(t)$ represent, respectively the energy present at the master and the slave side, and in the communication channel, as shown in figure

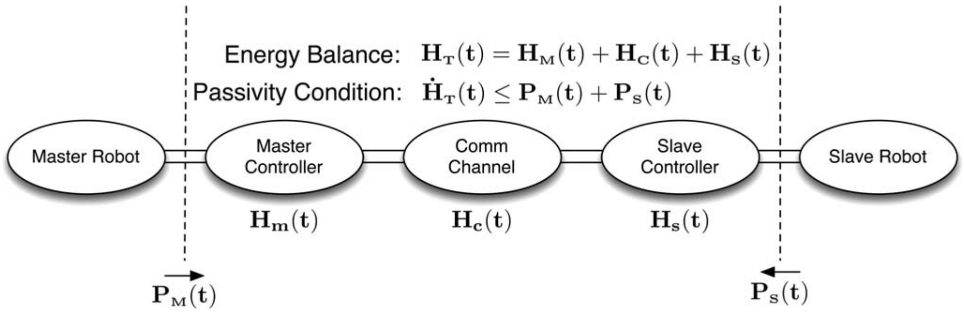


Figure 4.1: Schematic of the energy connections (double lines) of a teleoperation chain. (Picture by Franken et al., 2011)

4.1. Assuming, for the sake of simplicity, that the initially stored energy is zero, the passivity condition of the system is

$$H_T(t) \geq 0. \tag{4.2}$$

Physical energy exchange during operation takes place between the user and the master controller, as well as between the slave system and the external environment. Thus, a condition should also be set also for these external energy exchanges: the only necessary requirement to ensure a passive interconnection of the entire system with the physical world is

$$\dot{H}_T(t) \leq P_M(t) + P_S(t) \tag{4.3}$$

where $P_M(t)$ and $P_S(t)$ are, respectively, the power flowing from the master and slave robot into the master and slave controller, and $\dot{H}_T(t)$ is the rate of change of the energy balance of the system. Equations 4.2 and 4.3 ensures passivity of the system and a passive connection of the system with the physical world, respectively.

4.1.1 Scattering/Wave-Variable-Based Approaches

It is well known that the direct exchange of power variables (velocities and forces) between the master and slave controllers generates "virtual" energy due to the presence of time delays in the communication channel. The scattering and wave variables approaches, developed respectively in (Anderson and Spong, 1989) and (Niemeyer and Slotine, 2004b), apply a coding scheme to the power variables in order to treat the communication channel, in presence of time delay, as a passive element. Once that the communication channel is assured, if the controllers at both the master and slave sides are passive, the complete system is passive according to 4.3, a more detailed description can be found in (Fantuzzi). A wave variable contains both information related to the energy exchange which occurs at that side and the desired behaviour to be displayed by the other side. (Niemeyer and Slotine, 2004b) describes a wave variable as a general "move/push" command to be interpreted by the receiving controller, and the wave sent, describes the response of that device to the previously received command. This, when applied to a bilateral teleoperation system, means that the motion performed by the user and its resulting force feedback are delayed by the time of a complete round-trip of the communication channel. Other transparency-related problems arise due to the nature of the coding process and/or non-idealities in the communication channel (time-varying delay and package loss), e.g., position and force mismatch. Extensions to improve the performance include the use of Smith predictors (Ching and Book, 2006) and the combination of wave variables with the transmission of interaction measurements as discussed by (Tanner and Niemeyer, 2005).

4.1.2 Time Domain Passivity Control

A different solution to the passivity problem include the so called "time domain passivity control" (TDPC) algorithm, originally developed by (Hannaford and Ryu, 2002) for passive interaction with virtual environments, and then applied to bilateral telemanipulation (Ryu et al., 2004). The TDPC approach introduces a Passivity Observer (PO) and a Passivity Controller (PC). This algorithm enforces 4.3 with $H_C = 0$, as no communication channel is considered. Therefore, simultaneous information about the energy flows both at the master and the slave sides is required, and is, as such, not applicable to systems with time delays in the communication channel.

Two extensions have been proposed to extend the TDPC approach to the time-delayed situation. (Artigas et al., 2007) incorporate an energy reference algorithm, and further extensions of this approach also include a passive coupling between the continuous and the discrete domain. Reference algorithm applies a forward and a backward PO, which approximate the energy in the communication channel considering the locally transmitted and received power variables and an estimate of the fixed transmission delay. At each side of the communication channel, a PC maintains passivity according to the PO at that side.

(Ryu and Preusche, 2007) split the energy interaction into an incoming and outgoing energy flow E_{in} and E_{out}. Each side transmits its E_{in} to the other side where passivity of E_{out} with respect to the received value of E_{in} is maintained by a PC. As the transmitted packets symbolize an amount of energy, the passivity of this approach is perfectly robust against time-varying delays and even packet loss in the communication channel. These approaches have merged into a single algorithm, as proposed by the same authors, but are not suitable for the implementation of impedance reflection (IR) algorithms (Tzafestas et al., 2008). Here, the force given in feedback to the user is computed using a local, possibly adaptive, model of the remote environment, while the work of Artigas is centered around the transmission of power variables and cannot accommodate also the transmission of model parameters. Thus, in the algorithm of Ryu et al., the problem is that with an IR algorithm the energy extracted by the user E_out at the master side is likely to occur before E_in actually occurs at the slave side. This means that the PC at the master side will prevent the computed feedback force to be applied to the user as it would force the PO to become negative.

A first approach to use a TDPC algorithm with an IR algorithm was proposed by (Kawashima et al., 2008), where a TDPC structure is used to adapt the locally computed feedback force based on the actual measured, but delayed interaction force to make the system passive. This approach, however, requires precise knowledge about the time delay that is present in the communication channel.

4.1.3 Energy Bounding Algorithm

Another approach that originates from research toward passive interaction with virtual environments, is the energy bounding algorithm (EBA) proposed by (Kim and Ryu, 2010). After that, however, it has been applied to time-delayed bilateral telemanipulation (Seo et al., 2008). The EBA limits the generated "virtual" energy to the dissipated energy by friction at the master and slave sides. With this purpose, it uses models of viscous friction in the devices, which are possibly extended with assumptions about the viscous friction in the user's arm and/or environment. Deviations from the physical friction with respect to the modeled one, can jeopardize stability of the system, for which reason the lower bound of the friction should be selected with a conservative approach. Due to the nature of the derived update rule, the force applied by the control system cannot be adjusted when the devices are perfectly stationary. Finally, it appears, that in the bilateral telemanipulation application it can only work when the force exerted by the slave device is used as the feedback force to the user instead of the measured interaction force between the slave device and the remote environment. This can severely limit the achievable transparency in the presence of time delays and limits the implementable bilateral controller to that specific implementation.

4.1.4 Passive Set-Position Modulation

A recent approach to deal with bilateral telemanipulation is the passive set-position modulation (PSPM) framework that is proposed by Lee et al. (Lee and Huang, 2010), (Lee and Huang, 2008). This approach is centered around a spring-damper controller. The energy dissipated by the "virtual" damper is stored in an energy tank. The jump in spring potential due to the discrete jump of the set position by the control algorithm is limited to the available energy in the tank (a negative jump adds energy to the tank). In the bilateral telemanipulation application, excess energy in the tank is transmitted to the other side or is dissipated. There are several issues related to the working of the PSPM. Most notably, the underlying assumption is that part of the control system can be regarded as continuous time. The set-position signal is a discrete signal, but the position of the device, as used in the servo control loop, is considered as a continuous signal. However, the control system is always sampled even though the update rate for the set position might be much lower than the fundamental sampling rate of the servo control loop. It is well known that the description of a passive element in continuous time can generate energy when implemented on a discrete medium. Therefore, an extended form of Colgate's passivity condition (Colgate et al., 1993) that relates the parameters of the controller, the sample frequency, and the device friction is necessary to guarantee passivity of the system. By the assumption of a constant sampling time ΔTS, the condition described in (Lee and Huang, 2010) becomes

$$Bdev \geq 2BC + \frac{K_C \Delta TS}{2} \tag{4.4}$$

where B_{dev} , B_C , and K_C indicate the physical viscous device friction and the implemented viscous damping and stiffness in the PSPM element, respectively. Equation 4.4 states that the required physical viscous damping has to be at least twice as large as the implemented virtual damping for the system to be guaranteed to be passive. The input to the controller is a set position for the spring. This means that bilateral control algorithms that compute a desired force to be applied to the device(s) require intermediate data processing. This data processing transforms a desired control force into the required set position. This appears elaborate and noise sensitive due to the inherent presence of the velocity estimate, and it requires the set-position signal to be updated at the same frequency as the velocity estimate. This last factor degrades the validity of the assumption that the servo control loop can be considered to be in continuous time. Finally, the PSPM relies on the use of a constant viscous damper to extract energy into the energy tanks. This means that the response will already always be damped, even when there is enough energy in the tank. The excess extracted energy is artificially dissipated by thresholding the level of the energy tank. This constant damping also needs to be taken into account in any higher level control architecture that is connected through intermediate data processing to prevent an over-damped response of the system. With the PSPM, it is, therefore, difficult to separate the design of the controller to display the desired behavior and the manner in which passivity is maintained.

4.1.5 Two layer approach

In a teleoperation system, the slave device needs to display the behavior desired by the user, and the master device needs to accurately provide force feedback about the interaction between the slave device and the remote environment, unless this behavior violates the passivity condition of the telemanipulation system. This shows, as explained in (Franken et al., 2011), that a natural layering in control objectives arises. First, a desired control action needs to be computed so that the master and slave devices display the desired behavior/ information. Then, a "check" is to be performed of how this desired action will influence the energy balance of the system. If passivity will not be violated, it can directly be applied to the physical system, but if passivity is expected to be lost due to the desired control action, it should be modified before application to the physical system. Such an approach allows for the highest possible transparency given that passivity needs to be preserved.

This natural layering can also be directly transformed into a control structure. An algorithm that combines transparency and passivity in the discussed manner would be a two-layer structure, as shown in figure 4.2. The transparency layer contains a control algorithm to display the desired behaviour and obtain transparency. Ideally, this could be any type of bilateral control algorithm, with the requirement of that it computes forces to be applied to the master and slave devices. In order to compute the desired control action $\tau_{TL*}(k)$, access is required to a specific part of the measured interaction data, e.g., forces, positions, and/or velocities, where m and s instead of * indicate the master and the slave, respectively. The passivity layer on the other hand monitors and enforces the energy balance of the system. The benefit of the strict separation into layers is that the optimization of the

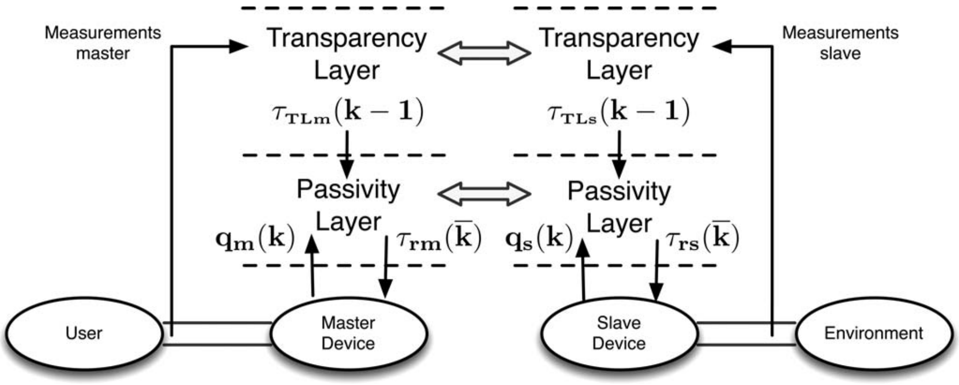

Measurements master
Transparency Layer
Transparency Layer
Measurements slave
$\tau_{\mathbf{TLm}}(\mathbf{k}-\mathbf{1})$
$\tau_{\mathbf{TLs}}(\mathbf{k}-\mathbf{1})$
Passivity Layer
Passivity Layer
$\mathbf{q_m}(\mathbf{k})$
$\tau_{\mathbf{rm}}(\bar{\mathbf{k}})$
$\mathbf{q_s}(\mathbf{k})$
$\tau_{\mathbf{rs}}(\bar{\mathbf{k}})$
User
Master Device
Slave Device
Environment

Figure 4.2: two layer teleoperation system scheme (Picture by Franken et al., 2011)

strategy used to ensure optimal transparency does not depend on the strategy used to ensure passivity and vice versa. As passivity does not have to be considered in the design of the transparency layer, the whole range of control techniques which are nonpassive, e.g., most filtering techniques, can be applied without problems. In addition, due to this strict separation into layers two two-way communication channels between the master and slave systems can be defined. One channel is used to communicate energy-exchange-related information between the passivity layers and the second channel to communicate information related to the desired behaviour to be displayed by the devices between the transparency layers. This means that no (de)coding process is required as with wave-variable-based approaches. Furthermore, no restrictions are necessary on the information exchanged between the transparency layers.

4.2 Port-Hamiltonian systems

However, the two layer approach loose its convenience in systems in witch force feedback is not required, and unilateral systems in general. Furthermore, the generality given by the black-box approach regarding the passivity control, is paid in terms of occupation of the communication channel, in witch informations about both the commands given by the user and the energy flows of the system are constantly present.

Therefore, an alternative should be explored, a valid solution is the "unified approach" that overcomes these limits, without loosing generality of application. In the firs place, a closer study of the system internal dynamics should be done.
Consider, as described in (Cervera et al., 2007), a lumped-parameter physical system given by a power-conserving interconnection defined by a constant Dirac structure D, and a number of energy-storing elements with total vector of energy-variables x. For simplicity we assume that the energy-variables are living in a linear space X, although everything can be generalized to the case of manifolds. The constitutive relations of the energy-storing elements are specified by their individual stored energies, leading to a total energy (or Hamiltonian) H(x). The space of flow variables for the Dirac structure D is split into $X \times F$ with $f_x \in X$ the flows corresponding to the energy-storing elements, and $f \in F$ denoting the remaining flows (corresponding to dissipative elements and external ports). Correspondingly, the space of effort variables is split as $X^* \times F^*$, with $e_x \in X$ the efforts corresponding to the energy-storing elements and $e2F$ the remaining efforts. Thus $D \subset X \times X^* \times F \times F^*$. On the other hand, the vector of flows of the energy storing elements is given by $\dot{x}$, and the vector of efforts is given by $\frac{\delta H}{\delta x}(x)$. We will write both vectors throughout as column vectors; in particular, $\frac{\delta H}{\delta x}(x)$ is the column vector with i-th component given by $\frac{\delta H}{\delta x_i}(x)$. Indeed, the energy storing elements satisfy the total energy balance

$$\frac{dH}{dT}(x(t)) = \frac{\delta^T H}{\delta x}(x(t))\dot{x}(t) \tag{4.5}$$

The flows and efforts of the energy-storing elements are interconnected by setting $f_x = -\dot{x}$ (the minus sign is included to have a consistent power flow direction) and $e_x = \frac{\delta H}{\delta x}(x)$. By substitution of the interconnection constraints into the specification of the Dirac structure D, that is, $(f_x, e_x, f, e) \in D$, this leads to the dynamical system

$$(-\dot{x}(t), \frac{\delta H}{\delta x(t)}, f(t), e(t)) \in D \tag{4.6}$$

called a port-Hamiltonian system.
Because of the power conserving property of Dirac structures we immediately obtain the power balance

$$\frac{dH(x(t))}{dt} = \frac{\delta^T H}{\delta x}(x(t))\dot{x}(t) = - < e_x(t)|f_x(t) >=< e(t)|f(t) > \tag{4.7}$$

expressing that the increase of of internal energy of the port-Hamiltonian system is equal to the externally supplied power. Equational representations of the port-Hamiltonian system 4.6 are obtained by choosing a specific representation of the Dirac structure D. For example, if D is given in matrix kernel representation

$$D = (f_x, e_x, f, e) \in X \times X^* \times F \times F^* | F_x f_x + E_x e_x + F_f + E_e = 0 \tag{4.8}$$

An important special case of port-Hamiltonian systems is the class of input-state-output port-Hamiltonian sys- tems, where there are no algebraic constraints on the state variables, and the flow and effort variables f and e are split into power-conjugate input-output pairs (u, y):

$$\begin{cases} \dot{x} = J(x)\frac{\delta H}{\delta x}(x) + g(x)u \\ y = g^T(x)\frac{\delta H}{\delta x}(x) \end{cases} \tag{4.9}$$

where $x \in X$ and the matrix J(x) is skew-symmetric, that is $J(x) = -J^T(x)$.

The Dirac structure of the system is given by the graph of the skew-symmetric map

$$\begin{bmatrix} -J(x) & -g(x) \\ g^T(x) & 0 \end{bmatrix} \tag{4.10}$$

Thus, the port-Hamiltonian framework is a generalization of standard Hamiltonian mechanics, where energetic characteristics and power exchange between subsystems are clearly identified. All physical systems, even multi-domain, can be represented using the port-Hamiltonian formalism.

Matrices $J(x) = -J(x)^T$ and $R(x) \geq 0$ represent the internal energetic interconnections and the dissipation of the port-Hamiltonian system, respectively, and $g(x)$ is the input matrix. The input u and the output y are dual variables and their product is (generalized) power. The pair (u, y) is called power port and is the means by which the system can energetically interact with the external world. The product $u^T y$ represents the power exchanged by the system with the external world. Therefore, to prove the stability of the system, it's a good choice to rely on a passivity-based control. An important property of passive systems is that is that the proper interconnection of passive subsystems leads to an overall passive system, so that in the end it is sufficient to prove that all its subsystems are passive w.r.t. their input-output ports $[\dot{q}, -\tau]$.

Assuming that both the external environment and the master are passive, it must be proven that the controller on the slave side is passive with respect to the pair $(\dot{x}, -F_{ext})$.
To study the energy balance of our system, it is convenient to use the storage function, witch can be defined as a positive semidefinite Lyapunov function $S : \mathbb{R}^m \rightarrow \mathbb{R}_+$, equal (in our case) to

$$S = \frac{1}{2}\dot{\tilde{\mathbf{x}}}^T \boldsymbol{\Omega}(\mathbf{x})\dot{\tilde{\mathbf{x}}} + \frac{1}{2}\tilde{\mathbf{x}}^T K_d \tilde{\mathbf{x}} + m_{ob} g h + \frac{1}{2}\dot{\mathbf{x}}^T m_{ob} \dot{\mathbf{x}} \tag{4.11}$$

in witch there are the contributions of potential and kinetic energy deriving from both the impedance control and the object compensation force, m_{ob}, h and g are respectively the object mass, its height respect to the ground, and the scalar representation of the gravity acceleration. The varying stiffness K_d references, as illustrated in chapter 2, would in theory introduce an additional non passive term, which has been separately analyzed in (Ferraguti et al., 2013) using the same energy tank approach that will be illustrated in this work. But, a stability analysis involving both the varying stiffness and the force control non-passive terms has not been studied yet in literature and doesn't have a mathematical coverage.

Therefore constant stiffness and damping are considered here, and in the following equations, and the only non passive-term is caused by the force control action, as studied in (Schindlbeck and Haddadin, 2015). The time derivative of our storage function is:

$$\dot{S} = \dot{\tilde{\mathbf{x}}}^T K_d \tilde{\mathbf{x}} + \frac{1}{2}\dot{\tilde{\mathbf{x}}}^T \dot{\boldsymbol{\Omega}}(\mathbf{x})\dot{\tilde{\mathbf{x}}} + \dot{\tilde{\mathbf{x}}}^T \boldsymbol{\Omega}(\mathbf{x})\ddot{\tilde{\mathbf{x}}} + F_{ob}^z \dot{z} + \dot{\mathbf{x}}^T m_{ob} \ddot{\mathbf{x}} \tag{4.12}$$

The last term denote the effect of the object inertia in the dynamics of the system, witch produce in the end effector a force equal to $m_{ob}\ddot{\mathbf{x}}$. This force is seen by the robot's end-effector as an external force, therefore it is convenient to rewrite the system equation (3.11) considering the following external force vector

$$\tilde{\mathbf{F}}_{\mathbf{ext}} = m_{ob}\ddot{\mathbf{x}} + \mathbf{F}_{\mathbf{ext}} \tag{4.13}$$

now , passivity should be studied w.r.t the pair $(\dot{\mathbf{x}}, -\tilde{\mathbf{F}}_{\mathbf{ext}})$, leading to

$$\dot{S} = \dot{\tilde{\mathbf{x}}}^T K_d \tilde{\mathbf{x}} + \frac{1}{2}\dot{\tilde{\mathbf{x}}}^T \dot{\boldsymbol{\Omega}}(\mathbf{x})\dot{\tilde{\mathbf{x}}} + \dot{\tilde{\mathbf{x}}}^T \boldsymbol{\Omega}(\mathbf{x})\ddot{\tilde{\mathbf{x}}} + F_{ob}^z \dot{z} \tag{4.14a}$$

$$\begin{aligned} &= \dot{\tilde{\mathbf{x}}}^T K_d \tilde{\mathbf{x}} + \frac{1}{2}\dot{\tilde{\mathbf{x}}}^T \dot{\boldsymbol{\Omega}}(\mathbf{x})\dot{\tilde{\mathbf{x}}} + \dot{\tilde{\mathbf{x}}}^T (-\mu(x,\dot{x})\dot{\tilde{\mathbf{x}}} - D_d \dot{\tilde{\mathbf{x}}} - K_d \tilde{\mathbf{x}} \\ &+ \tilde{\mathbf{F}}_{\mathbf{ext}}) + F_{ob}^z \dot{z} \end{aligned} \tag{4.14b}$$

considering that $\frac{1}{2}\dot{\tilde{\mathbf{x}}}^T(\dot{\boldsymbol{\Omega}}(\mathbf{x}) - 2\mu(x,\dot{x}))\dot{\tilde{\mathbf{x}}} = 0$ we have

$$= \dot{\tilde{\mathbf{x}}}^T \tilde{\mathbf{F}}_{\mathbf{ext}} - \dot{\tilde{\mathbf{x}}}^T \mathbf{D_d} \dot{\tilde{\mathbf{x}}} + F_{ob}^z \dot{z} \tag{4.14c}$$

here, the last term show a variation of energy deriving from the end-effector velocity in the vertical axis direction $(\dot{\mathbf{z}})$. While the term $\dot{\tilde{\mathbf{x}}}^T D_d \dot{\tilde{\mathbf{x}}}$ is positive, the sign of $F_{ob}^z \dot{z}$ is not known, thus the passivity condition $\dot{S} \leq \dot{\tilde{\mathbf{x}}}^T \tilde{\mathbf{F}}_{\mathbf{ext}}$ can be potentially violated.

4.3 Energy Tank

In order to solve this problem, an energy tank based passivity controller was added to the main one. As explained in (Ferraguti et al., 2013), basically the energy tank can be seen as a virtual reservoir filled with the energy dissipated by the system, in order to keep track of it and use it as a sort of *passivity margin* to implement some non-passive actions. The dissipated power, expressed in the above equation by the term $\dot{\tilde{\mathbf{x}}}^T D_d \dot{\tilde{\mathbf{x}}}$, is used to store energy in the tank, whose state (denoted by x_t) is defined as

$$\dot{x}_t = \frac{\beta}{x_t}(\dot{\tilde{\mathbf{x}}}^T D_d \dot{\tilde{\mathbf{x}}}) + u_t \tag{4.15}$$

where β is a parameter set to stop the energy flow directed to the tank when a certain upper limit $\bar{T}$ is reached, otherwise also if the system is supposed to be passive, in practice, non passive actions may be accidentally implemented; it is defined as

$$\beta = \begin{cases} 1 & \text{if } T \leq \bar{T} \\ 0 & \text{else} \end{cases} \tag{4.16}$$

The term u_t allow us to control the energy exchange from the tank to the main controller, and is set as $u_t = -\omega^T \dot{x}$, where $\omega(F_{ext}, t)$ is the control input, in witch is considered the error between the external force and the sensed object weight $\mathbf{F_{ob}}$:

$$\mathbf{F_{err}} = \tilde{\mathbf{F}}_{\mathbf{ext}} + \mathbf{F_{ob}} \tag{4.17}$$

Considering that the force controller is continuously compensating the object weight $\mathbf{F_{ob}}$, besides the passivity control action, this provide us also a more robust classification between accidental disturbances and unsafe contact loss and consequently a more compliant behaviour.
The resulting control variable is

$$\omega(F_{ext}, t) = \frac{\alpha}{x_t}(\mathbf{F_{err}}) \tag{4.18}$$

where α is defined as

$$\alpha = \begin{cases} 1 & \text{if } T \geq \varepsilon \\ 0 & \text{else} \end{cases} \tag{4.19}$$

and is used to stop the energy flow from the tank when its lower limit ε is reached. The resulting total energy stored in the tank is

$$T(x_t) = \frac{1}{2}x_t^2 \tag{4.20}$$

where x_t must be strictly greater than zero, in order to avoid singularities. After that the passivity control is applied, we can rewrite the storage function as

$$S = \frac{1}{2}\dot{\tilde{x}}^T M(q)\dot{\tilde{x}} + \frac{1}{2}\tilde{x}^T K_d \tilde{x} + \frac{1}{2}x_t^2 \tag{4.21}$$

and the time derivative $\dot{S}$ can be computed, following the same procedure done in (4.14), as

$$\begin{aligned}\dot{S} &= \dot{\tilde{x}}^T \tilde{F}_{ext} - \dot{\tilde{x}}^T D_d \dot{\tilde{x}} + \dot{\tilde{x}}^T \omega x_t + \beta(\dot{\tilde{x}}^T D_d \dot{\tilde{x}}) - x_t \omega^T \dot{\tilde{x}} \\ &= \dot{\tilde{x}}^T \tilde{F}_{ext} - \dot{\tilde{x}}^T D_d \dot{\tilde{x}} + \beta(\dot{\tilde{x}}^T D_d \dot{\tilde{x}})\end{aligned} \tag{4.22}$$

Due to (4.16), we have that

$$-\dot{\tilde{x}}^T D_d \dot{\tilde{x}} + \beta(\dot{\tilde{x}}^T D_d \dot{\tilde{x}}) \leq 0$$

and, therefore,

$$\dot{S} \leq \dot{\tilde{x}}^T \tilde{F}_{ext} \tag{4.23}$$

witch leads to the passivity condition.

Finally, let's see how the energy-tank impact on our controller described in chapter 3.

4.3.1 Contact Recognition

As mentioned, to distinguish between accidental contacts with the external environment from object-loss, is a critic issue for these kind of applications. In fact is important to maintain as far as possible a compliant behavior when an unknown contact occurs, thus only the low-stiffness impedance control should react to these interactions. Also if excluding the compensation force from the external force vector, helps to go in this direction, is not enough to completely avoid the tank discharge, as the tank is unloaded by this term also when a "safe" contact occurs, leading to an unsafe switching behavior.

To overcome this limit was made the assumption, repeatedly seen in literature, that the previously called "safe contacts" are characterized by low velocity, while a sudden object-loss would have high velocity. With this hypothesis, an additional filtering action was implemented that, below an certain threshold v_{up} (experimentally determined) reduce the contribution of the force error F_{err} to the tank dynamics when the velocity decreases, being

$$\tilde{\mathbf{F}}_{\mathbf{err}} = \begin{cases} \mathbf{F}_{\mathbf{err}} & \text{if } \dot{\tilde{\mathbf{x}}} \geq v_{up} \\ C\mathbf{F}_{\mathbf{err}} & \text{else} \end{cases} \tag{4.24}$$

where the coefficient $C \in \mathbb{R}$ linearly varies from 0 to 1.

This is the tank control input used in the experiments, the practical effect of the energy-tank on the rest of our control action is that the non-passive term of our system, is expressed in the following equation

$$\mathbf{F}_{\mathbf{tot}} = \mathbf{F}_{\mathbf{i}} + \alpha \mathbf{F}_{\mathbf{comp}} \tag{4.25}$$

the compensation force, is present only if the tank level T is greater than its lower limit, when a non-passive action occurs, and the tank is emptied, α becomes equal to zero and the control action of $\mathbf{F}_{\mathbf{comp}}$ is consequently excluded.

Chapter 5

Simulation

Both simulation and experimental setup of our work is based on the XBotCore platform, described in the next section. This ROS compatible software platform allows to test our controller, implemented in a plugin (written in C++), both on the real robot and on its software model, with all its kinematic and dynamics properties. Therefore the communication protocols between XBotCore and Kuka Lightweigh Arm (the robot chosen for our experiments) is described. All the simulation were done with the software Gazebo, witch is really reliable and easy to use for robotic algorithms testing, as will be said in the second section. Finally a test of the controller performances was done with Gazebo, simulating a contact loss.

5.1 XBotCore

The software architecture of the robot relies upon XBotCore (Muratore et al., 2017) (*Cross-Bot-Core*) – a recently developed open-source[1] and light-weight Real-Time (RT) platform for robotics, designed to be both an RT robot control framework and a software middleware. The design of a software platform that lies at the foundations of a complex system, such as a robotic system, is the most crucial phase in the software development process. XBotCore was designed to be both an RT control system and an easy-to-use, flexible and reusable middleware for RT or non RT tasks. XBotCore design goals are the following:

- Hard RT control system: it must perform computation within predictable timing constraints

- 1 kHz control frequency: robotics applications may require high frequency control loops, e.g. RT Pattern Generator for Biped Walking or haptics applications

- Cross-Robot compatibility: it has to work with any kind of EtherCAT-based robot, without any code modification. It is crucial to be able to reuse the software platform with different robots, or different part of the same robot

- External Framework integration: it has to be possible to use XBotCore as a middleware for any kind of external software framework (RT or non RT)

[1] https://github.com/ADVRHumanoids/XBotCore

- Plug-in Architecture: users and third parties should be able to develop their own modules. In a robotic system platform we need an highly expandable software structure

- Light-weight: we don't want too many dependencies on other libraries, it should be easy to install and set up. Moreover we expect to run XBotCore on embedded PCs with low performance requirements in terms of memory and CPU. We therefore need a small footprint and to avoid high CPU usage

- Simplicity: it must be simple. Complex systems may have unneeded and over-engineered features. For robotics application we need the full control over the software platform. KISS ("Keep It Simple, Stupid") principle is essential; simplicity is a key goal in XBotCore design and unnecessary complexity should be avoided

- Flexibility: XBotCore has to be easily modified or extended in order to be used in applications or environments other than those for which it was specifically designed

- Open-source: open-source software provides transparency in the software implementation since any developer can study and modify the code, eventually to the benefit of the robotics community. Moreover a flexible license is essential for the free distribution of XBotCore in other open-source projects

XBotCore satisfies RT requirements ensuring, as said, 1 kHz hard RT control loop even in complex multi-DOF systems, moreover it provides a simple and easy-to-use middleware Application Programming Interface (API), for both RT and non-RT external control frameworks. This API is completely flexible with respect to the framework a user wants to utilize. It is also possible to reuse the code written using XBotCore API with different robots (cross-robot feature) thanks to the Robot Hardware Abstraction Layer (R-HAL) introduced in (Rigano et al., 2018).

The XBotInterface library constitutes the high-level API used inside XBot Core for the purposes of robot control at the joint level and kinematic/dynamic modeling. As a matter of fact, it provides a framework-independent abstraction layer for any robotic platform, thus making it suitable for use also outside the XBotCore framework. Special attention was taken in order to make it real-time safe, so that it can used inside RT frameworks like OROCOS and XBotCore itself. Moreover, it is fully integrated with the state-of-the-art Eigen3 library for linear algebra, that provides good performance and tuning capabilities for use inside a real-time loop.

The XBotInterface library it can be used for

- sense a robot joint states in terms of position, velocity, torque, impedance, ...

- read sensors that are commonly used for robot control (IMU, force-torque sensors, ...)

- send commands to a robot in terms of desired position, velocity, torque, impedance

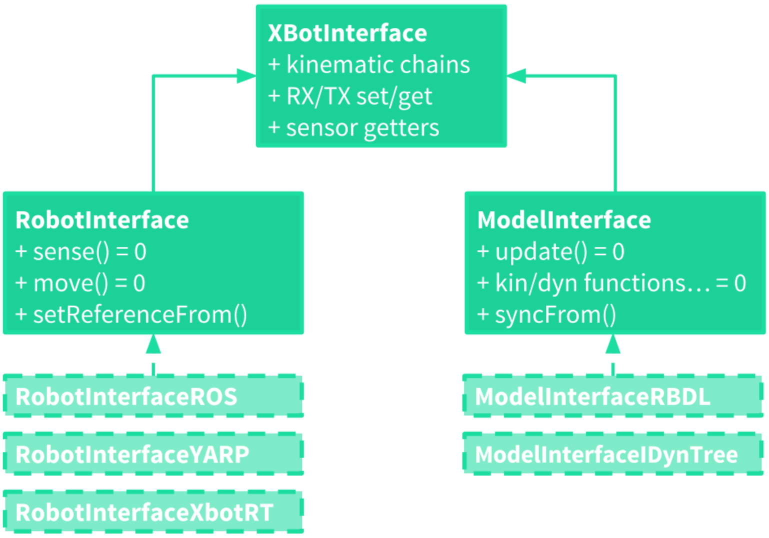


Figure 5.1: XBotInterface class hierarchy structure

- compute a robot kinematics (FK, jacobians, ...) and dynamics (gravity compensation, inverse dynamics, inertia matrix, ...)

Some useful additional functionalities are also offered, like:

- a lightweight console logging utility, which is completely real-time safe on Xenomai systems
- a mat file logging utility, completely real-time safe as well

The XBotInterface library is designed so as to be as generic and configurable as possible:

- it can be used with any robot which is described in terms of a URDF file plus an SRDF file for semantic description
- it can be used inside any robotic framework/middleware. Out-of-the-box implementations are available for ROS, YARP, OROCOS, and of course the XBotCore RT layer. This means that code written using XBotInterface will be very easily portable among all these frameworks, and also new ones that may come in the future!
- it can be used with any kinematics/dynamics library as a back-end. Out-of-the-box implementation are available for RBDL and iDynTree.

Moreover, the actual implementation that is used for communication with the robot and kinematics/dynamics can be selected at runtime, i.e. without the need to recompile any code.

The software platform provides a set of open-source plugins for the Gazebo simulator[2], which allows for a transparent code portability from simulation to the

[2] https://github.com/ADVRHumanoids/GazeboXBotPlugin

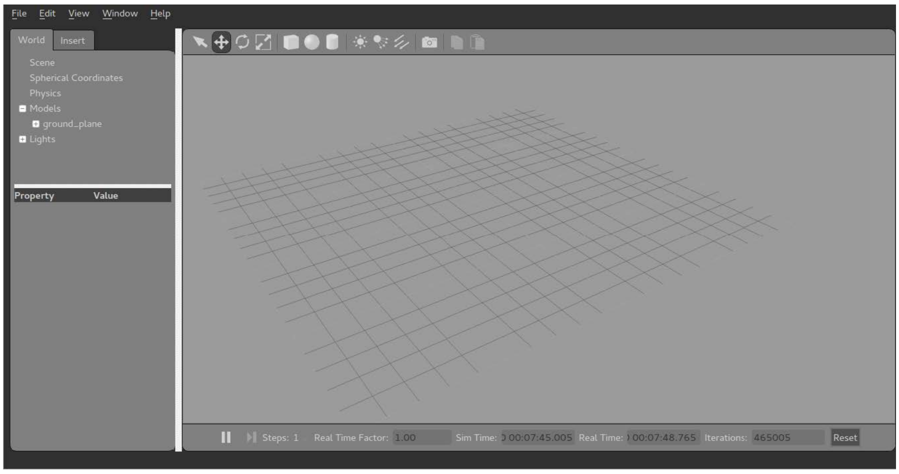


Figure 5.2: overwiew of Gazebo main window

real robotic hardware. The R-HAL permits to seamlessly program and control any robotic platform providing a relatively uniform abstraction that hides the specifics of the underlying hardware. Consequently, this allows the robot software developers to easily transfer/reuse their code on different robots. All the R-HAL implementations are built as a shared library loaded at runtime according to what specified in a configuration file. In particular the R-HAL implementation for the KUKA IIWA has been provided relying on the FRI (Fast Robot Interface) to transfer data between the robot controller and the external computer. It is important to mention that the control was done in non-RT mode because the FRI API communication provided by KUKA is intrinsically not RT safe. However a porting of the KUKA API to the RT domain could be possible with no particular issues ensuring always no changes in the XBotCore plugin developed.The control behaviour is implemented using one different XBot plugin. One of them is used to realize the Cartesian impedance controller.

5.2 Gazebo

Robot simulation is an essential tool in every roboticist toolbox. A well designed simulator makes it possible to rapidly test algorithms, design robots, and perform regression testing using realistic scenarios. Gazebo offers the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. It sports a robust physics engine, high-quality graphics, and convenient programmatic and graphical interfaces. Its main features are:

- Dynamics Simulation: By default Gazebo is compiled with support for ODE as physics engine, but it also has access to other high-performance physics engines including Bullet, Simbody, and DART;

- Advanced 3D Graphics: Using OGRE, Gazebo provides realistic rendering of environments including high-quality lighting, shadows, and textures. OGRE (Object-Oriented Graphics Rendering Engine) is a scene-oriented, flexible

3D engine written in C++ designed to make it easier and more intuitive for developers to produce applications utilizing hardware-accelerated 3D graphics;

- Sensors and Noise: Gazebo offers the possibility to generate sensor data, optionally with noise, from laser range finders, 2D/3D cameras, Kinect style sensors, contact sensors, force-torque sensors, and more;
- Plugins: Robot-independent Gazebo plugins for sensors, motors and dynamic reconfigurable components are available within the plugin package. It is also possible to develop custom plugins for robot, sensor, and environmental control. Plugins provide direct access to Gazebo's API;
- Robot Models: Many robots are provided including PR2, Pioneer2 DX, iRobot Create, and TurtleBot. It is possible to build your own using SDF and to contribute your model to Gazebo's online-database to benefits you and every other user of Gazebo;
- TCP/IP Transport: It allows to run simulations on remote servers, and interface to Gazebo through socket-based message passing using Google Protobufs;
- Cloud Simulation: There is the possibility to use CloudSim, a framework for modeling and simulation of cloud computing infrastructures and services, to run Gazebo on Amazon, Softlayer, or your own OpenStack instance;
- Command Line Tools: Extensive command line tools facilitate simulation introspection and control.

While similar to game engines, Gazebo offers physics simulation at a much higher degree of fidelity, a suite of sensors, and interfaces for both users and programs. Typical uses of Gazebo include:

- testing robotics algorithms;
- designing robots;
- performing regression testing with realistic scenarios.

Recent Gazebo models are described using a new format called the Simulation Description Format (SDF) that was created for use in Gazebo to solve the shortcomings of the old URDF format. SDF is a complete description for everything from the world level down to the robot level. It is scalable, and makes it easy to add and modify elements. The SDF format is itself described using XML, which facilitates a simple upgrade tool to migrate old versions to new versions. It is also self-descriptive [74]. The main element of an SDF file is the world element, which encapsulates an entire world description: models, scene, physics, joints and plugins. The model element is the one used to define a complete robot. The description of a model is given by the definition of a set of links, i.e., a collection of Collision and Visual objects. Collision Objects is a geometry that defines a colliding surface, while Visual Objects is a geometry that defines visual representation such as meshes.

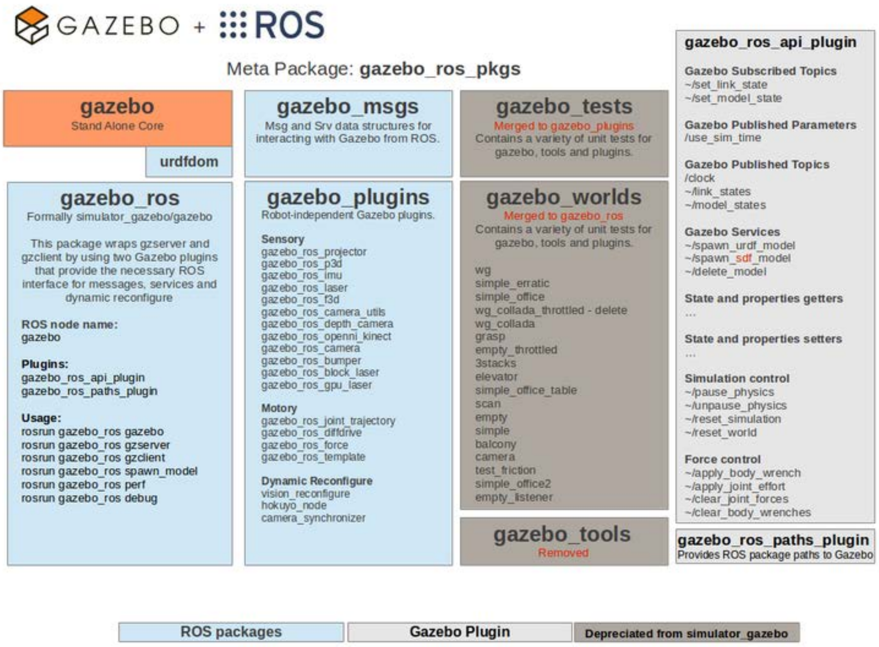


Figure 5.3: Gazebo-Ros interface

Gazebo offers the possibility to modify the main elements, i.e., World, Model and Sensor, taking advantage of plugins. A plugin is a C++ library that is loaded by Gazebo at runtime. It has access to the relative Gazebo's API, which allows a plugin to perform a wide variety of tasks including moving objects, adding/removing objects, and accessing sensor data. In Figure 3.5 is shown the Gazebo element hierarchy. As of Gazebo 1.9 and ROS Hydro, Gazebo no longer has any direct ROS dependencies and is now installed as an Ubuntu stand-alone package.

Historically using Gazebo with ROS required a specific version of Gazebo to be built with the legacy *'simulator_gazebo'* stack. To achieve ROS integration with stand-alone Gazebo, a new set of ROS packages named *gazebo_ros_pkgs* has been created to provide wrappers around the stand-alone Gazebo. They provide the necessary interfaces to simulate a robot in Gazebo using ROS messages, services and dynamic reconfigure.

5.3 Contact loss simulation

Using the described software architectures, a simulation of a contact loss situation was done in order to prove the validity of the energy-tank passivity controller.
In order to do that, a model of the Kuka LWR arm is loaded in the Gazebo simulator, and it is moved firstly applying small forces impulses along y direction to simulate safe movements, and then a huge force in z axis direction to simulate an object grasping and a contact loss. The fist movements are supposed to load the energy tank, while the final movement should immediately unload it.
In a XBotCore non real-time plugin were implemented:

- the storage function of the system
- the tank dynamics described in 4
- the overall force/impedance controller described in chapter 3

and their variation with and without the tank are analyzed. The purpose of computing the storage function is to prove that the tank is actually filled from the damping term when the robot is moving and the system is passive (as described in chapter 4), while is emptied when a non passive situation occurs. The corresponding tank levels are measured to see if the dynamics worked correctly. Finally the force commanded to the robot was analyzed to see if the tank detached the compensation term correctly after the contact loss.

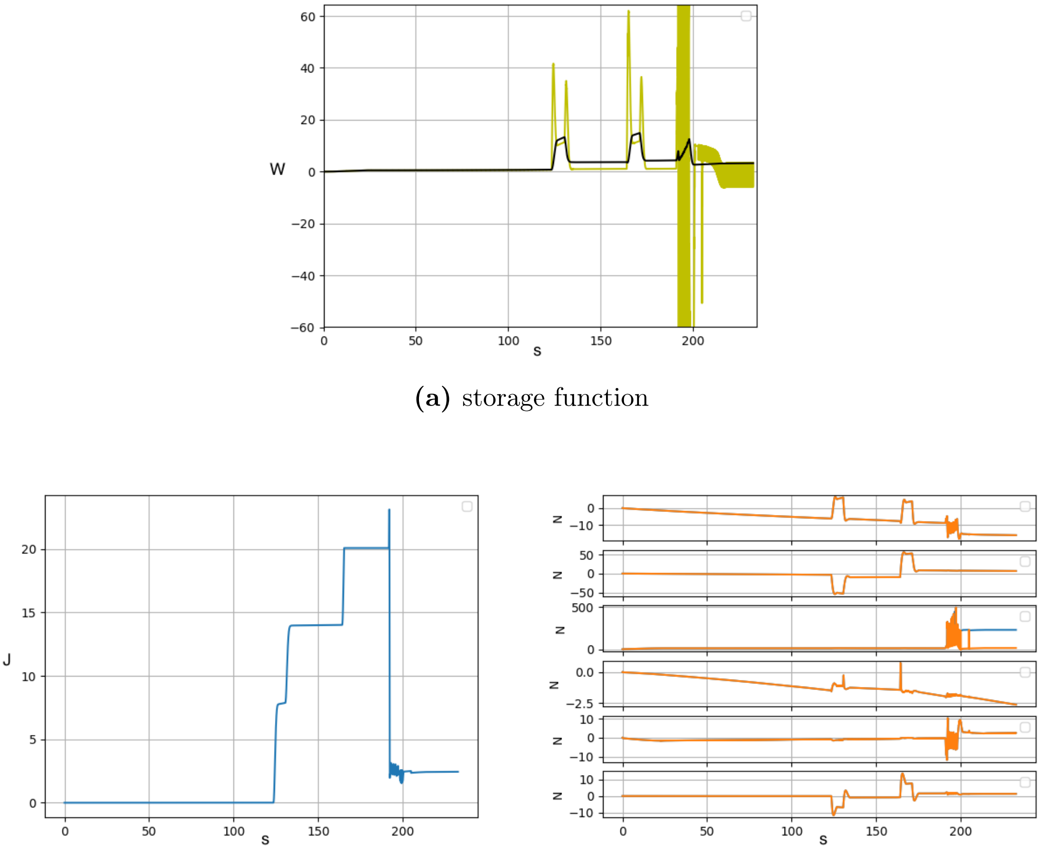
Stiffness and damping were set as diagonal matrices with elements respectively equal to $K_d = diag\{200, 200, 200, 15, 15, 15\}[N/m]$, and $d_i = 2\xi_i k_i$, with $\xi_i = 0,7$. For having a better behaviour in an experimental setup, the velocity signal should be filtered as done with force reference in (6), so that the filtered velocity $\dot{\tilde{x}}_f$ is

$$\dot{\tilde{x}}_f = s\dot{\tilde{x}} + (1-s)\dot{\tilde{x}}_f$$

.
To implement the compensation of a grasped object in a virtual environment, only the force exchanges in the vertical direction were considered, assuming that the object center of mass will coincide with the end-effector one. So negative vertical forces were applied in the end-effector in order to simulate the gravity force F_{ob}, while inertia terms were neglected. When the sensed external force overcomes the value of the threshold, the force controller is immediately swhiched on, without time delays, the state-machine approach described in chapter 3 was instead used in real-experiments.

Figure 5.4 shows how the tank is filled due to the small movements of the robot along y axis, until the upper limit of 20 [J] is reached. Then, a force of -85 [N] in z direction was applied for 2 seconds, to simulate an heavy object grasping and a sudden release that leads to an unstable behaviour this force is voluntarily high in order to highlight the unstable behaviour of the controller (without the passivity part in this case).

In 5.4a the storage function of the controller without the tank(in yellow) shows the highly unstable behavior of the system, while the storage function of the controller with the tank (in black) is always greater than zero (that, as said, proves the passivity of the system). In 5.4b are shown the corresponding tank levels, and in 5.4c the commanded force with and without the tank, as we can see in the third component, corresponding to the z axis, the commanded force with the tank (in orange) after the contact loss doesn't follow the unstable behavior of the plain controller (in blue).



(a) storage function

(b) tank

(c) commanded force

Figure 5.4: resulting system storage functions (a), tank level (b) and commanded forces (c) in all phases of the simulation

This happens because after the end of the applied force impulse, the energy is drained from the tank, and when the energy of the tank reaches its lower limit of 2 [J], the force controller is successfully switched off.

Chapter 6

Experimental Results

To prove the validity of our work, two experiments were done: in the first one, the passivity controller implemented in the robot was tested separately, then the complete teleoperation framework was involved. In particular, using Kuka LWR robotic arm, were executed the following two tasks:

- Lift and sudden contact-loss of a weight, in movement, setting different stiffnesses.
- Grasping task using the teleimpedance with commanded low stiffness, in which Pisa/IIT SoftHand (Della Santina et al., 2015) was used as end-effector

6.1 Passivity controller performances in motion

In the fist task the energy-tank passivity controller of the slave robot was tested while executing an eight-shaped trajectory along x and y axis, with the purpose to evaluate its performances in terms of maximum force applied by the end-effector and its position error after a contact loss. These two measures were analyzed when varying stiffness and applied payload. The final objective of this experiment is to demonstrate how the contribution of the proposed stability controller increase when the stiffness decreases. The position reference was set, in the world frame, respectively to

$$\begin{aligned} x &= A \sin(2t) \\ y &= 2A \sin(t) \end{aligned} \tag{6.1}$$

in which A is the amplitude and t the time of the sinusoidal trajectory.

Three trials were done, with stiffness references set to $K_d = 100, 200$ and 300; for each K_d value, weights of 3 and $4Kg$ were placed at the end effector. Force and position measurements after an unexpected contact loss with and without the tank were compared.

Experiment results setting stiffness $K_d = 100$

In figure 6.1, are represented the sensed external force (in purple) and the consequent variation in the total force reference computed by the controller (in blue) and in both its components: the impedance controller force (yellow) and the compensation force (in orange). Figures 6.1a and 6.1b present the results obtained when applying $3Kg$, respectively with the energy tank on (in 6.1a) and off (6.1b). The same curves are shown in 6.1c and 6.1d with the $4Kg$ load.

After the object detection the weight of the object is computed (as explained in chapter 3), and the total controlling action, equal to the one of the impedance controller, goes rapidly to contrast the sensed weight force ($-30N$ in the first case and $-40N$ in the second). Then, after two seconds, the weight is computed and the compensation force term is active, therefore the weight of the object applied in the end-effector is not sensed anymore from the impedance controller, that returns to its original set-point equal to 0. The total force now oscillates around the compensation force, as a result of the errors caused by the object inertia when the robot is moving, that have to be continuously corrected by the impedance control action.

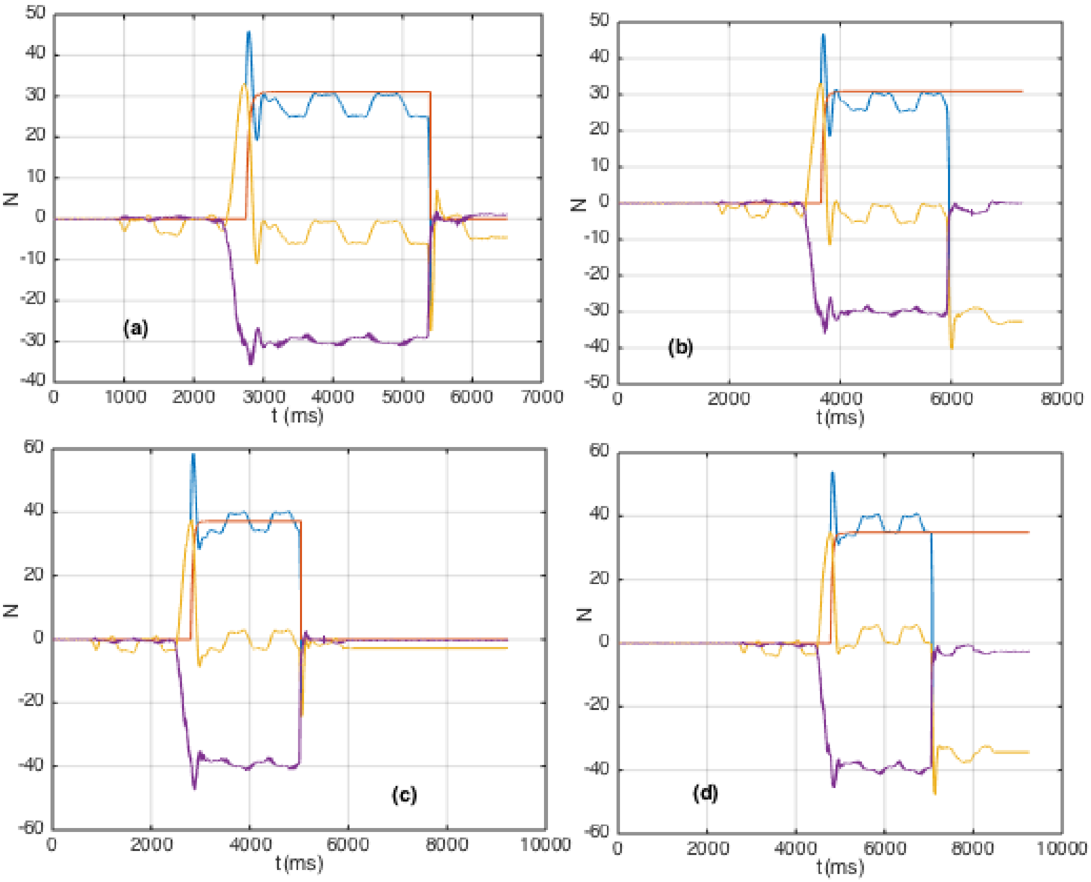


Figure 6.1: all the forces contribution measured applying $3Kg$ (with tank in a, without in b) and $4Kg$ (with tank in c, without in d) .

When the contact is lost, if the tank is deactivated, the force compensation term remains active, and the impedance controller will stabilize around the opposite of its value ($-30N$ or $-40N$) in order to contrast its action, as shown in figures 6.1b and 6.1d. Due to the low value of the commanded stiffness, the impedance control action isn't strong enough to return to the desired position reference, so the error is constantly equal to this value. If the tank is active, the force compensation term is immediately detached (figures 6.1a and 6.1c); thus the impedance controller is able to return to its set point, not having to continuously compensate any error.

This difference in behavior with and without the energy-tank, is evident also from the position error signals, shown in figure 6.2. Here, it is represented the displacement from the position reference during all the task execution, with the energy tank contribution active (in blue) and without it (in orange), for the $3Kg$ case in 6.2a and for the $4Kg$ case in 6.2b. When the tank is off, in both cases, the position error after the contact loss is constant. Instead, when it is active, the sudden reaction of the controller limits the error below $20cm$, while it would have been nearly the double (detailed results will be provided in the table at the end of this section).

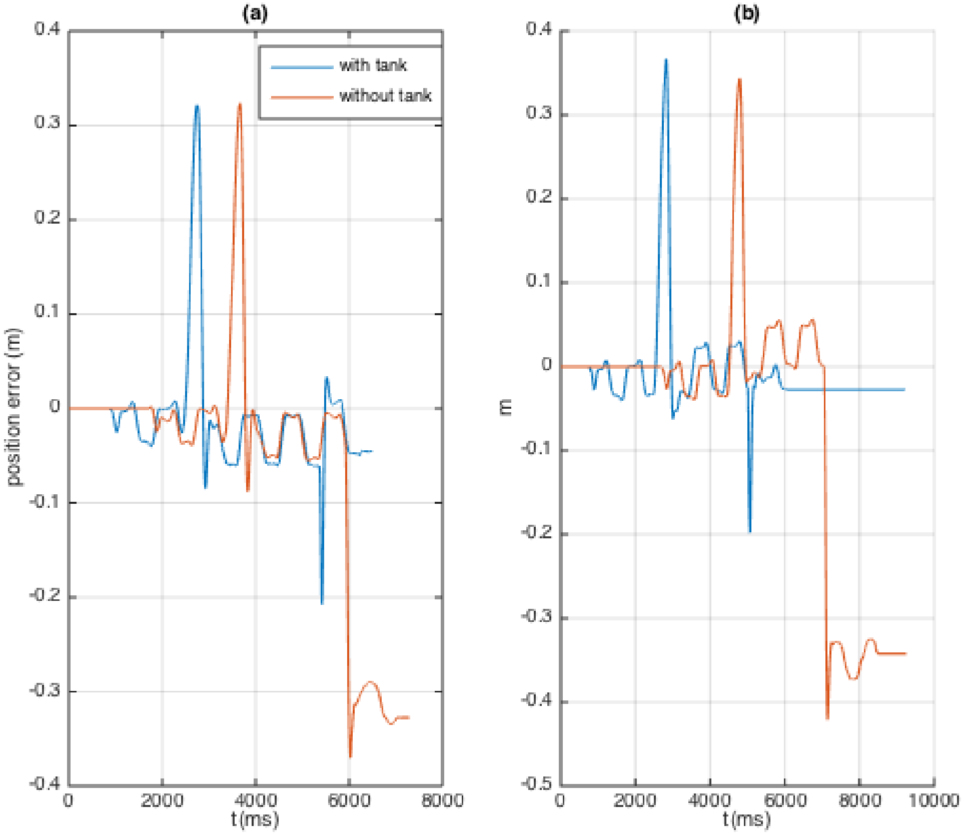


Figure 6.2: position errors measured with and without the tank when applying $3Kg$ (a) and $4Kg$ (b).

Experiment results setting stiffness $K_d = 200$

Results found in this experiment are similar to the previous case, although the error reduction decreased in percentage, especially in the $3Kg$ load case, while it became far more relevant when increasing it to $4Kg$, as will be seen in detail in the tables. This shows, for medium-range stiffnesses, an high dependency of the position and force error from the applied load The time needed by the controller to recover from the error after the contact loss (when the tank is active), is very low respect to the other case, and it appears to be independent from the weight and stiffness differences. Anyway the precise value of this time interval is highly dependent from the motion and the time instant in which the object is detached, therefore is considered only its range of variation, between 50 and 100 milliseconds for all trials.

Experiment results setting stiffness $K_d = 300$

In the last trial of this autonomous control experiment, K_d is set to a quite stiff value, therefore 3 Kg weight is not enough to show the effect of our passivity controller in this case. This happens because the spring value is high enough to allow the force deriving impedance controller to have a strong response to the position error sensed after the contact loss, to limit its and autonomously return to its set point before that the tank actually reaches its lower limit.

Therefore, 4 and $5Kg$ were chosen for this experiment, same behavior of the previous cases is shown here, also if as expected the differences in terms of reduction of the position error in the vertical axis is not relevant, as shown in figure 6.3 with respect to the $4Kg$ trial, only if increasing the load to $5Kg$ there is some difference, also if not comparable to the reduction percentages seen in the first case (stiffness 100).

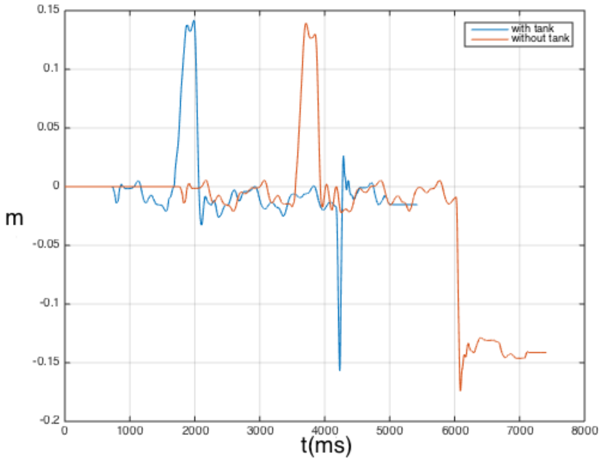


Figure 6.3: position errors measured for stiffness 300 with and without the tank when applying $4Kg$.

The same consideration can be done if looking at the values of the impedance controller forces with respect to the compensation force terms. In figure 6.4 are shown the impedance forces (yellow and blue curves) and the respective compensation forces (purple and orange curves) respectively with and without the tank, for the $4Kg$ trial.

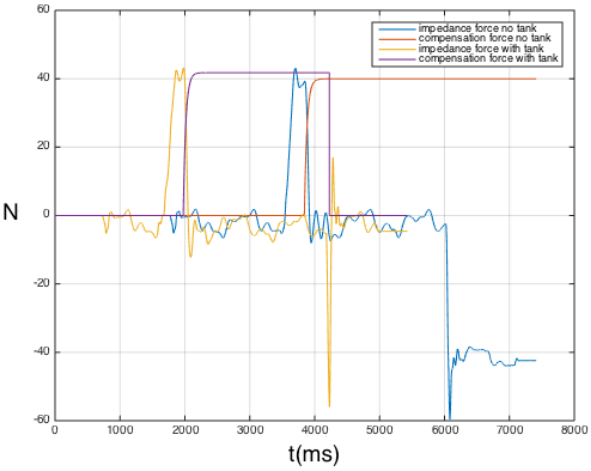


Figure 6.4: compensation and impedance forces measured for stiffness 300 with and without the tank when applying $4Kg$.

The difference made by the energy-tank stability controller, in the end, is the steady detachment of the force controller after the contact loss, when recognized. To evaluate quantitatively the effect of this, two parameters should be considered: the maximum position displacement reached by the end effector respect to its reference and the maximum force applied by the end effector. Since the total force is equal to the sum of the force applied by the impedance controller and the compensation force, the maximum value of the impedance force before returning to zero (when the compensation term is detached), should be considered.

measured position errors						
Applied Loads	**Stiffness 100**		**Stiffness 200**		**Stiffness 300**	
	Tank	No Tank	Tank	No Tank	Tank	No Tank
3Kg	20.79 cm	37.01 cm	15.73 cm	17.50 cm	-	-
4Kg	19.83 cm	42.10 cm	14.09 cm	23.93 cm	15.70 cm	17.40 cm
5Kg	-	-	-	-	12.82 cm	20.67 cm

Table 6.1: measured position errors for each stiffness and weight values

As expected, the greatest reduction of the position error is for the lowest stiffness of $K_d = 100$, where the tank reduce it of almost the 50% in both the $3Kg$ and $4Kg$ trials. On the other hand, its contribution is less evident in the $K_d = 300$ case, especially in the $4Kg$ case. This happens because the error reduction is proportional to the applied load, so also if in medium stiffness ($K_d = 200$) with $3Kg$, the difference is almost insignificant, it becomes evident with $4Kg$.

Another effect of the tank is that, for every stiffness, when increasing the weight, the maximum error decreases, while the opposite behaviour should normally happen, as shown in the results obtained without the tank. This happens because the increasing load, accelerate the rate of tank unloading, and anticipate the moment in which the force compensator is detached from the rest of the controller.

measured impedance force errors						
Applied Loads	**Stiffness 100**		**Stiffness 200**		**Stiffness 300**	
	Tank	No Tank	Tank	No Tank	Tank	No Tank
3Kg	27.4498 N	40.4197 N	39.6191 N	39.9259 N	-	-
4Kg	24.5012 N	47.7305 N	36.5556 N	55.9162 N	56.0567 N	59.5945 N
5Kg	-	-	-	-	50.8533 N	70.3630 N

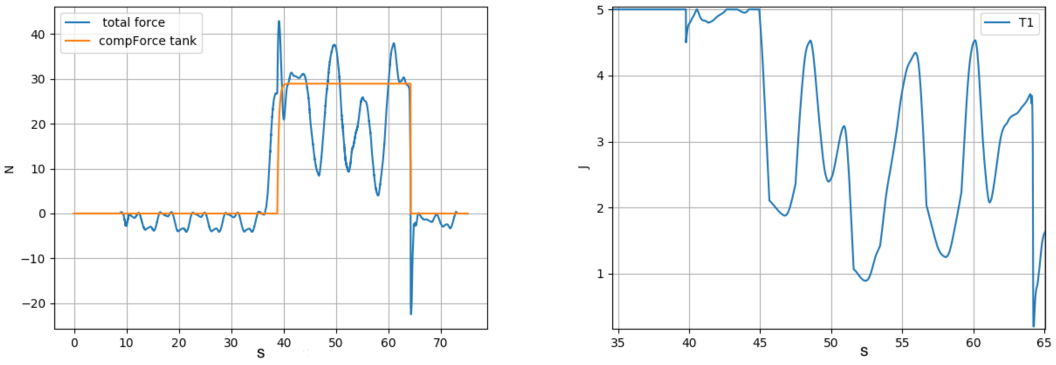
Table 6.2: measured force errors for all stiffness and weight values

The same considerations done with the position errors are valid also for the maximum force applied by the impedance controller before being detached. This is normal, being the $\mathbf{F_i}$ defined as a force proportional to the position and velocity errors.

In particular, without the tank action, the force continues being compensated also after the contact loss, and the impedance term have to contrast it, so the set-point become equal to the opposite of the object weight. The action of the spring in the impedance controller should allow, after a certain time, to recover from the error, but in almost all cases, if the stiffness is low and the load is high (in proportion), this doesn't happen and the error stay constant to its maximum value. This doesn't happen when the tank is active, where the immediate detachment of the compensation force term allows to follow quickly the correct reference.
So, in the end, the effect of the energy-tank on these error signals is evident not only in their amplitude, but also in their shapes. In fact when the tank is active, they both have an impulsive behaviour, reaching the force or position references always in a time comprised in an interval of 50-100 ms, while without the tank action the controller (due to its low stiffness value) is often unable to reach its references, and the error signals present a step shape.

6.2 Reaction to external perturbances

To prove the validity of our contact-recognition filtering action, described in section 4.3.1, two experiments were done putting in the end-effector a weight of $3Kg$ with a stiffness of $K_d = 100$ and a weight of $4Kg$ with $K_d = 200$. When the weights are lifted, huge external disturbances in vertical direction were provided pushing manually the end effector in both positive and negative direction; after that the weights were suddenly took off from the end effector, simulating an object loss.



(a) **(b)**

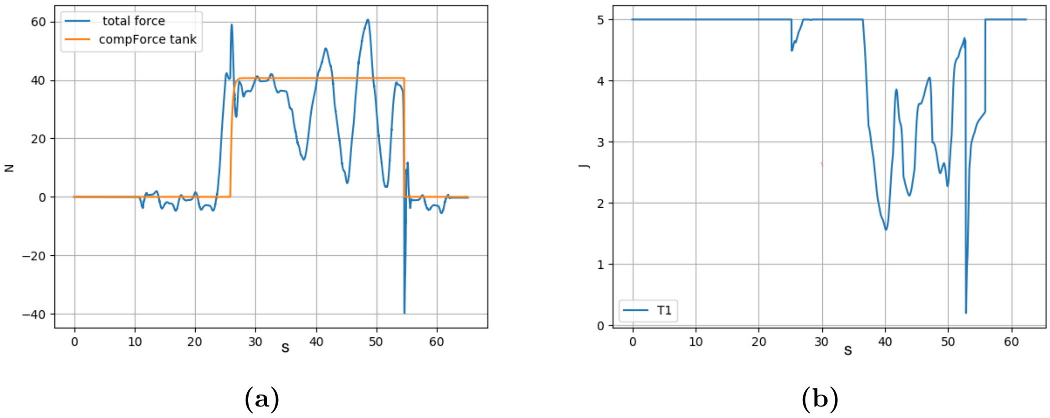
Figure 6.5: external perturbation effects with low stiffness

In figures 6.5 and 6.6, are shown the compensation force activation and deactivation w.r.t the total force (figures 6.5a and 6.6a), and the correspondent tank levels (figure 6.5b and 6.6b).

It is evident how after that the weight force is recognized and compensated, the passivity controller distinguishes the other external disturbances coming from the environment (in this case the human arm) from the previous weight, and consequently maintain a compliant behaviour.

Thus, there isn't any variation in the force compensation action, in the period of time between $t = 40$ and $t = 65$ (following the time line of figure 6.5a), in that interval the tank level oscillates because of the combined actions of the input power flow provided by the damper D_d, and the discharging term $\dot{\tilde{\mathbf{x}}}\tilde{\mathbf{F}}_{\mathbf{err}}$. In the end (t=64), when the contact with the object is lost, an external force directed in the positive vertical direction is sensed, with a correspondent high velocity, therefore the contribution of $\mathbf{F}_{\mathbf{err}}$ is not reduced and the tank level immediately reach the lower limit (in this case equal to 0.2).

The tank is initialized at its maximum level to study the maximum possible error in experiments with an immediate object loss (as the ones in the next section), this upper threshold was set to a quite low value, to avoid that excessive energy storage would allow unsafe motions (as explained in chapter 4). The passivity controller is initialized when the force compensation part is on ("grasp" state in figure 3.1) and is detached from the rest of the controller after that it reaches its lower limit. This is done in order to prevent the force compensation controller from being switched on and off repeatedly during the object grasping (in witch an error is expected due



(a)

(b)

Figure 6.6: external perturbation effects with medium stiffness

to the two seconds delay), and to give a security margin after its deactivation. In fact, in figure 6.6b is shown how, after that the force control part is turned off, a "bouncing" effect could immediately reactivate it and cause an unsafe motion.

6.3 Teleoperated control

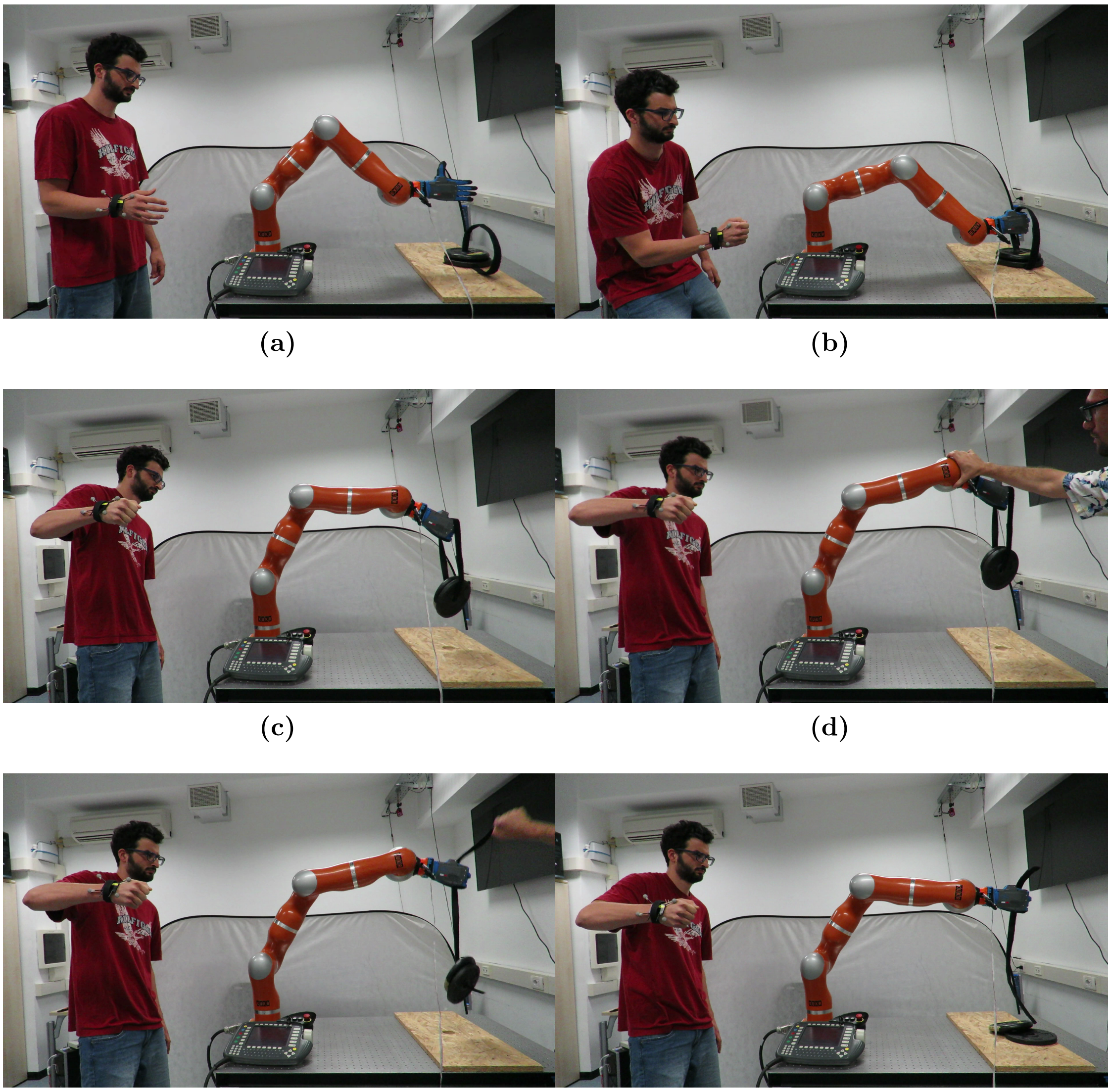
After that the passivity controller was tuned with the autonomous control trials, experiments providing human motion references were done, stiffness was chosen $K_d = 200$ in order to have a trade-off between the need of having a low stiffness for the passivity controller performances (as seen in previous section), and the fact that low stiffness means low velocity with which the slave robot follows human motion.

A passive marker was applied in human operator's wrist, after the initial calibration, its positions were acquired with an Optitrack system (Natural Point, Inc.) and sent to the Kuka, equipped for this experiment with the Pisa-IIT SoftHand as end-effector. Motions were considered relatively to the initial point, so in the calibration phase, the initial position should be measured in order to be added as offset to the future references. Now, the robot can safely follow human arm movement.

In figure 6.7 are shown the several phases of the grasping task: the considered starting position allowed comfortable visual feedback (figure 6.7a); then a load of $4Kg$ was grasped ((figure 6.7b)), when the sensed external force in the vertical direction overcome the threshold the same procedure described in chapter 3 was followed, but with a measurement time of 6 seconds instead of 2, in order to give more time to the controller to provide an accurate measure of the object weight to compensate.

During this period of time, due to the chosen low stiffness, the robot is not able to follow operator's references. If the sensed force is constant after the 6 seconds, the slave controller compensate it (figure 6.7c) consenting the robot to return to reference point. Robot is able to follow motion as before the grasping and still be compliant also if an external disturbance on the end-effector occurs (figure 6.7d), the tank doesn't completely unload and the force terms stay active showing how the controller is able to distinguish different type of contacts. When an unexpected object-loss occurs (figure 6.7e) the tank unloads and turns off the force compensation, with a vertical position error comparable to the results found in the autonomous task, and the robots returns to the reference (with only impedance control term active).

The same experimental procedure was also done without the tank. Figure 6.8 shows compares the obtained position errors in vertical axis (figures 6.8a and 6.8b), the impedance controller reference force with respect to the sensed external force (figures 6.8c and 6.8d) , and the force compensation term activation (figures 6.8e and 6.8f). Results appear to be similar to what found in autonomous control trials, in terms of spatial and temporal differences in position and force errors, confirming that the validity of our approach stands also for more complex teleoperated tasks.

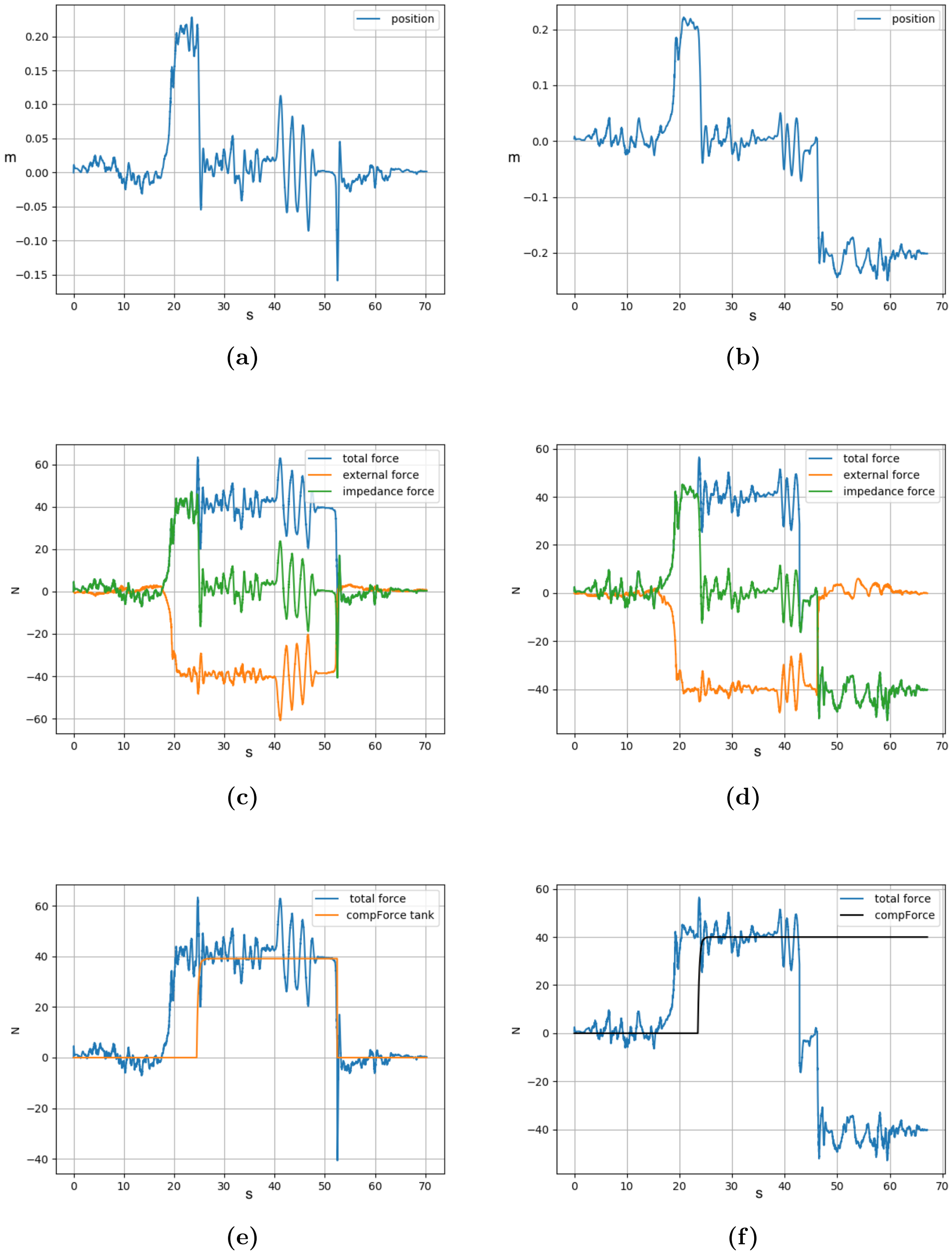


(a) (b)

(c) (d)

(e) (f)

Figure 6.7: Phases of the experiment with teleoperated robot arm.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6.8: comparison between active and inactive tank controller in teleoperated robot task.

Conclusions and future work

In this work the Teleimpedance framework was used to teleoperate a slave robot without any need of force feedback. Position references were provided by the master side, while commanded low stiffness ensures a safe and compliant behavior. In the slave side, the autonomous control part was enriched with a force controller added to the basic impedance one, using only its internal force-torque sensors, the robot was able to distinguish between different types of contacts, and regulate its control action as consequence, dynamically adapting itself to the external environment.

A passivity controller based on the energy tank paradigm was used in order to ensure stability, using the energy dissipated by the system to allow control actions otherwise impossible to implement without violating the passivity condition. When unsafe events like an unexpected object drop occurs, the tank detaches the force compensation term quickly enough to maintain the error in a reasonably small range, despite this, robustness in control action is still provided, as external perturbation added in both experiments showed.

Also if these preliminary results were good, it is still not enough for applications that require high precision, like robotic surgery, in fact the vertical-axis position error after the contact loss, as shown in experiments, should be reduced of at least an order of magnitude. A future improvement to go in this direction can be a smarter control of the activation and deactivation of the force controller consequent to variations in the tank dynamics, with a proper activation function different from the current step behavior (especially in the detachment case). A quicker object-weight estimation technique should also be implemented, in order to avoid the huge position error that occurs in the beginning of the force compensating action, caused by the measurement time delay.

The most important improvement, however, is for sure the integration of the complete Teleimpedance framework with this stability control action, including the varying stiffness term given by the EMGs (as illustrated in 2). This couldn't be done at this preliminary stages because of the additional non-passivity that varying stiffness would have added to our controller, aside the force control term. By the way, similar techniques using the energy tank have been introduced to provide passivity with a varying stiffness term (Ferraguti et al., 2013), so an extension of our approach seems the right path to integrate this controller with the complete Teleimpedance.

Bibliography

Arash Ajoudani, Nikos Tsagarakis, and Antonio Bicchi. Tele-impedance: Teleoperation with impedance regulation using a body–machine interface. *The International Journal of Robotics Research*, 31(13):1642–1656, 2012.

K. Akazawa, TE. Milner, and RB. Stein. Modulation of reflex EMG and stiffness in response to stretch of human finger muscle. *Journal of Neurophisiology*, 49: 16–27, 1983.

Alin Albu-Schaffer and Gerd Hirzinger. Cartesian impedance control techniques for torque controlled light-weight robots. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 657–663. IEEE, 2002.

Robert J Anderson and Mark W Spong. Bilateral control of teleoperators with time delay. *IEEE Transactions on Automatic control*, 34(5):494–501, 1989.

Panagiotis K Artemiadis and Kostas J Kyriakopoulos. Emg-based control of a robot arm using low-dimensional embeddings. *IEEE Transactions on Robotics*, 26(2):393–398, 2010.

Jordi Artigas, Carsten Preusche, and Gerd Hirzinger. Time domain passivity for delayed haptic telepresence with energy reference. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1612–1617. IEEE, 2007.

L Barbé, B Bayle, M De Mathelin, and A Gangi. Online robust model estimation and haptic clues detection during in vivo needle insertions. In *The First IEEE/RAS-EMBS Int. Conf. on Biomedical Robotics and Biomechatronics*, pages 341–346, 2006.

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

M Casadio, A Pressman, S Acosta, Z Danzinger, A Fishbach, FA Mussa-Ivaldi, K Muir, H Tseng, and D Chen. Body machine interface: remapping motor skills after spinal cord injury. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pages 1–6. IEEE, 2011.

Joaquín Cervera, AJ Van Der Schaft, and Alfonso Baños. Interconnection of port-hamiltonian systems and composition of dirac structures. *Automatica*, 43(2): 212–225, 2007.

Ho Ching and Wayne J Book. Internet-based bilateral teleoperation based on wave variable with adaptive predictor and direct drift control. *Journal of Dynamic Systems, Measurement, and Control*, 128(1):86–93, 2006.

Febo Cincotti, Donatella Mattia, Fabio Aloise, Simona Bufalari, Gerwin Schalk, Giuseppe Oriolo, Andrea Cherubini, Maria Grazia Marciani, and Fabio Babiloni. Non-invasive brain–computer interface system: towards its application as assistive technology. *Brain research bulletin*, 75(6):796–803, 2008.

J Edward Colgate, Paul E Grafing, Michael C Stanley, and Gerd Schenkel. Implementation of stiff virtual walls in force-reflecting interfaces. In *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, pages 202–208. IEEE, 1993.

SJ. De Serres and TE. Milner. Wrist muscle activation patterns and stiffness associated with stable and unstable mechanical loads. *Experimental Brain Research*, 86:451–458, 1991.

Cosimo Della Santina, Giorgio Grioli, Manuel Catalano, Alberto Brando, and Antonio Bicchi. Dexterity augmentation on a synergistic hand: the pisa/iit soft-hand+. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 497–503. IEEE, 2015.

J. M. Dolan, M. B. Friedman, and M. L. Nagurka. Dynamic and loaded impedance components in the maintenance of human arm posture. *IEEE Transactions on Systems, Man, and Cybernetics*, 23:698–709, 1993.

A. Eusebi and C. Melchiorri. Force reflecting telemanipulators with timedelay: Stability analysis and control design. *IEEE Transactions on Robotics and Automation*, 14(4):635–640, 1998.

Cristian Secchi Stefano Stramigioli Cesare Fantuzzi. Control of interactive robotic interfaces.

Federica Ferraguti, Cristian Secchi, and Cesare Fantuzzi. A tank-based approach to impedance control with variable stiffness. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4948–4953. IEEE, 2013.

Michel Franken, Stefano Stramigioli, Sarthak Misra, Cristian Secchi, and Alessandro Macchelli. Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency. *IEEE transactions on robotics*, 27(4): 741–756, 2011.

DW. Franklin, R. Osu, E. Burdet, M. Kawato, and T. Milner. Adaptation to stable and unstable dynamics achieved by combined impedance control and inverse dynamics model. *Journal of Neurophysiology*, 90(5):3270–3282, 2003.

R. Goertz, R. Blomgren, J. Grimson, G. Forster, W. Thompson, and W. Kline. The ANL model 3 master-slave electric manipulator - its design and use in a cave. *Transactions of the American Nuclear Society*, 4(2):219–220, 1961.

H. Gomi and R. Osu. Task-dependent viscoelasticity of human multi joint arm and its spatial characteristics for interaction with environments. *Journal of Neuroscience*, 18:65–78, 1998.

B. Hannaford and R. Anderson. Experimental and simulation studies of hard contact in force reflecting teleoperation. In *International Conference on Robotics and Automation*, pages 584–589, 1988.

Blake Hannaford and Jee-Hwan Ryu. Time-domain passivity control of haptic interfaces. *IEEE Transactions on Robotics and Automation*, 18(1):1–10, 2002.

John Paulin Hansen, Kristian Tørning, Anders Sewerin Johansen, Kenji Itoh, and Hirotaka Aoki. Gaze typing compared with input by head and hand. In *Proceedings of the 2004 symposium on Eye tracking research & applications*, pages 131–138. ACM, 2004.

David J Hernandez, Vladimir A Sinkov, William W Roberts, Mohamad E Allaf, Alexandru Patriciu, Thomas W Jarrett, Louis R Kavoussi, and Dan Stoianovici. Measurement of bio-impedance with a smart needle to confirm percutaneous kidney access. *The Journal of urology*, 166(4):1520–1523, 2001.

Ulrich Hoffmann, Jean-Marc Vesin, Touradj Ebrahimi, and Karin Diserens. An efficient p300-based brain–computer interface for disabled subjects. *Journal of Neuroscience methods*, 167(1):115–125, 2008.

Neville Hogan. Impedance control: An approach to manipulation. In *American Control Conference, 1984*, pages 304–313. IEEE, 1984.

Aulikki Hyrskykari, Päivi Majaranta, and Kari-Jouko Räihä. From gaze control to attentive interfaces. In *Proceedings of HCII*, volume 2, 2005.

T. Imaida, Y. Yokokohji, M. Oda T. Doi, and T. Yoshikawa. Ground space bilateral teleoperation of ETS-VII robot arm by direct bilateral coupling under 7-s time delay condition. *IEEE Transactions on Robotics and Automation*, 20(3):499–511, 2004.

Pei Jia, Huosheng H Hu, Tao Lu, and Kui Yuan. Head gesture recognition for hands-free control of an intelligent wheelchair. *Industrial Robot: An International Journal*, 34(1):60–68, 2007.

Hairong Jiang, Juan P Wachs, Martin Pendergast, and Bradley S Duerstock. 3d joystick for robotic arm control by individuals with high level spinal cord injuries. In *Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on*, pages 1–5. IEEE, 2013.

Kenji Kawashima, Kotaro Tadano, Ganesh Sankaranarayanan, and Blake Hannaford. Model-based passivity control for bilateral teleoperation of a surgical robot with time delay. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1427–1432. IEEE, 2008.

Jong-Phil Kim and Jeha Ryu. Robustly stable haptic interaction control using an energy-bounding algorithm. *The International Journal of Robotics Research*, 29 (6):666–679, 2010.

Marco Laghi, Arash Ajoudani, Manuel Catalano, and Antonio Bicchi. Tele-impedance with force feedback under communication time delay. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 2564–2571. IEEE, 2017.

Dale A Lawrence. Stability and transparency in bilateral teleoperation. *IEEE transactions on robotics and automation*, 9(5):624–637, 1993.

Dongjun Lee and Ke Huang. Passive position feedback over packet-switching communication network with varying-delay and packet-loss. In *Haptic interfaces for virtual environment and teleoperator systems, 2008. haptics 2008. symposium on*, pages 335–342. IEEE, 2008.

Dongjun Lee and Ke Huang. Passive-set-position-modulation framework for interactive robotic systems. *IEEE Transactions on Robotics*, 26(2):354–369, 2010.

G. Leung, B. Francis, and J. Apkarian. Bilateral controller for teleoperators with time delay via mu-synthesis. *IEEE Transactions on Robotics and Automation*, 1995.

KA. Mann, FW Werner, and AK Palmer. Frequency spectrum analysis of wrist motion for activities of daily living. *Journal of Orthopaedic research*, 7:304–306, 1989.

Matthew T Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):418–432, 1981.

T. E. Milner. Contribution of geometry and joint stiffness to mechanical stability of the human arm. *Experimental Brain Research*, 143:515–519, 2002.

TE. Milner, C. Cloutier, AB Leger, and DW Franklin. Inability to activate muscle maximally during cocontraction and the effect on joint stiffness. *Experimental Brain Research*, 107:293–305, 1995.

Darius Miniotas, Oleg Špakov, Ivan Tugoy, and I Scott MacKenzie. Speech-augmented eye gaze interaction with small closely spaced targets. In *Proceedings of the 2006 symposium on Eye tracking research & applications*, pages 67–72. ACM, 2006.

Luca Muratore, Arturo Laurenzi, Enrico Mingo Hoffman, Alessio Rocchi, Darwin G Caldwell, and Nikos G Tsagarakis. Xbotcore: A real-time cross-robot software platform. In *Robotic Computing (IRC), IEEE International Conference on*, pages 77–80. IEEE, 2017.

F. A. Mussa-Ivaldi, N. Hogan, and E. Bizzi. Neural, mechanical, and geometric factors subserving arm posture in humans. *Journal of Neuroscience*, 5(10): 2732–2743, 1985.

G. Niemeyer and J. Slotine. Telemanipulation with time delays. *International Journal of Robotics Research*, 23(9):873–890, 2004a.

Günter Niemeyer and Jean-Jacques E Slotine. Telemanipulation with time delays. *The International Journal of Robotics Research*, 23(9):873–890, 2004b.

R. Osu and H. Gomi. Multijoint muscle regulation mechanism examined by measured human arm stiffness and EMG signals. *Journal of Neurophysiology*, 81:1458–1468, 1999.

R. Osu, DW. Franklin, H. Kato, H. Gomi, K. Domen, T. Yoshioka, and M. Kawato. Short- and long-term changes in joint co-contraction associated with motor learning as revealed from surface EMG. *Journal of Neurophysiology*, 88:991–1004, 2002.

Christian Ott. Cartesian impedance control: The rigid body case. In *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, pages 29–44. Springer, 2008.

E. J. Perreault, R. F. Kirsch, and P. E. Crago. Multijoint dynamics and postural stability of the human arm. *Experimental Brain Research*, 157:507–517, 2004.

Luka Peternel, Tadej Petrič, Erhan Oztop, and Jan Babič. Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human-in-the-loop approach. *Autonomous robots*, 36(1-2):123–136, 2014.

Gert Pfurtscheller, Brendan Z Allison, Günther Bauernfeind, Clemens Brunner, Teodoro Solis Escalante, Reinhold Scherer, Thorsten O Zander, Gernot Mueller-Putz, Christa Neuper, and Niels Birbaumer. The hybrid bci. *Frontiers in neuroscience*, 4:3, 2010.

Giuseppe F Rigano, Luca Muratore, Arturo Laurenzi, Enrico M Hoffman, and Nikos G Tsagarakis. A mixed real-time robot hardware abstraction layer (r-hal). *Encyclopedia with Semantic Computing and Robotic Intelligence*, 2018.

Jee-Hwan Ryu and Carsten Preusche. Stable bilateral control of teleoperators under time-varying communication delay: Time domain passivity approach. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3508–3513. IEEE, 2007.

Jee-Hwan Ryu, Dong-Soo Kwon, and Blake Hannaford. Stable teleoperation with time-domain passivity control. *IEEE Transactions on robotics and automation*, 20(2):365–373, 2004.

Christopher Schindlbeck and Sami Haddadin. Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 440–447. IEEE, 2015.

G. Schreiber, A. Stemmer, and R. Bischoff. The fast research interface for the kuka lightweight robot. In *IEEE Conference on Robotics and Automation (ICRA)*, 2010.

L. Selen, P. J. Beek, and J. Van Dieen. Can co-activation reduce kinematic variability? a simulation study. *Biological Cybernetics*, 93:373–381, 2005.

Changhoon Seo, Jaeha Kim, Jong-Phil Kim, Joo Hong Yoon, and Jeha Ryu. Stable bilateral teleoperation using the energy-bounding algorithm: Basic idea and feasibility tests. In *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*, pages 335–340. IEEE, 2008.

Homayoun Seraji. Adaptive admittance control: An approach to explicit force control in compliant motion. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 2705–2712. IEEE, 1994.

T. Sheridan. Space teleoperation through time delay: Review and prognosis. *IEEE Transactions on Robotics and Automation*, 9(5):592–606, 1993.

D. Shin, J. Kim, and Y. Koike. A myokinetic arm model for estimating joint torque and stiffness from EMG signals during maintained posture. *Journal of Neurophisiology*, 101:387–401, 2009.

Neal A Tanner and Günter Niemeyer. Improving perception in time-delayed telerobotics. *The International Journal of Robotics Research*, 24(8):631–644, 2005.

Keng Peng Tee, David W Franklin, Mitsuo Kawato, Theodore E Milner, and Etienne Burdet. Concurrent adaptation of force and impedance in the redundant muscle system. *Biological cybernetics*, 102(1):31–44, 2010.

R. Trumbower, M. Krutky, B. Yang, and E. J. Perreault. Use of self-selected postures to regulate multijoint stiffness during unconstrained tasks. *PLoS One*, 4 (5), 2009.

T. Tsuji, P. G. Morasso, K. Goto, and K. Ito. Human hand impedance characteristics during maintained posture. *Biological Cybernetics*, 72:475–485, 1995.

T. Tsuji, Y. Takeda, and Y. Tanaka. Analysis of mechanical impedance in human arm movements using a virtual tennis system. *Biological Cybernetics*, 91(5): 295–305, 2004.

M. T. Turvey. Action and perception at the level of synergies. *Human Movement Science*, 26(4):657–697, 2007.

Costas Tzafestas, Spyros Velanas, and George Fakiridis. Adaptive impedance control in haptic teleoperation to improve transparency under time-delay. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 212–219. IEEE, 2008.

A vd Schaft and A Schaft. L2-gain and passivity in nonlinear control, 1999.

Roel Vertegaal. A fitts law comparison of eye tracking and manual input in the selection of visual targets. In *Proceedings of the 10th international conference on Multimodal interfaces*, pages 241–248. ACM, 2008.

PA. Weiss. *The Self-differentiation of the basic patterns of coordination.* Baltimore, Md, The Williams & Wilkins Company, 1941.

Christopher D Wickens. Multiple resources and performance prediction. *Theoretical issues in ergonomics science*, 3(2):159–177, 2002.