UNIVERSITA CAMPUS BIO−MEDICO DI ROMA

FACOLTA DI INGENGNERIA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA BIOMEDICA

Development of a Bilateral Safe Teleoperation Control for CERN accelerators harsh environment

Relatore:
Ch.ma Prof.ssa Loredana Zollo

Correlatore:
Dott. Ing. Mario Di Castro (CERN)
Dott. Ing. Alessandro Masi (CERN)

Candidato:
Alessandro Mosca
Matr. 08107

Anno Accademico 2014/15

Table of Contents

Acknowledgment

Introduction

This work has been carried out in collaboration with the European Center for Nuclear Research (CERN) of Geneva. Main object of this thesis consists in the development and testing of a Bilateral Teleoperated Control firmware Master-Slave, in order to fulfil telemanipulation tasks in harsh and non-structured environment, as the one inside the Large Hadron Collider (LHC), the world's largest and most powerful particle accelerator. The LHC environment is characterized by a not negligible ionizing radiation values. The ionizing radiation is caused by the radioactive decay of spallation-induced radionuclides produced inside the components and their surroundings during beam operation. Current conditions require the intervention of the human factor within that environment, even for performing simple and basic operations such as: screwing, unscrewing and turning a switch.

Considering these aspects, the thesis work aims, as long term objective, the creation of a robotics team that will replace, completely, the human presence making these tasks, contributing, at the same time, to the reduction of workers 'exposure to ionizing radiation.

The Slave Side robot is represented by the Schunk PowerBall LWA4.6, a manipulator composed by six Degrees of Freedom (DOF) which reflects the cinematic chain of a classic "Antropomorphic Manipulator with Spherical Wrist". The robot, provided by the multinational manufacturing company "SCHUNK Intec, Inc.", was devoid of any application-programming interface (API), for that reason, the first step, was creating all it would allow us to interface with the robot, from reading of individual states of the joint (angular position, angular velocity, angular acceleration and current) to sending command to the each joints.

The Geomagic Touch Haptic Device (or Sensable Phantom Omni Haptic Device), instead, represents the Master Side robot. Used in research, 3D modeling, Original Equipment Manufacturer (OEM), scientific or medical simulations and more, also this device is characterized by six DOF. Its kinematic chain is particular. It has the first three joints depute on identifying the Cartesian position of the end-effector. The last three joint compose the well-known "Gimbal Joint", which defines the Cartesian orientation of the end-effector. Only three of its joints are actuated, specifically, the first three, which may imparting forces along the three spatial directions.

Tests were carried out to prove the proper working of the control system and the success of the predetermined task.

In addition to the main objective, the thesis work aims to accomplish the following ones:

- Implementation of a communication protocol
- Kinematic study and development of real time algorithm for the inverse kinematic
- Possibility of moving the slave robot joint-by-joint
- Design and development of user friendly GUI

Below a brief description of each chapter gods.

In Chapter 1, an introduction to CERN is made, followed by a description of the Large Hadron Collider (LHC) and its structure and basic components. Subsequently, we focus on one of LHC's vital parts, the Collimation system, providing details about its function, as well as future trends.

In Chapter 2, a description of the state of the art and different bilateral Teleoperated control architecture are introduced with a characterization of the experimental setup and the specification of the tasks that the robot will fulfill.

In Chapter 3 is dedicated to the analysis of the framework control developed and to the realization of:

- the robot Application Programming Interface (API) for the Schunk PowerBall LWA4.6

- the communication between Master and Slave through a TCP/IP Socket Protocol
- Forward Kinematic
- Inverse Kinematic

In Chapter 4 is focused on the description of the architecture used for the bilateral control, the kind of control used for the master and the slave robot and presentation of the results obtained using the control developed.

The Chapter 5 is reserved to the possible future works and conclusion.

Chapter 1

Introduction to CERN

In the first part of this chapter a general knowledge of the CERN is provide, followed by a description of the LHC, its structure and basic component.

The second part presents the Collimation system, one of the LHC' s vital part, and the reason why it is so important for the successful conduct of operations research inside the tunnel. Subsequently, will be analysed the robotic systems actually used to CERN, their task and contexts of use.

The chapter concludes by highlighting those aspects that had led to the creation of this work.

CERN, the European Organization for Nuclear Research, is considered one of the world's largest and most respected laboratories for scientific research and is located in the northwest suburbs of Geneva on the Franco-Swiss border. Founded on September 29, 1954 by a group of 12 European countries (Belgium, Denmark, France, Germany, Italy, Greece, Yugoslavia, Norway, Holland, Great Britain, Swedish and Switzerland), actually has 22 member states and Israel is the first (and currently only) non-European country granted full membership, but it boasts collaborations with nations from all over the world [1] [2] [3] .

Figure 1.1: CERN logo

Since its foundation, CERN has been at the forefront of science research and several important discoveries have taken place in it, leading among others to two Nobel physics prizes (the last received for the discovery of the Higgs boson) and the invention of the World Wide Web.

More than 10000 scientists and 600 institutions from more than 100 countries make it one of the most multicultural and diverse environments, participating in the activities of CERN, whose main area of research is particle physics, the study of the fundamental constituents of matter and the forces acting between them.

1.1 CERN Accelerator Complex

The Standard Model (SM) represents a set of physical theories, which includes the whole knowledge about the elementary particle, of which we are made and everythings' that exists in nature, the fundamental forces which govern the behaviour of the universe.

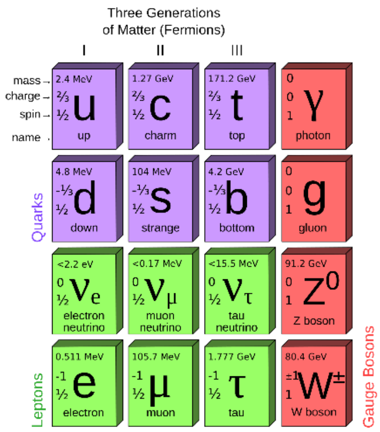

Figure 1.2 : Scheme of Particles described by Standard Model

The SM works fine and supported by multiple experimental tests, but the model does not explain some phenomena, as the black matter, the absence of antimatter in the Universe and the gravitation. To answer to these and other questions, the physics and engineers of the CERN use a particular machine called "Particles Collider" [4].

The Large Hadron[1] Collider[2] is the largest, the most powerful and the most complex particle collider ever build. The Particle Collider accelerates the particle beam (e.g. protons) leading them near to speed of light. Subsequently, the particles are made to collide each other at high energies[3] to reproduce conditions similar to the first instants of the Universe.

The CERN complex accelerator is composed by seven main accelerators, built in different periods, which create a chain in which the particles 'energy is progressively increased. These accelerators are [5]:

- Two LINAC (Linear Accelerator) that provide low energy (~50 MeV) particles.
- PS (Proton Synchrotron) Booster which accepts the particles from the LINAC and increases their energy up to 1,5 GeV for the protons.
- Then, in PS (Proton Synchrotron), the particles achieve the energy of 28 GeV.
- The SPS (Super Proton Synchrotron) is a little circular accelerator of 2 km of diameter, able to increase the energy

---

[1] The term hadron refers to composite particles composed of quarks held together by the strong force (as the electromagnetic force holds atoms and molecules together).

[2] A collider is a type of a particle accelerator with two directed beams of particles.

[3] The first accelerator used was the Large Electron-Positron (LEP) collider, which is able to accelerate electrons and positrons up to 100 GeV. The LEP was substituted by the LHC, which, initially, was able to achieve energy equal to 7 TeV. Now, after two years of work, the energy inside the collider achieves values that will be gradually raised up to 14 TeV.

of the particles up to 450 GeV. It represents the final stage, before entering in the LHC.

- LHC, started operating on 10 September 2008, covers a circumference of 27 km and is designed for maximum energy of 14 TeV. It is the highest in history and should allow experimental conditions comparable to those of the first moments of the Universe immediately after the Big Bang.

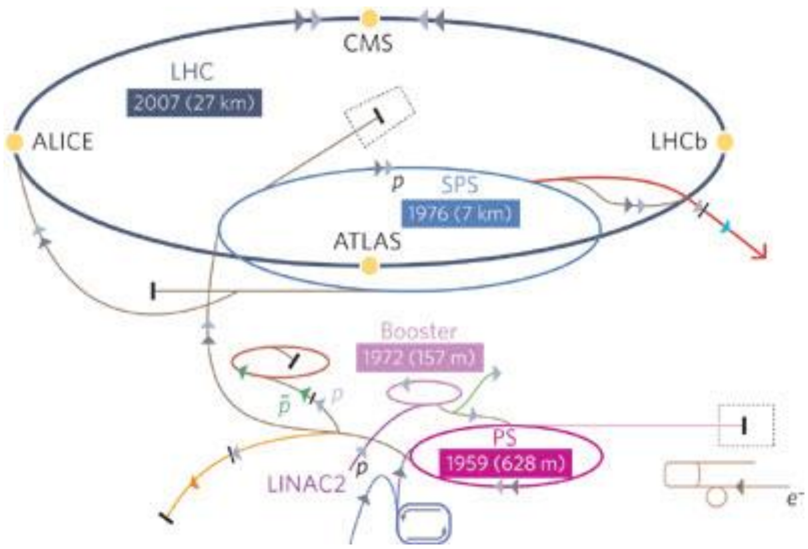

Figure 1.3: Layout of the CERN accelerator complex

During the collisions, new particles are created which are observed thanks to sophisticated and complex detectors. Actually, along the 27 km of the LHC, four particle detectors have been constructed and housed in huge underground caverns, corresponding to the four collision points of the two beams [6] [7].

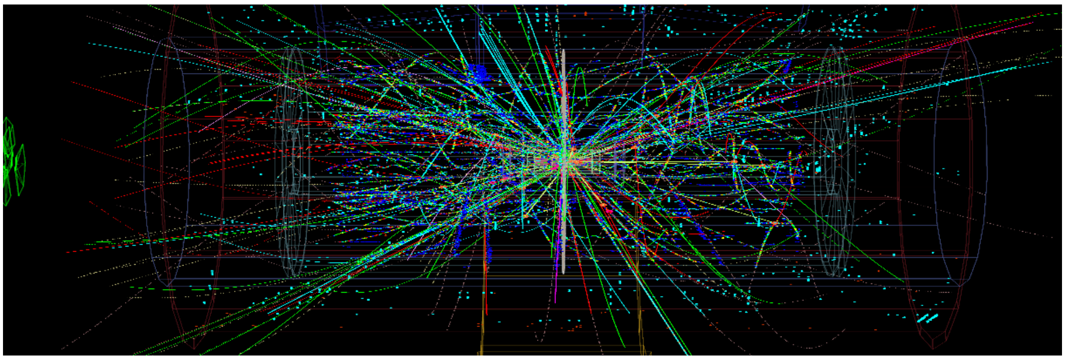

Figure 1.4: Computer reconstruction of protons collision

To each detector correspond a specific experiment:

- ALICE (A Large Ion Collider Experiment) studies heavy−ion collisions at an energy of 2.76 TeV per nucleon pair. The resulting temperature and energy density are expected to be high enough to produce quark - gluon plasma, a state of matter believed to have existed a fraction of the second after the Big Bang [8].
- ATLAS (A Toroidal LHC ApparatuS) is intended to investigate many different types of physics that might become detectable in the energetic collisions of the LHC. Some of these are confirmations or improved measurements of the Standard Model, while many others are possible clues for new physical theories [9].
- LHCb (Large Hadron Collider beauty) is dedicated to the study of the antimatter [10].
- CMS (Compact Muon Solenoid) is looking for evidence of physics beyond the SM, such as supersymmetry or extra dimensions, and studies aspect of heavy ion collisions [11].

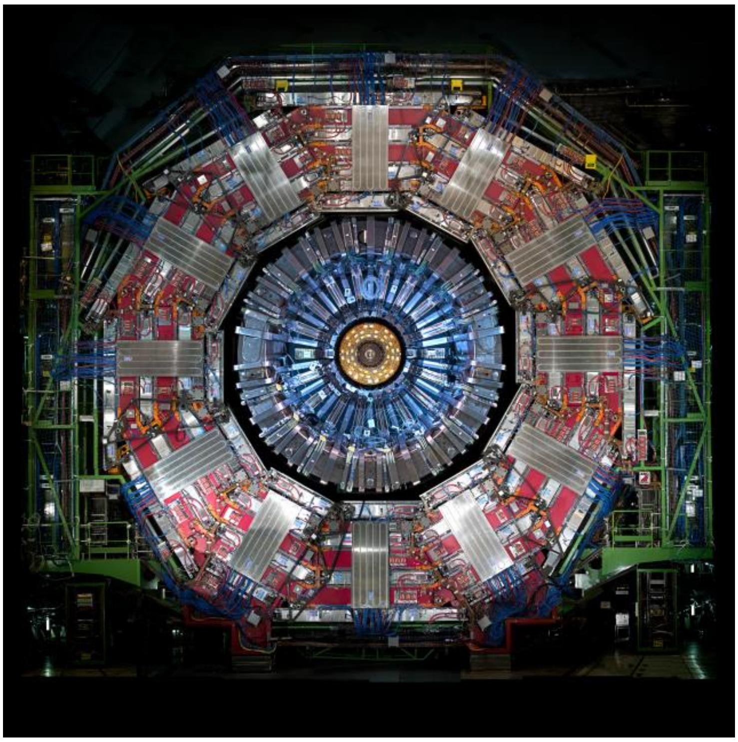

Figure 1.5: a cross section of the open CMS detector

In addition there are also the TOTEM (Total Cross Section, Elastic Scattering and Diffraction Dissociation at the LHC), LHCf (Large Hadron Collider forward), and MoEDAL (the Monopole and Exotics Detector At the LHC) [12].

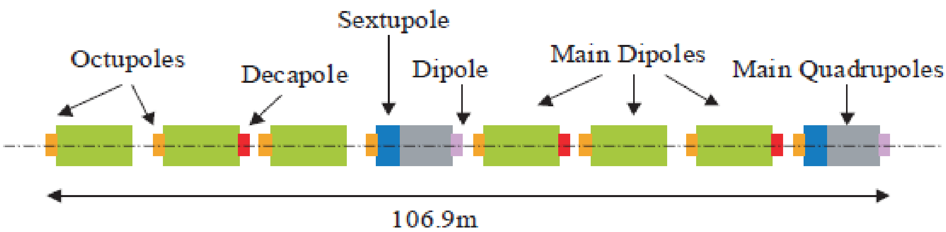

Figure 1.6: Schematic layout of the LHC arc cell. The main structure consist of main dipole and quadrupole magnets.

Important elements that composed the LHC are the superconducting magnets. They are responsible for keeping the beams stable and precisely aligned. Specifically dipole magnets are one of the most complex parts of the LHC and are used to bend the trajectory of the particles. This is necessary, as LHC is a circular accelerator, which means that the beam circulates many times in closed orbit before reaching its peak energy level, as opposed to a linear accelerator, where charged particles travel along a straight trajectory and go through a number of accelerating structures. There are 1232 main dipoles installed in the LHC, each 14.3 metres long and weighing in at 35 tonnes.

Other vital parts are the quadrupole magnets. Those help to keep the particles in a tight beam; in other words, they focus the beam. The idea behind this is that when particles are bunched together, they are more likely to collide in greater numbers when they reach the LHC detectors. In addition to the dipole and quadrupole magnets, there are also smaller sextupole, octupole and decapole magnets for correcting small imperfections in the magnetic field at the extremities of the dipoles.

Another critical element of the LHC is the **beam dump system**. Beams do not circulate inside accelerators indefinitely. As particles collide with the sides of the beam pipe or with each other, the beams ‘degrade’ and so they become less likely to give collisions of interest. Accelerator physicists can choose to ‘dump’ the beams, removing them from the accelerator and sending them to be safely absorbed at a ‘beam dump’ - usually a radiation-shielded block deep underground. To understand the necessity of the beam dump system, the nominal LHC beam contains an unprecedented stored energy of 350 MJ, contained in 2808 bunches with a beam sigma of the order of 0.3 mm. The extremely high destructive power of such a beam imposes an external dump, where the beam must be extracted completely from the LHC, diluted to reduce the peak energy density and then absorbed in a dedicated system.

Finally yet importantly is the LHC collimation system.

1.2 Collimation System

The transverse energy density of the nominal beam is 1000 times higher than previously achieved in proton storage rings. Tiny fractions of the stored beam suffice to quench[4] a super-conducting LHC magnet or even to destroy parts of the accelerators. The LHC collimation system provides protection to the accelerator against unavoidable regular and irregular beam losses and achieves the necessary beam halo cleaning.

Contrary to an ideal experiment, real-world experience has shown that certain processes lead to beam losses. The manipulations needed to prepare the beams for collision, such as injection, the energy ramp and "squeeze", as well as the collisions themselves, all involve unavoidable beam losses. Moreover, as it is easily understood, those losses become greater as the beam current and luminosity are increased. What is more, due to the LHC's

[4] "Quenching" occurs when any part of a superconducting cable of a magnet goes from the superconducting to the normal resistive state. It occurs when either the critical temperature or the critical current or the critical field is by-passed, in case of movement of the superconductor by several mm (friction and heat dissipation), cooling failures and beam loss.

superconducting environment, an efficient beam−loss cleaning to avoid “quenches” from uncontrolled losses is crucial.

The main purpose of the collimation system is to clean away the beam halo while maintaining losses at sensitive locations below safe limits. The current system ensures that peak losses below 0.01% of the energy lost from the beam is deposited in the cold magnets.

The fully installed collimator consists of five main parts: (1) the base−support, which serves as stand of the collimator in the tunnel, (2) the alignment plate, which allows the adjustment of the collimator, (3) the lower plugin, which provides cooling water and electrical connections to the collimator. The aforementioned parts constitute the base−support assembly, which can be seen in Fig. 7.

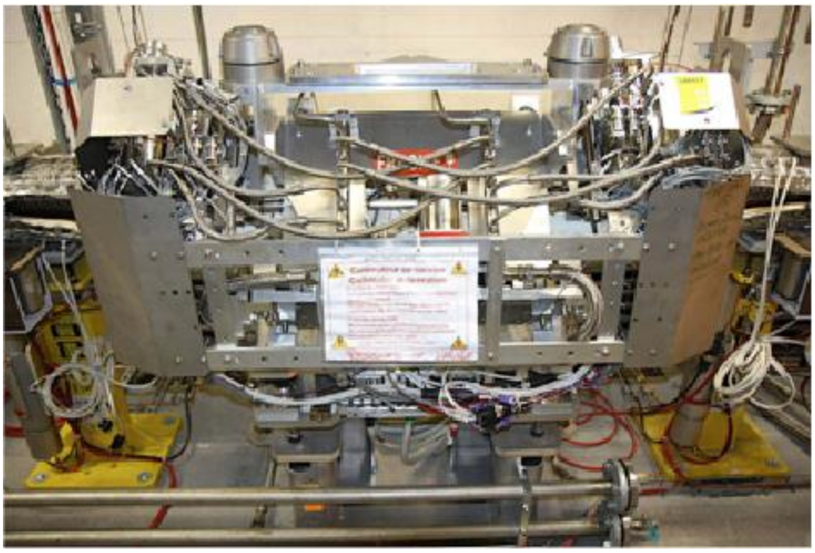

Figure 1.7: LHC Collimator

The rest of the collimation system consists of (4) the vacuum tank with the two jaws and motors for moving the jaws and (5) the upper support assembly, which hosts all needed parts for the installation in the tunnel on a base-support assembly.

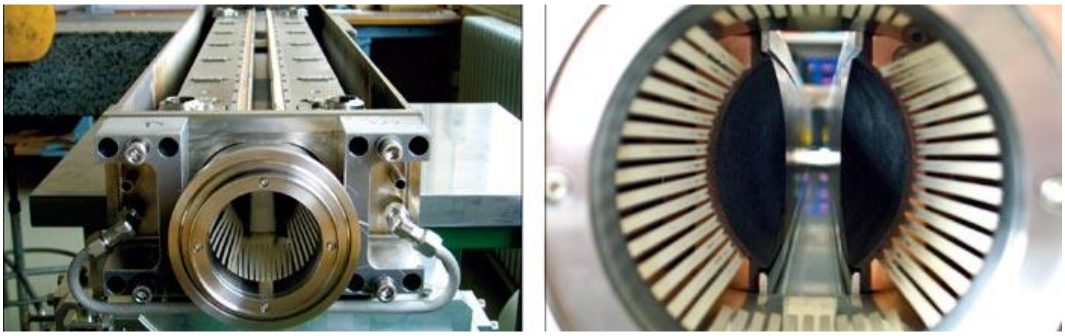

Figure 1.8: An LHC collimator with the vacuum tank open, left, and a "beam's eye-view" of the collimator aperture, right.

The discrimination between base-support and upper-support assembly serves to quickly install and exchange a collimator in case of damage.

The main tank consists of two jaws that define a slit for the beam, effectively constraining the beam halo from both sides. These jaws are enclosed in a vacuum tank that can be rotated in the transverse plane to intercept the halo, whether it is horizontal, vertical or skew. Precise sensors are used to monitor the jaws' position and collimators gaps. There are also temperature sensors mounted on the jaws. The entire aforementioned are connected to the beam-interlock system and trigger a beam dump if potentially dangerous conditions are detected.

The LHC collimation system utilizes a multi-stage cleaning process. Primary, secondary and tertiary collimators and absorbers are utilized to reduce halo particles to tolerable levels. Robust carbon-based and non-robust but high-absorption metallic materials are used for different purposes.

The LHC collimation system is the largest and most advanced cleaning system ever built for a particle accelerator. It consists of 84 two-sided movable collimators of various designs and materials, which together with injection protection collimators translate into a total of 396 degrees-of-freedom.

1.3 CERN & remote system

Automatic and remote system are already widely used in CERN's particle accelerator facilities. The reasons are due to the high level of ionizing radiation[5] [15] present in many environments.

Collisions between the particles present in the circulating beams and the components making up the accelerators result in the accelerator components becoming radioactive to varying degrees. This means that even when the beams are no longer circulating, personnel access to certain areas of the accelerators is not possible

[5] Radiation is the transport of energy through particles or waves, such as X-rays. It occurs, for example, when an unstable atom breaks down to form a stable atom, and releases some of its energy. Rocks, the sun and space all naturally emit ionizing radiation that we can detect on Earth. Artificial ionizing radiation also comes from medical treatment or in small amounts through mining and nuclear power industries. Though safe in small doses, in high doses ionizing radiation can be dangerous as the energy can knock electrons off atoms in living bodies, ionizing and damaging them. The ionizing radiation dose is usually measured in Sieverts (Sv), which is a measure of the effect of radiation on the body

until sufficient time has passed for the radiation levels to reduce due to decay. This explains why, the remote inspection, measurement and handling are the main application fields of these kind of systems.

For the LHC tunnel, a remote inspection train has been developed that is capable of carrying out remote visual inspections as well as taking measurements of radiation and oxygen levels along the tunnel. The train runs on a monorail I beam secured to the roof of the tunnel [16].

Figure 1.9: The train passing an LHC sector door

Remote handling development activities at CERN were initiated in the 1960s, with much work done throughout the 1970s and 1980s. [17]. This work included development of remote mobile inspection devices (Private Eye series), a remote mobile multi-purpose remote handling vehicle equipped with twin force-reflecting Mascot servo manipulators (Mantis).

Figure 1.10: Mantis at work

Actually, the most significant handling remote systems are the ISOLDE [6](Isotope Separator On Line DEvice) robot and the Remotely Operated Vehicle (ROV).

The existing system is made up of two six-degree-of freedom

industrial robots mounted on linear axes that are permanently installed in the radiation area. The system uses two KUKA robots[7] with all electronics removed and the non-radiation-tolerant components, such as the wiring loom and some motor components, replaced with suitable alternatives.

[6] ISOLDE is a facility dedicated to the production of a large variety of radioactive ion beams for many different experiments in the fields of nuclear and atomic physics, solid-state physics, materials science and life sciences.

[7] KUKA KR 30 : (www.kuka-robotics.com/en/products/industrial_robots/medium/kr30_3/)

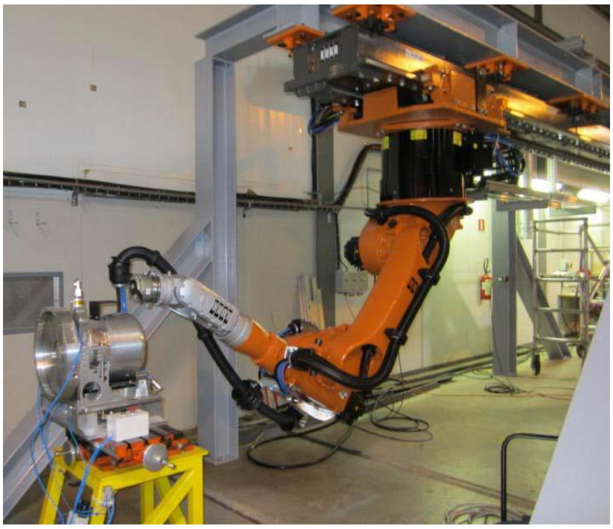

Figure 1.11: Robot trials in mock-up of target area

The robots use resolvers rather than encoders for axis position measurement, which makes the relocation of electronics away from the robot arm possible without the need for major changes in the robot controller. The robots are pre-programmed to carry out transfers of radioactive targets from their point of irradiation to shielded storage shelves.

Regarding the ROV, CERN identified the need for a general-purpose, remotely operated vehicle to carry out a variety of tasks in several different radioactive areas of CERN' s accelerator complex. The tasks include:

- Remote visual inspection
- Remote radiation dose-rate measurement
- Remote manipulation of loads and dedicated tools

A Telemax ROV was purchased from Telerob (Germany). The Telemax has a base equipped with tracks, is capable of climbing stairs, and is fitted with a six-axis manipulator arm.

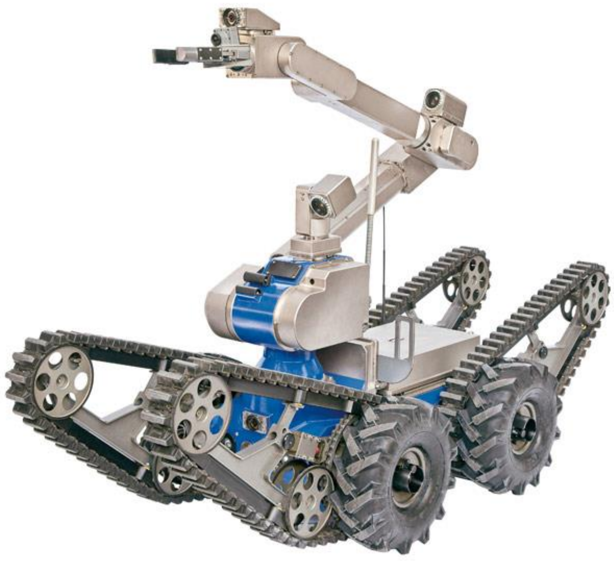

Figure 1.12: Telemax ROV

The arm has some robotic capabilities, allowing the operator to select a tool mode of operation where the movements of the arm joints are combined to provide orthogonal movements of the gripper.

1.4 Motivation

. The European Research Centre hosts inside many complex structures, which require ongoing maintenance. Machinery and devices that are not located in harsh environments cause no problems to technical personnel in charge of their maintenance, but the different is the case in which they work in radioactive environment.

The radiation presents at CERN, created by colliding beams of particles together, is different to radiation from a nuclear power plant. The radiation occurs only when the particle beam is on, and turning it off stops emissions immediately, anyway, the access to that area is forbidden until the radiation level goes below a safe threshold, slowing maintenance operations. Moreover, radiation sometimes activates some pieces of components surrounding the collision points, causing them to become radioactive. Although the level of radioactivity is not so high, but a prolonged exposure could cause irreparable damage to the biological system.

In order to safeguard and protect technical personnel against possible damages, to reduce their radiation exposure and making maintenance operations faster, not caring about the radiation level, this work aims to create a robotic e robust system able of substituting human presence during specific tasks inside harsh and radioactive environments.

Chapter 2

State of the Art

This chapter concerns about a brief history on telemanipulation, describing the main architecture used in bilateral control.

Subsequently, the focus of discussion will shift on haptic device and force feedback, what are they application field and what are the advantages that can lead.

2.1 Telemanipulation History

The word "Telemanipulation" derived by the union of the Greek word *tele*, which means "far off, afar, at or to a distance", and the Latin word *manipulus*, which means "handle". As is possible deduct, the term telemanipulation is refers to an action or a task performed in a different place than the one in which is located the human actor; it makes possible to control remotely a device, to whom the operator is able of forcing a desired movement. Although the telemanipulation is born with the aim of allowing the work in dangerous environments for humans, it soon found space in different fields of application; this technique is in fact indicated whenever there is the necessity to operate in environments:

- Physically far from the operator, (e.g. space, underwater...)
- Very small and in which is necessary a lot of precision (robotic surgery)
- Harsh (radioactive environments)

As is guessed, the telemanipulation system is composed by different main components: the human operator, through a robotic interface called "Master", guides a remote robotic system called "Slave".

Depending on the type of task to be performed and on the context in which the robot will have to go to work, all the different kinds of control architectures could be divided in three main groups:

- Direct control: no intelligence or autonomy in the system, the robot's movements are directly controlled by the human user.
- Shared control: if the user feedback is augmented from virtual reality or other automatic aids.
- Supervisory control: the operator gives high-level commands that are refined and executed by the slave.

The direct control (that is the one useful for aiming the thesis's goal) could be generically divided in two different types: *"unilateral"* and *"bilateral"*, Depending on the direction of information flow.

The unilateral telemanipulation (figure 2.1) owes its name because a single information flow is present between master and slave. It aims to control the slave's position using a passive master robot, which is not provided by actuators but only by position encoder, through which is possible measure the position of the device and replace the same movement to the remote one.

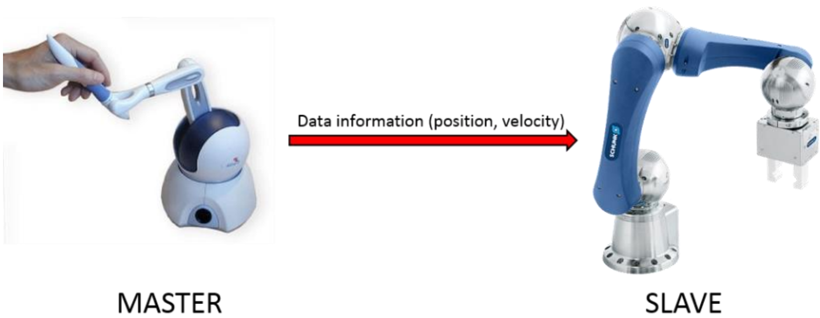

Figure 2.1: Unilateral Telemanipulation System

Raymond Goertz created the first master-slave unilateral manipulator, but also the first master-slave system, in 1948 [18]. It was controlled mechanically and used in order to handle radioactive materials

Figure 2.2: Raymond Goertz with his prototype manipulators.

Then, in 1954, Goertz modified his first version obtaining the first electrical master-slave manipulator, enabling the physical separation of two devices. [19].

The presence of a single unique communication channel from master to slave has a great limit: the human operator does not receive any feedback from the environment or from the robot, which he controls.

Figure 2.3: Bilateral Telemanipulation System

In order to resolve this kind of problem the simple way is adding a new communication channel (figure 2.3) that, from the slave robot to the master one, carries information about the robot and the surrounding environment.

Was born, so, the bilateral telemanipulation technique. In this way, through information as position, velocities of the end effector or force perceived with the help of a torque/force sensor located close to the end effector, is possible inform the operator about the environment.

Main feature for this kind of system is that both master and slave robot must be *"active"* [8].

Use of telemipulators may be found in a number of different areas. At the moment, this technology is applied in a number of different fields:

---

[8] A robot is defined active when all its joints are provided with motors that let to move them, and it is able to resist external perturbation.

A robot is defined passive when all its joints are not equipped with motors: in this way is impossible move it, and it is completely transparent to external perturbation. Is important to know that, using specific kind of control is possible make passive an active robot, but the opposite is impossible.

- Space: most part of teleoperation in space is performed in activities related to shuttles. Usually, the operator performs a direct control of the task executed by the manipulator. For space applications a dual arm force reflecting telerobotic system was developed by Bejczy et al. at the Jet Propulsion Laboratory (JPL) [20]. In 1993 the first telerobotic system (ROTEX) [21] was flown in the space with the German Spacelab Mission D2 on board the Space Shuttle Columbia.
- Underwater: at the moment mainly used for business, military missions and scientific expeditions. This field presents some difficulties like the high pressure to which the robot is subjected, poor visibility and communication problems
- Medicine: Now there is a relevant interest in applying Teleoperated devices in microsurgery, or minimally invasive surgery. In 2001, for the first time, a transatlantic telesurgery demonstration was made using a ZEUS system located in New York and performing a laparoscopic cholecystectomy on a patient located in Strasburg (France), and a lot of minimally invasive treatments are till made using robotic system (e.g. DaVinci [22], ALF-X [23].
- Hazardous environments: the possibility of using robots able to handle radioactive or very dangerous materials, lead to a drastic reduction of contagious by the personnel, increasing at the same time, the safety of operations.
- Training and simulators in telerobotics

2.2 Haptic Device and Force Feedback

The devices that make possible the exchange of information in the form of forces take the name of *“Haptic Devices”* .

Figure 2.4: Haptic Devices

The term haptic, deriving from the Greek word *haptikos* that refers to the sense of touch [24], in the robotic and mechatronic field is referred for researching of methods and tools for communication and perception of tactile sensations. Therefore, a haptic interface is then a device that allows manipulating a robot, real or virtual, and receiving in response the tactile sensations.

The application fields for this kind of technology are a lot, just think about all those circumstances in which the only vision of the work that is being done is not enough to ensure proper control and optimal, or virtual simulations that thanks to a tactile could become very realistic. By combining visual and tactile information an operator has the feeling of being on the site is operating but is in kilometres far and perceiving what he is touching can carry out his task in the best way [25][26].

Possible application field are [24]:

- Increased virtual reality: is possible recreate a virtual work environment and, adding the 3D technology to robotic haptic system, the users can do their training virtually.
- Robotic surgery: although many of the surgery robot, like DaVinci, do not make use of haptic systems, many researches are pushing their introduction, because a better perception of the consistency of human tissue to be operated could lead to a significant performance improvement in the operating room.
- Telemanipulation: the presence of haptic feedback could be useful for avoiding undesired collision, or for alerting the operator about possible error. Briefly, it could be a benefit especially in the absence of a proper visual feedback.

Interest in telemanipulation and haptic interfaces is the basis of this argument; in particular, the goal to achieve is that of implementing and experimentally test a bilateral sensorless control, avoiding the use of sensors force.

2.3 Bilateral Control

The bilateral control is characterized by the presence of the flows of communication, from master to slave and vice versa. Different channels could compose a single communication flow, and contain a specific information (position, velocity, acceleration, force and torque). Before going into details of individual architecture, is useful provide some notions about two fundamental theoretical concepts: *transparency* and *stability*.

2.3.1 Transparency and Stability

In any bilateral teleoperator system design, the essential desire is to provide a faithful transmission of signals (positions, velocities, forces) between master and slave to couple the operator as closely as possible to the remote task.

Generically a bilateral teleoperated system could be represented using the general two-port model (figure 2.5) [27].

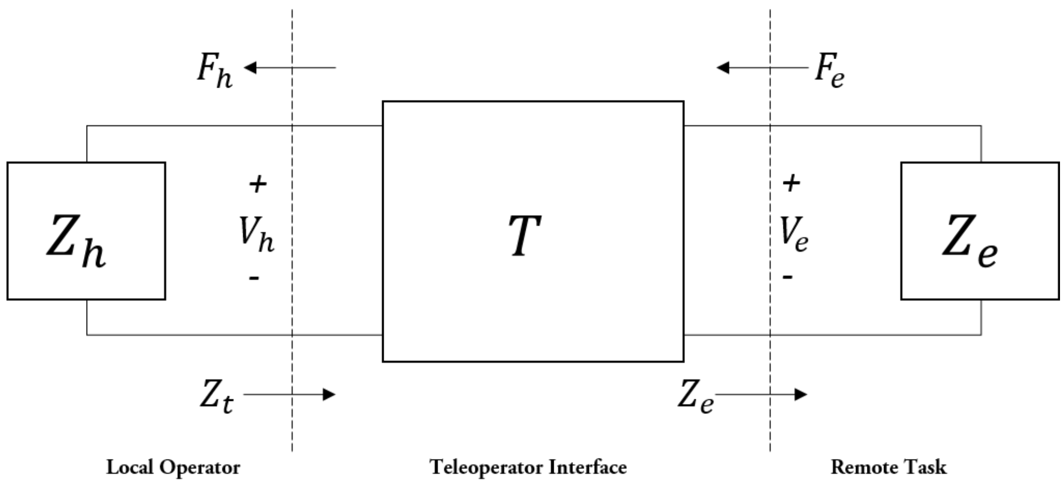

Figure 2.5: 2-port network model of bilateral control

Where:

- Z_h represents the human impedance
- F_h represents the operator' s forces on the master
- V_h represents the master/human operator velocity
- Z_t represents impedance transmitted to the operator
- Z_e represents the task impedance
- F_e represents the slave forces on the environment
- X_e represents the slave position

Ideally, the teleoperation system would be completely *transparent*, so operators feel that they are directly interacting with the remote task. A transparent system requires that the impedance transmitted Z_t or felt by the operator, match the environment impedance Z_e [28].

$$Z_t = Z_e$$

This expression comes out applying two conditions:

$$F_e = F_h$$

$$V_e = V_h$$

For the teleoperator two-port the general hybrid matrix formulation:

$$\begin{bmatrix} F_h(s) \\ V_h(s) \end{bmatrix} = \begin{bmatrix} H_{11}(s) & H_{12}(s) \\ H_{21}(s) & H_{22}(s) \end{bmatrix} \begin{bmatrix} V_e \\ -F_e \end{bmatrix}$$

Applying the transparency conditions written before, it obtained:

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

This matrix represents the hybrid matrix that satisfied the ideal transparency. Each element of the hybrid matrix has a specific meaning:

- $H_{11}(s)$ represents an input impedance in free motion condition ($F_e(s) = 0$)
- $H_{12}(s)$ represents a parameter that measures the force tracking
- $H_{21}(s)$ represents a parameter the measures the velocity tracking
- $H_{22}(s)$ represents an output admittance, without any kind of input by the operator ($V_h(s) = 0$)

Regarding the stability, is demonstrated that the teleoperated system is absolutely stable if these conditions are valid:

- H_{11} and H_{22} have not poles in right semiplan
- The poles of H_{11} and H_{22} on imaginary axes are simple poles
- For $s = jw$ with w real:

$$Re(H_{11}) \geq 0 \; Re(H_{22}) \geq 0$$

$$2Re(H_{11})Re(H_{22}) - Re(H_{11}H_{22}) - |H_{11}H_{22}| \geq 0$$

Is all these three conditions are verified, the system is absolutely stable.

2.3.2 Architectures

It will be, now, introduced the main teleoperated control architecture. Generically, is possible to classify them in different way:

- According to the controllers used on master and slave side (position–position, force–position, etc...)
- According to the number of channels that composed the communication flow.

The most complete bilateral architecture is the *"four–channel architecture"* (figure 2.6) [29].

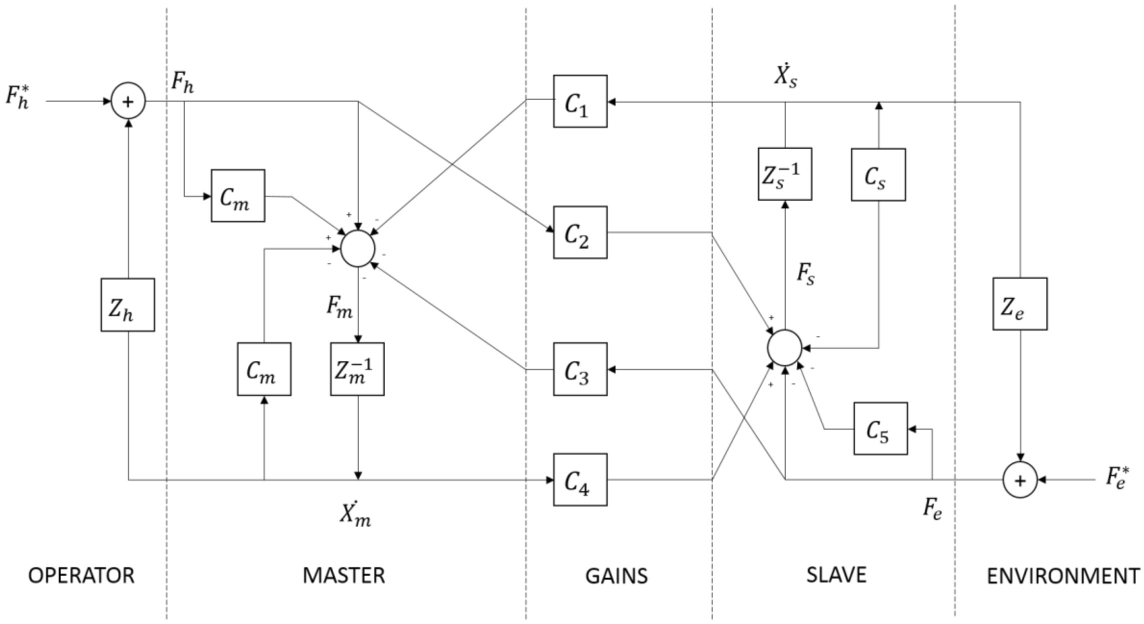

Figure 2.6: Four–channel architecture scheme

This kind of teleoperated control is characterized by four–communication channel, two of them used for communicating position/velocity information and the other two for force information. As is possible to see in the figure, can be identified five different modules:

- Operator, where Z_h represents the transfer function describing the hand of the user who interacts with the master.

In this module is present F_h^* that represents the hexogen forces, intended as external noises.

- Master, includes the control specific for the master robot, and the function that describes the robot
- Control, each block represents a kind of control specific for the kind of information
- Slave, includes the control specific for the slave and the function that describes the robot
- Environment, described by its transfer function Z_e. Also here is present a disturbing forces F_e^* .

Resolving the dynamic system is possible to obtain the follow result:

$$D = -C_3 C_4 + z_{ts}(1 + C_6)\ dove\ z_{ts} = z_s + C_s$$

$$H_{11} = (z_{ts} z_{tm} + C_1 C_4)/D$$

$$H_{12} = [z_{ts} C_2 - (1 + C_5) C_4]/D$$

$$H_{21} = -[z_{tm} C_3 + (1 + C_6) C_1]/D$$

$$H_{22} = -[C_2 C_3 - (1 + C_5)(1 + C_6)]/D$$

This result points out how selecting appropriately values is possible to achieve the perfect condition of transparency. For example, this ideal condition is achieved choosing $C_1 = z_{ts}, C_2 = 1 + C_6, C_3 = 1 + C_5\ e\ C_4 = -z_{tm}$.

From the general four-channel architecture, we can refer to simpler architecture at two or three channel.

The position Error Based (PEB) [30] Control is a two-channel communication control in which only position information are exchanged between master and slave. The reduction of number of channel lead to an inevitable loss of transparency capacity of the system.

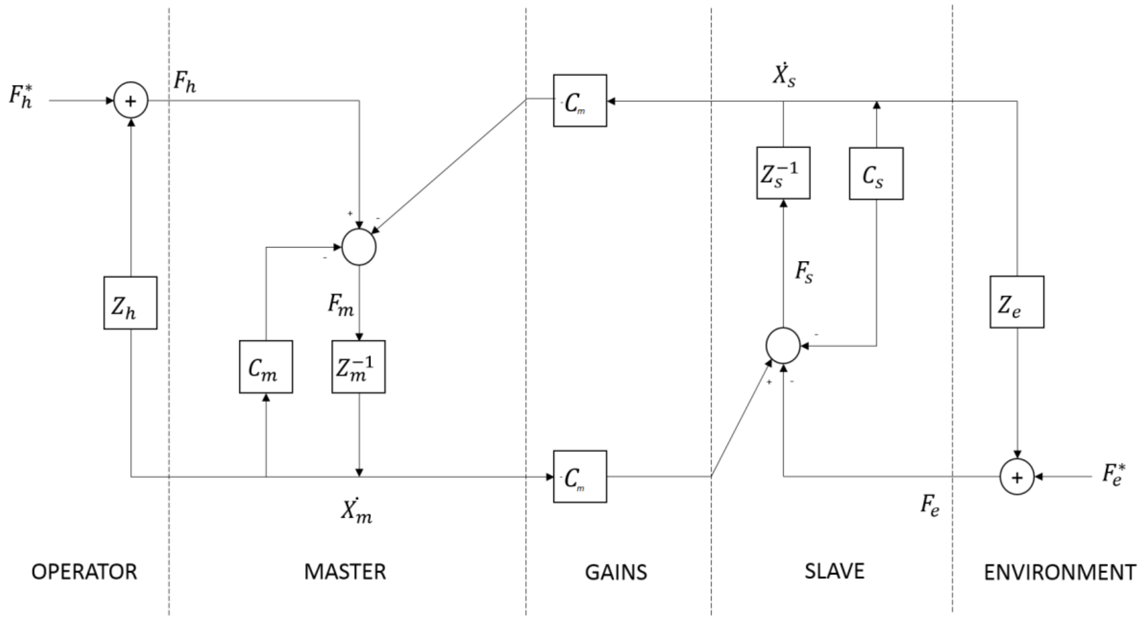

Figure 2.7: Position Error Based Architecture scheme

In fact, resolving the system the following expression of hybrid matrix components are found:

$$H_{PEB} = \begin{bmatrix} z_m + \dfrac{c_m z_s}{z_{ts}} & \dfrac{c_m}{z_{ts}} \\ -\dfrac{c_s}{zts} & \dfrac{1}{z_{ts}} \end{bmatrix} con\ z_{ts} = c_s + z_s\ e\ z_{tm} = c_m + z_m$$

The Force Reflection (RF) (figure 2.8) [31] control was the first control scheme appeared in literature, and the most intuitive and simple. Position information is transmitted from master to slave, and a force feedback from slave to master is present.

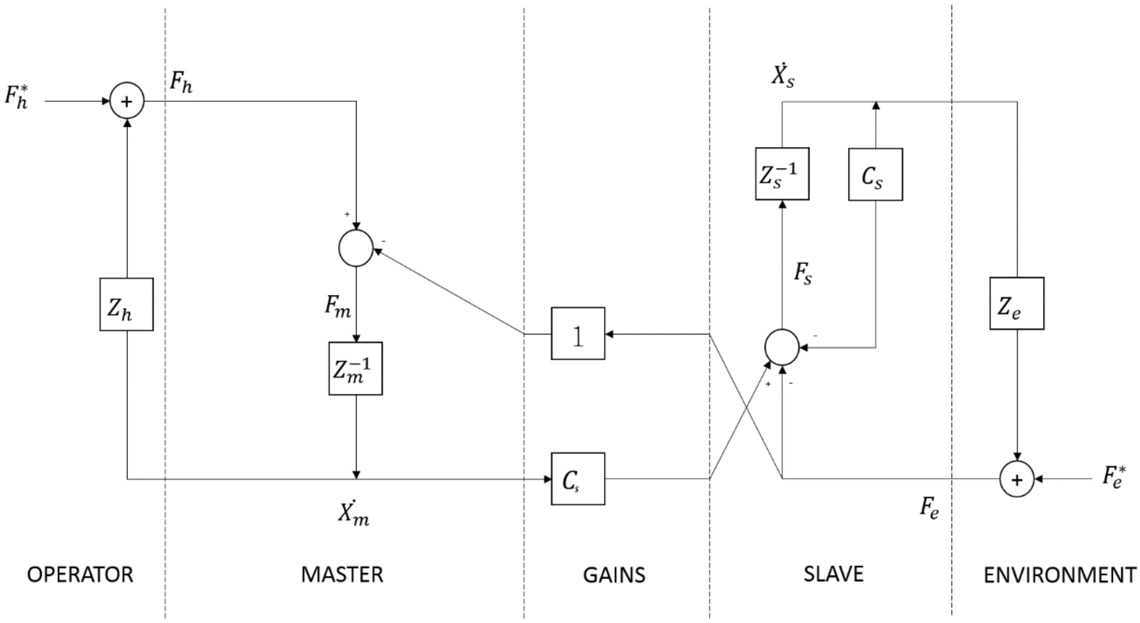

Figure 2.8: Direct Force Reflection Architecture scheme

This control requires a force sensor on the end−effector of the slave robot to apply it properly and allows achieving better results, in terms of transparency, than the PEB control. In fact, the FR' s hybrid matrix is closer to the ideal one:

$$H_{FR} = \begin{bmatrix} z_m & 1 \\ -\dfrac{c_s}{zts} & \dfrac{1}{z_{ts}} \end{bmatrix}$$

The communication between master and slave, especially if occurs through the Internet (LAN, Wi−Fi or 4G), may be affected by delay which, if too high, to system instability. Melchiorri and Arcara made a study concerned the effect of the delay on the classical bilateral control scheme [32]. They referred for the sake of simplicity to a SISO linear dynamic model of the master (m) and slave (s) robots:

$$f_m = (m_m s^2 + b_m s)x_m$$

$$f_s = (m_S s^2 + b_s s)x_s$$

Moreover, it is considered that the forces applied to both the manipulators depend on the "external" interactions (environment and human operator) and on the adopted control scheme. In general, these forces can be written as

$$f_m = f_h - f_{mc}$$

$$f_s = f_e + f_{sc}$$

where f_h, f_e are the forces imposed by the human operator and by the remote environment respectively, while f_{mc}, f_{sc} are the forces computed by the master and slave controllers.

The stability of a telemanipulation scheme is strongly related to the amount of time delay in the transmission channel. In practice, they defined two main cases:

- Intrinsically stable (IS), that is stability is automatically guaranteed independently of time delay.
- Possibly stable (PS) that can be rendered stable, for any value of the delay T.

The last one is the case of the four-channel, position error based and force reflecting control.

Another important property of a telemanipulation system is the ability of the slave device, when it is not in contact with environment, to track as closely as possible the movements of the master.

Scheme	Tracking Error
4C	$\frac{c_2 - c_3}{2(b_m + c_m)}$
PEB	$\frac{1}{2g_c k_c}$
FR	$\frac{b_m + k_c T}{b_m k_c (1 + g_c)}$

Table 2.1: Tracking errors for the different control schemes

As is possible to see from the table above, the PEB and 4C scheme has the advantage that the tracking error is not depended on the time delay T, unlike FR.

Chapter 3

Materials and Methods

This chapter is focused on a detailed introduction to the instruments and devices used to achieve the thesis goals. The two robots that will act as "slave device" and "master device" will be introduced and described.

A brief paragraph will be dedicated analysing the communication system and the embedded system, essential for mounting on a mobile base.

3.1 Slave Device

The Schunk PowerBall LWA 4.6 is a robotic arm designed with industry and research in mind. Its kinematic chain is composed by six degrees-of-freedom (DOF) arranged according to a typical structure of an anthropomorphic manipulator with spherical wrist (Figure 3.1).

Figure 3.1:Schunk Powerball LWA4.6

The Powerball' s six revolute joints are grouped every two into the three silver metallic balls (see figure below). Each of the six points can rotate ±170 degrees and the arm can carry up to 6 kg (13.2lbs) of weight [32].

Unfortunately, none software from the Schunk Company has not been maintained, so the work involved also the development of a firmware and the definition of all the functions that let the robot to be used. Firmware and functions are defined in C++ libraries on a computer transportable platform on which is installed Ubuntu 12.04 as operative system.

This system is very important because lets, in future, to mount the robotic arm on a mobile platform. This system communicates directly with robot initially using the CANOpen Protocol, but subsequent communication problem with the joints have led to use an old version of protocol provide by the Schunk, the Schunk Motion Protocol.

In order to achieve the objectives desired of grasping and screwing, a gripper is attached to the end effector. The gripper used is the Schunk PG−plus 70, a servo−electric 2 finger parallel gripper [33] with sensitive grip force control and long stroke, with control regulating and power electronics completely integrated. For this gripper, specific fingers have been designed and manufactured using a fused deposition 3D printer.

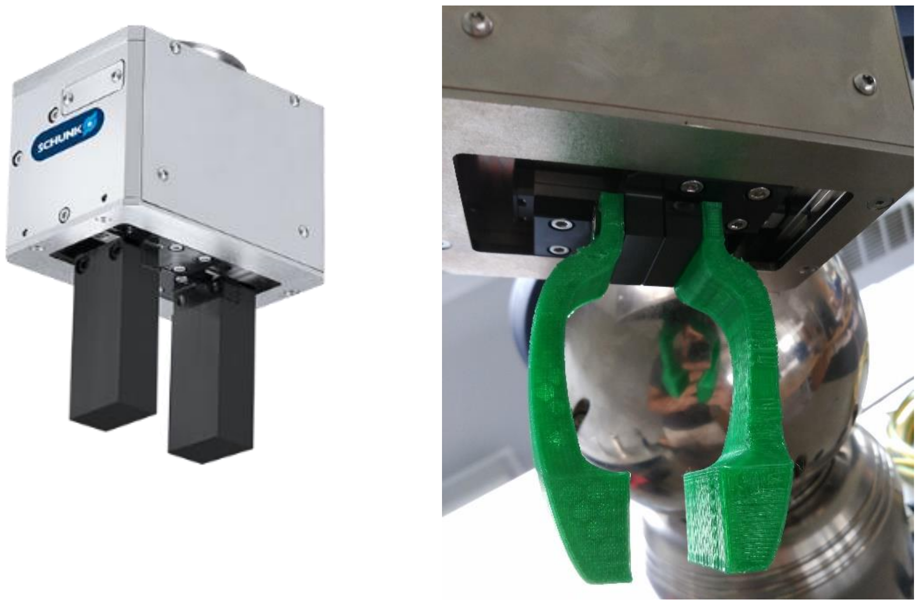

Figure 3.2: Schunk PG−plus 70 gripper and 3D printed fingers

3.1.1 Forward Kinematic

The mechanical structure of a manipulator is characterized by a number of DOF that determinate how the joints and links are arranged in the space. The Forward Kinematic (FK) aims to determinate the pose of the end effector in function of the values of the joints. Defined a base frame $O_b - x_b y_b z_b$, the FK function is expressed by the following homogenous matrix transformation [34]:

$$T_e^b(q) = \begin{bmatrix} n_e^b(q) & s_e^b(q) & a_e^b(q) & p_e^b(q) \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

which returns the Cartesian position and orientation of the end effector respect the joint configuration.

The FK' s calculus is a recursive process made by simple multiplication between the homogeneous matrix transformations $A_i^{i-1}(q_i)$ with $i = 1,2, \dots \dots, n.$ The i*th* homogeneous matrix represents position and orientation of the i*th* frame, referred to the *i−1th* link, respect to the *i−1th* frame, which referred to the *i−1th* link. A commonly used convention for selecting frame of reference in robotic applications is the Denavit−Hartenberg (DH) convention. In this convention, each homogeneous matrix transformation $A_i^{i-1}(q_i)$ is represented as a product of four basic transformations [35]:

$$A_i^{i-1}(q_i) = R_{z,\vartheta_i} \cdot Trans_{z,d_i} \cdot Trans_{x,a_i} \cdot R_{x,\alpha_i}$$

Where the four quantities $\vartheta_i, a_i, d_i, \alpha_i$ are parameters associated with link I and joint i. The four parameters $a_i, \vartheta_i, d_i, \alpha_i$ are generically given the names link length, link twist, link offset and angle joint, respectively. These names derive from specific aspects of the geometric relationship between two coordinate frames. Since the matrix $A_i^{i-1}(q_i)$ is a function of a single variable, it turns out that three of the above four parameter are constant for a given link, while the fourth (because there are only revolute joints) is the joint variable. Through this method is so possible to calculate the FK function, defining the parameters described above. The final matrix that describes the pose of the end-effector according to the joint variables is:

$$T_e^b(q) = T_0^b(q) \cdot [A_1^0(q_1) \cdot A_1^1(q_2) \cdot \ldots \cdot A_n^{n-1}(q_n)] \cdot T_e^n(q)$$

About the Schunk Powerball, the DH parameters that described its kinematic are represented into the table below:

Joint Axis	$d_i(m)$	$a_i(m)$	$\alpha_i(rad)$	$\vartheta_i(rad)$
1	0.205	0	$-\frac{\pi}{2}$	ϑ_1
2	0	0.350	π	$\vartheta_2 - \frac{\pi}{2}$
3	0	0	$-\frac{\pi}{2}$	$\vartheta_3 - \frac{\pi}{2}$
4	0.305	0	$\frac{\pi}{2}$	ϑ_4
5	0	0	$-\frac{\pi}{2}$	ϑ_5
6	0.225	0	0	ϑ_6

Table 3.1: Schunk Powerball LWA4.6 Denavit-Hartenberg parameters

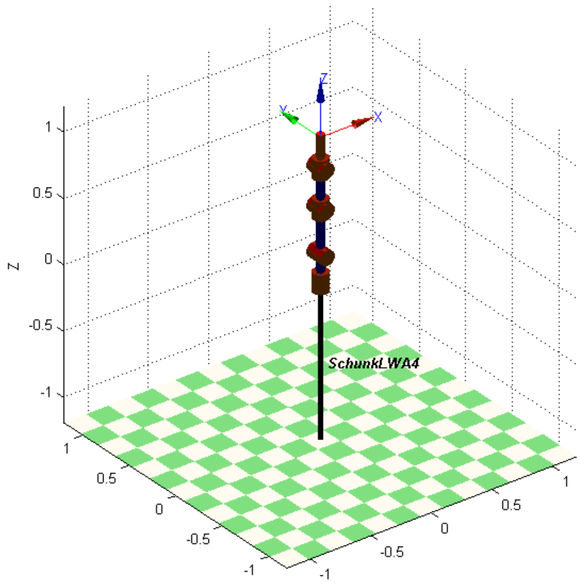

Figure 3.3: Schunk Powerball LWA4.6 kinematic chain (obtained using Robotics Toolbox for MATLAB[9])

3.1.2 Singularities Analysis

A complete study of the kinematic requires a singularity analysis. The particular structure of the robot lets also to decoupling this study in:

- Calculus of singularities for the base structure (first three links).
- Calculus of the singularities for the wrist (last three links).

The study of the singularities starts from the calculus of the Jacobian. The Jacobian is an N×6 matrix, depending only by the

[9] The Toolbox, created by Peter Corke, provides many functions that are useful for the study and simulation of classical armtype robotics, for example such things as kinematics, dynamics, and trajectory generation. The Toolbox is based on a very general method of representing the kinematics and dynamics of serial-link manipulators [link Peter corke]

configuration, $J(q)$, (N indicates the number of DOF of the manipulator) that creates a linking between the space of the end-effector velocities and the space of the joint velocities:

$$v = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J(q)\dot{q}$$

Their studying and characterization is of great importance because:

- They represents configuration in which a leak of robot mobility is observed
- In this configuration infinite solution of the IK problem could exist
- Around a singularity point, low velocities in the Operative Space lead to very high velocities into the Joint Space. This is due to the Jacobian whose determinant drifts towards zero.

The last point suggests that the easiest way to identify a singularity configuration is to study carefully the kinematic structure and looking for the particular configuration in which the Jacobian's determinant is zero.

Taking advantage of the decoupling is possible to identify the following singularities configuration [36]:

$$sin(\theta_3) = 0 \qquad \textbf{elbow singularity}$$

$$a_2 \cos(\theta_2) + d_4 \cos(\theta_3 + \theta_2) = 0 \qquad \textbf{shoulder singularity}$$

$$\theta_5 = 0, \pi \qquad \textbf{wrist singularity}$$

Avoiding this angle values we are sure that we are not in a singularity configuration.

3.1.3 Inverse Kinematic

Once defined the Denavit–Hartenberg parameters and introduced the FK function for the Schunk Powerball, let discuss about the Inverse Kinematic (IK) problem.

The FK defined position and orientation of the end–effector according to joint configuration, a vector whose elements represent the value in radiant of each joint, the solution of the IK problem provides the joint configuration depending on the pose of the end–effector, namely the values the each joint variable should have to reach the exact position and orientation desired. The resolution of this problem is important to translate the motion commands, assigned in the Operative Space and referred to the end–effector, in corresponding motions in the Joint Space that allow to carry out the desired motion.

The IK solution is not easy to find it, this for several reasons [34]:

- The equations are not linear and not always exist an analytical solution.
- Could exist multiple solutions.
- Could exist infinite solutions, like the case of the redundant manipulator.
- Could not exist any solution, because is outside the manipulator's workspace.

Generally, there are two different way to resolve the IK problem:

- Analytical Method: allows to determine a close form solution using algebraic and geometric intuitions for the individuation of significant points thank which is possible to represent position and/or orientation in a less joint variable number. It determines all the possible solutions.

- Numerical Method: used when is not possible or not easy to find a solution applying the analytical method, effectively is possible to apply it to every kinematic chain. This method finds an approximate solution using an iterative process. The most famous algorithm are the inverse, pseudoinverse and transpose of the Jacobian.

About the Schunk Powerball, the IK problem has been resolved using the analytical method, also thanks to its anthropomorphic kinematic chain with spherical wrist. In order to find the analytical solution, the Pieper’ s approach is presented.

The Pieper’ s approach

The kinematic structure of many manipulators is organized in way to separate the IK problem in two subproblem [37] [38]:

first subproblem: inverse solution for the Cartesian position of the end-effector.

second subproblem: inverse solution for the Orientation of the end-effector.

Under these conditions, we say that the manipulator is decoupled, and if the manipulator has six DOF, the possible solutions pass from 16 to 8: there are four different postures of the first three joints and for each solution there exist two possible solutions for the last three axes for a specified end-effector orientation [39]. Pieper affirmed that for six DOF manipulator exists a closed form solution if one of these conditions is satisfied:

- Three consecutive joints whose axes of rotation intersect in a point.
- Three consecutive joints whose axes of rotation are parallel.

Looking the Figure 3.3 is possible to see how the Schunk Powerball' s kinematic chain respects the first condition. Thus, the inverse kinematics calculation could be split up into a position and an orientation subproblem that simplifies the calculation.

First subproblem

The first step is calculating the centre of the wrist. Starting from knowledge of the T_6^0 homogenous matrix transformation and the DH parameters table, the centre of the wrist depends only from the first three joint variables:

$$p_w(q_1, q_2, q_3) = p - d_6 a$$

where:

$$p = T_6^0(1:3,4)$$
$$a = T_6^0(1:3,3)$$

Find the wrist position is possible to resolve the first subproblem and find the solution for q_1, q_2, q_3 .

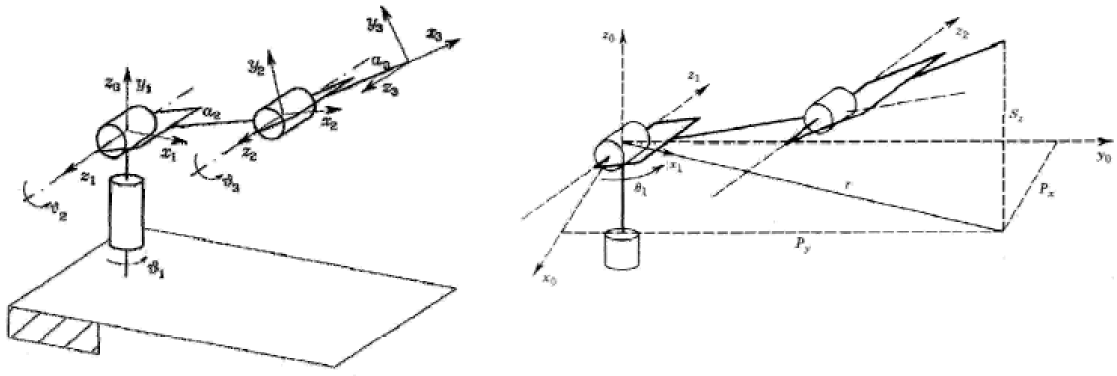

Figure 3.4: Base structures scheme

Observing the figure 3.3 geometrically is possible to determinate the first unknown q_1:

$$\boldsymbol{q_{1,1} = atan2(p_{w,y}, p_{w,x})}$$

$$\boldsymbol{q_{1,2} = \pi - \ atan2(p_{w,y}, p_{w,x})}$$

Once determinate q_1, the structure will be completely planar respect q_2, q_3. So is possible to use the well know solution for a two link planar manipulator:

$$c_3 = \frac{p_{w,x}^2 + \ p_{w,y}^2 + \ p_{w,z}^2 - \ a_2^2 - \ d_4^2}{2a_2 d_4}$$

$$s_{3,1} = +\sqrt{1 - \ c_3^2};\ s_{3,2} = -\sqrt{1 - \ c_3^2}$$

$$s_{2,1} = \frac{(a_2 + d_4 c_3)\left(p_{w,z} - d_1\right) + d_4 s_{3,2}\sqrt{p_{w,x}^2 + \ p_{w,y}^2}}{p_{w,x}^2 + \ p_{w,y}^2 + \ p_{w,z}^2}$$

$$s_{2,2} = \frac{(a_2 + d_4 c_3)(p_{w,z} - d_1) - d_4 s_{3,2}\sqrt{p_{w,x}^2 + p_{w,y}^2}}{p_{w,x}^2 + p_{w,y}^2 + p_{w,z}^2}$$

$$s_{2,3} = \frac{(a_2 + d_4 c_3)(p_{w,z} - d_1) - d_4 s_{3,1}\sqrt{p_{w,x}^2 + p_{w,y}^2}}{p_{w,x}^2 + p_{w,y}^2 + p_{w,z}^2}$$

$$s_{2,4} = \frac{(a_2 + d_4 c_3)(p_{w,z} - d_1) + d_4 s_{3,1}\sqrt{p_{w,x}^2 + p_{w,y}^2}}{p_{w,x}^2 + p_{w,y}^2 + p_{w,z}^2}$$

$$c_{2,1} = \frac{-(a_2 + d_4 c_3)\sqrt{p_{w,x}^2 + p_{w,y}^2} + d_4 s_{3,2}(p_{w,z} - d_1)}{p_{w,x}^2 + p_{w,y}^2 + p_{w,z}^2}$$

$$c_{2,2} = \frac{(a_2 + d_4 c_3)\sqrt{p_{w,x}^2 + p_{w,y}^2} + d_4 s_{3,2}(p_{w,z} - d_1)}{p_{w,x}^2 + p_{w,y}^2 + p_{w,z}^2}$$

$$c_{2,3} = \frac{-(a_2 + d_4 c_3)\sqrt{p_{w,x}^2 + p_{w,y}^2} + d_4 s_{3,1}(p_{w,z} - d_1)}{p_{w,x}^2 + p_{w,y}^2 + p_{w,z}^2}$$

$$c_{2,4} = \frac{(a_2 + d_4 c_3)\sqrt{p_{w,x}^2 + p_{w,y}^2} + d_4 s_{3,1}(p_{w,z} - d_1)}{p_{w,x}^2 + p_{w,y}^2 + p_{w,z}^2}$$

$$\boldsymbol{q_{3,1} = atan2(s_{3,1}, c_3)}; \qquad \boldsymbol{q_{3,2} = atan2(s_{3,2}, c_3)}$$

$$\boldsymbol{q_{2,1} = atan2(s_{2,1}, c_{2,1})}; \qquad \boldsymbol{q_{2,2} = atan2(s_{2,2}, c_{2,2})}$$

$$\boldsymbol{q_{2,3} = atan2(s_{2,3}, c_{2,3})}; \qquad \boldsymbol{q_{2,4} = atan2(s_{2,4}, c_{2,4})}$$

Calculated the first three variable is possible to obtain $T_3^0(q_1, q_2, q_3)$, and pass to next step.

Second subproblem

Knowing the T_3^0 and T_6^0 is easily compute the T_6^3:

$$T_6^0 = T_3^0 \cdot T_6^3$$

$$T_6^3 = (T_3^0)^{-1} \cdot T_6^0 = T_0^3 \cdot T_6^0$$

From this matrix is possible to extract the rotation matrix R_6^3 , thank to which is possible to compute the remaining joint variables.

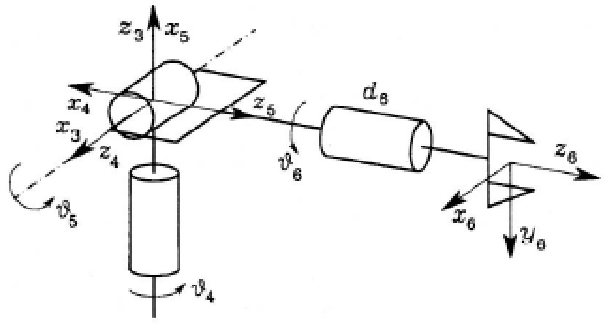

Figure 3.5: Spherical wrist scheme

Effectively, these angles correspond to the Euler angles ZYZ respect the frame 3.

$$R_6^3 = \begin{bmatrix} n_x^3 & s_x^3 & a_x^3 \\ n_y^3 & s_y^3 & a_y^3 \\ n_z^3 & s_z^3 & a_z^3 \end{bmatrix}$$

$$\begin{cases} q_4 = atan2(a_y^3, a_x^3) \\ q_5 = atan2(\sqrt{(a_x^3)^2 + \left(a_y^3\right)^2}, a_z^3) \\ q_6 = atan2(s_z^3, -n_z^3), \quad if\ q_5 \,\epsilon\, (0, \pi) \end{cases}$$

$$\begin{cases} q_4 = atan2(-a_y^3, -a_x^3) \\ q_5 = atan2(-\sqrt{(a_x^3)^2 + \left(a_y^3\right)^2}, a_z^3) \\ q_6 = atan2(-s_z^3, n_z^3), \quad if\ q_5 \,\epsilon\, (-\pi, 0) \end{cases}$$

In this way, the IK problem is resolved obtaining eight solution, corresponding to different configuration of the elbow and the wrist, but only one of them could be used. A method for the chosen of the best solution has been implemented considering important aspects:

1) Nearest joint configuration for the previous one;
2) Joint configuration that does not lead to singularity configuration.

For the point 1), the nearest joint configuration is chosen computing the difference between the norm of each solution vector and the norm of the previous joint configuration. The solutions are sorted in ascending order of norm and the one corresponding to the lower value represents the final solution. Before sending the command to the robot, you ensure that this configuration does not lead the robot to a singularity configuration. If should not be like this, the next immediately solution is chosen, and go on. Below is represent a summary scheme:

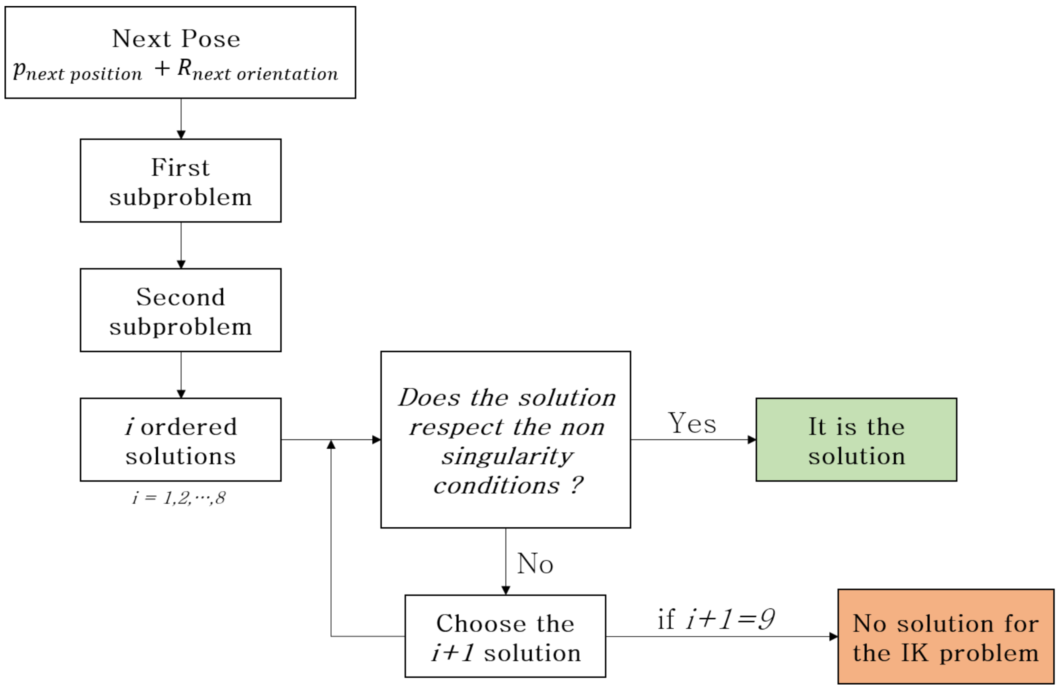

Figure 3.6: IK problem solution scheme

3.2 Master Device

The Master device of the bilateral telemanipulation system is a serial robot, the Geomagic® Touch™ (PHANToM Omni) [40], shown in figure 3.7. PHANToM Omni is a six DOF positional sensing robotic arm, with a range of motion like hand movement pivoting at wrist. The wrist movement are reproduced by the gimbal angles show in Figure 3.6.

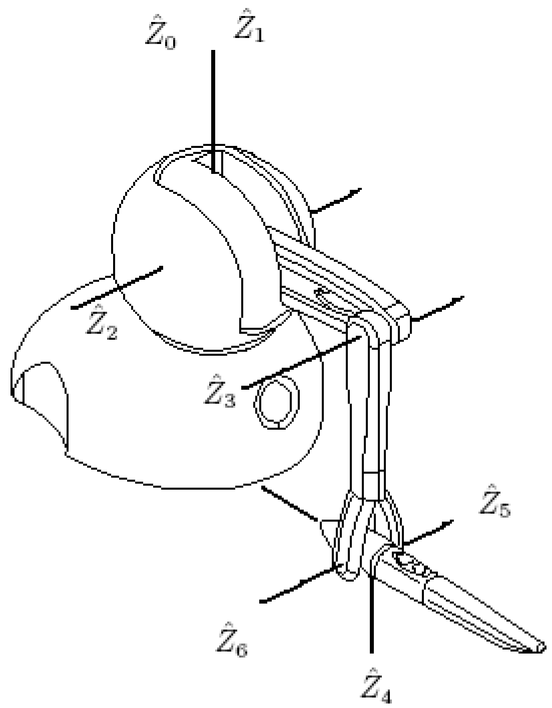

Figure 3.7: Geomagic® Touch™ (PHANToM Omni) with axis

The most important feature is that it is a motorized device that applies force feedback on the user' s hand, allowing them to feel virtual objects and producing true-to-life touch sensations. It is actuate only on three DOF and it is able to reproduce up to a end-effector nominal force of 3.3 N [41].

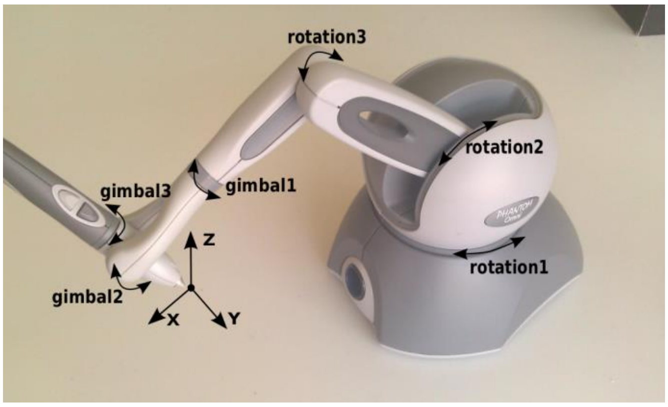

Figure 3.8: PHANToM Omni: gimbal angles and degrees of freedom

In Table 3.2 are shown complete technical specifications [42].

	Geomagic® Touch™
Workspace	~6.4 W x 4.8 H x 2.8 D in > 160 W x 120 H x 70 D mm
Range of motion	Hand movement pivoting at wrist
Nominal position resolution	> 450 dpi ~0.055 mm
Maximum exercisable force and torque at nominal position (orthogonal arms)	0.75 lbf/3.3 N
Stiffness	x−axis > 7.3 lb/in (1.26 N/mm) y−axis > 13.4 lb/in (2.31 N/mm) z−axis > 5.9 lb/in (1.02 N/mm)
Force feedback (6 Degrees of Freedom)	x, y, z
Position sensing/input (6 Degrees of Freedom) [Stylus gimbal]	x, y, z (digital encoders) [Roll, pitch, yaw (± 5% linearity potentiometers)]
Interface	RJ45 Compliant Ethernet Port

Table 3.2: Geomagic® Touch™ technical specification

The two buttons placed on the stylus of the haptic interface can be used for different purposes. In this case, they have been used for commanding the gripper, specifically the white button opens the gripper and the gray one closes it.

Unlike the slave robot, the Geomagic® Touch™ provides all the libraries for programmers into the OpenHaptics Toolkit [43]. This toolkit includes the Haptic Device API (HDAPI), the Haptic Library API (HLAPI), utilities, PHANTOM Device Drivers (PDD), and source code examples (figure 3.9).

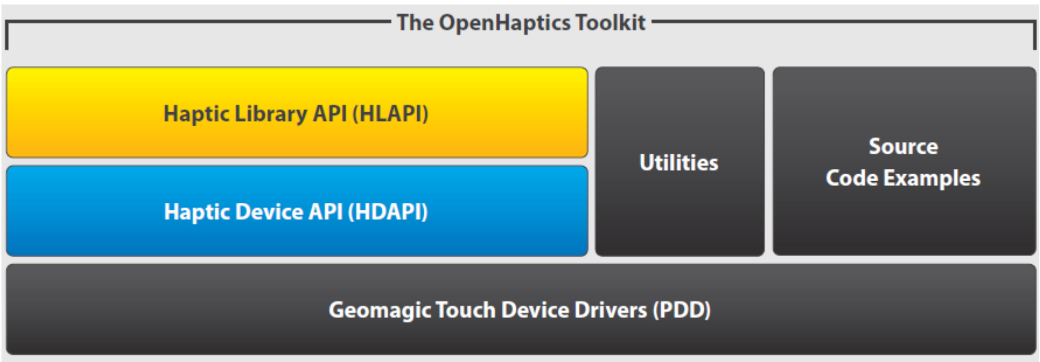

Figure 3.9: OpenHapctics Toolkit

The HDAPI provides low-level access to the haptic device, enables haptics programmers to render forces directly, offers control over configuring the runtime behaviour of the drivers, and provides convenient utility features and debugging aids. The HLAPI provides high-level haptic rendering and is designed to be familiar to OpenGL API programmers. It allows significant reuse of existing OpenGL code and greatly simplifies synchronization of the haptics and graphics threads. The Device Drivers support all 3D Systems' Touch and Phantom haptic devices.

3.2.1 Forward Kinematic

In order to describe the kinematics of the PHANTOM, the frames locations must be determined. Figure 3.8 shows the PHANTOM with the frame axes fixed onto it.

While it is useful to see where the frames t on the actual robot, this can prove to be cumbersome to analyze. Figure 3.10 shows just the axes with the respective DH parameters. The DH parameters determined, as listed in Table 3.3

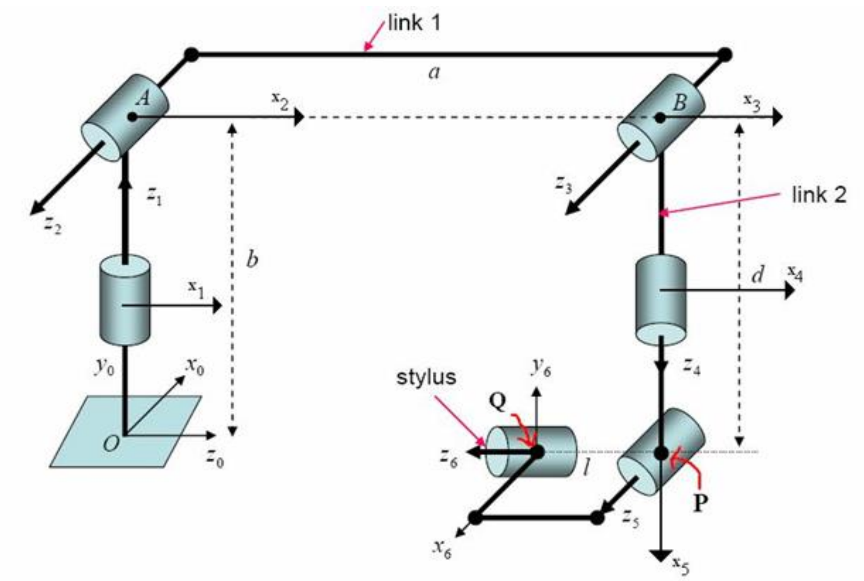

Figure 3.10: PHANToM Omni kinematic chain

Joint Axis	$d_i(m)$	$a_i(m)$	$\alpha_i(rad)$	$\vartheta_i(rad)$
1	0	0	0	ϑ_1
2	0	0	$\frac{\pi}{2}$	ϑ_2
3	0	0.1275	0	ϑ_3
4	0.149	0	$\frac{\pi}{2}$	ϑ_4
5	0	0	$-\frac{\pi}{2}$	ϑ_5
6	0	0	$\frac{\pi}{2}$	ϑ_6

Table 3.3: PHANToM Omni Denavit–Hartenberg parameters

With the DH parameters and this relation to the workspace known, it is possible to formulate the forward kinematics [41].

3.3 GUI–Graphical User Interface

The Graphical User Interface (GUI) has been implemented and designed using a specific software QtCreator, which lets to create and integrate GUI in C++ programming language.

Through the GUI is possible to control completely the robot selecting from three possibilities:

- Moving the robot using the Geomagic® Touch™ Omni Phantom
- Moving the robot along the direction x, y and z of the base robot frame. This feature performs linear trajectories
- Moving the robot into the Joint Space, controlling it joint–by–joint.

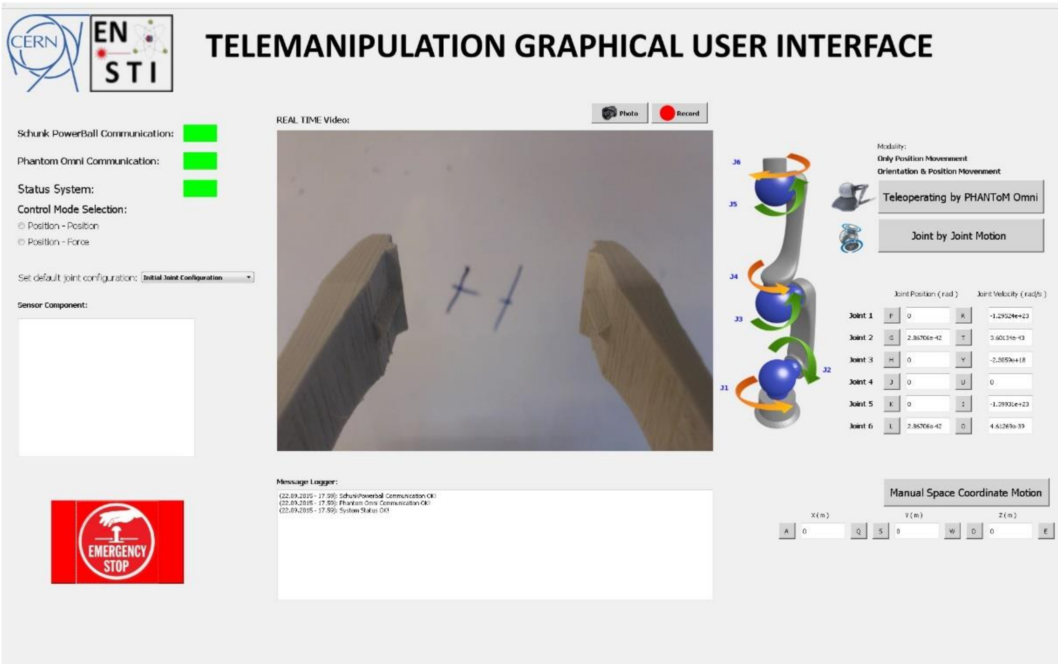

Figure 3.13: Telemanipulation Graphical User Interface

Other features of the interface include the possibilities of monitoring the correct execution of the task through the central windows which returns the video from a camera located on the top of the gripper, a window logger that memorize every command send, different led that supervise the communication between master and slave, and, at last, a red button to activate in emergency case pressing a specific button on the keyboard. Is also possible to set the sensibility of the movement: the master robot position are sent premultipling by a scale factor, which can be properly set in real time to obtain movement with more or less precision.

To the GUI has been joined a simulation system creating using the CAD software Autodesk Inventor (Figure 3.12).

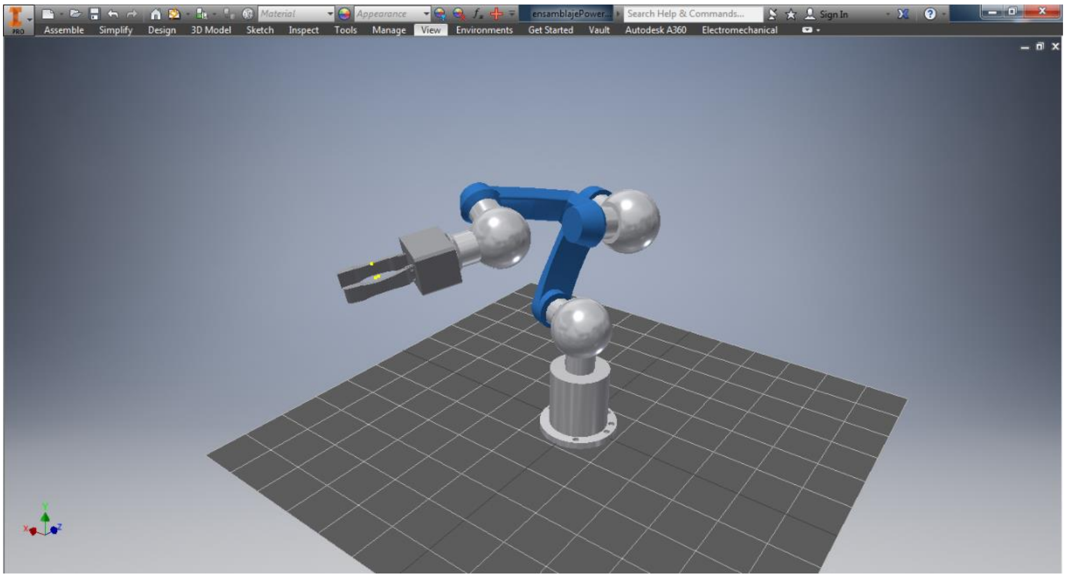

Figure 3.12: Schunk Powerball LWA4.6 CAD

The system replace on the screen the real robot movement. This is important because using only a single camera located on the gripper, only the end effector is visible, but the user will be completely blind about what happen behind the camera and especially what is the conformation of the entire robot. In this way, the user has a feedback about the entire disposition of the robot into the space and will be able to modify it avoiding eventual collisions.

3.3 Communication System

All the elements described so fare are linked to each other thanks to the communication system.

The communication system, implemented in C++, uses the Ethernet system to connect the devices and the Socket TCP/IP for the communication. Due to the different option useable to move the robot, every information exchanged between Master and Slave is labelled with a specific name (Figure 3.12).

```
#include <matrix.h>

#ifndef PACKET_HPP
#define PACKET_HPP


const unsigned short ARM_JOINT_BY_JOINT_PACKET = 1;
const unsigned short ARM_WORLD_COORDINATE_PACKET = 2;
const unsigned short ARM_WORLD_POSITION_PACKET = 3;
const unsigned short ARM_WORLD_FORCE_PACKET = 4;
const unsigned short ARM_JOINT_CONFIG_PACKET = 5;

struct Packet {

};

```

Figure 3.12: Code stretch

Particularly, to each label correspond a specific data packet. For example, to `ARM_JOINT_BY_JOINT_PACKET` corresponds an array containing the six values of each joint, or to `ARM_WORLD_COORDINATE_PACKET` corresponds the homogeneous matrix transformation 4-by-4 containing the position and orientation information.

The information travel into Ethernet Network through two-communication port, one connected directly with GUI, and the other connect with simulation system. In the last case, the communication is unidirectional, from the slave robot information about the joint values are sent.

The figure below represents the entire system.

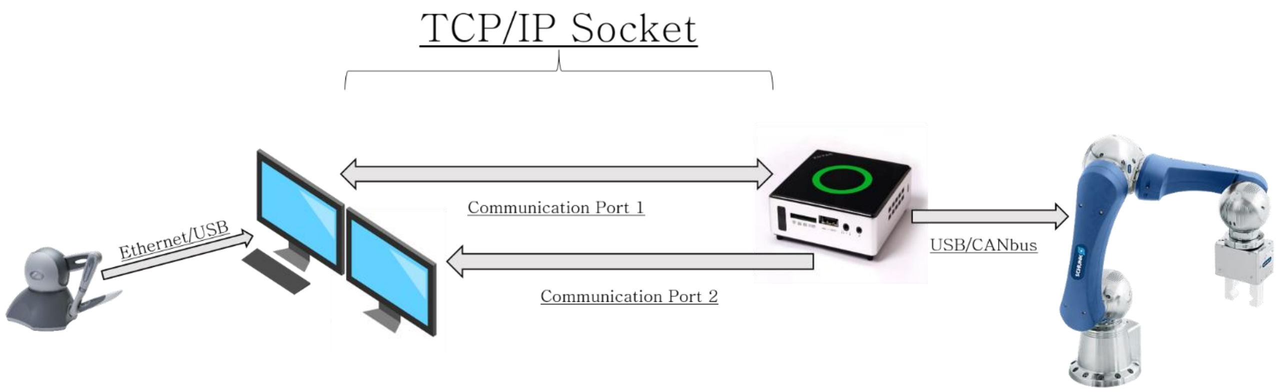

Figure 3.13: Communication system scheme

Chapter 4

Implementation & Results

Main topic of this first part of the chapter is the description of the teleoperated control system, its architecture and software implementation.

The second part concerns the exposition of the results obtained through different tests and tasks simulation.

4.1 Teleoperated Control System

As described into the paragraph 2.3.2, the literature offers a lot of different possibilities of creating a master–slave teleoperated control system. The choice depends by different factors like the task for which the system is designed for, the kinds of robot available. As widely described into the previous chapter, the slave robot is not equipped with a torque/force sensor on the gripper; this led to exclude a priori any kind of architectures that provide a force feedback from the slave side to the master one. Due to this detail, the Position Error Based (PEB) control was chosen.

4.1.1 Architecture

In the figure below is represented the entire teleoperated system.

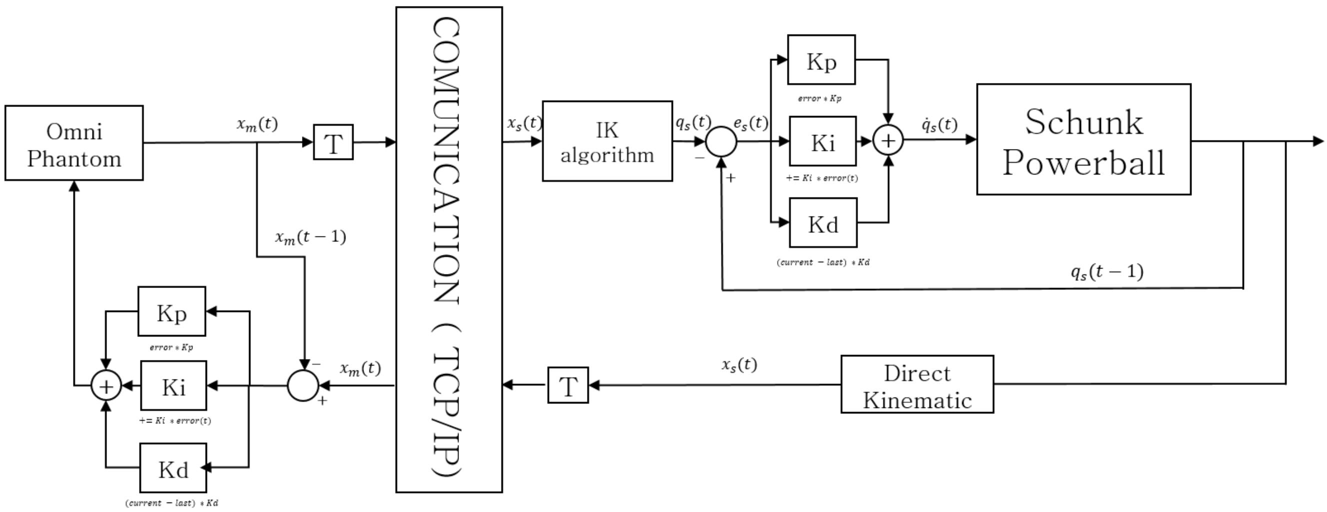

Figure 4.14: Teleoperated control system scheme

The communication block divides the scheme in two macro-blocks, the master on the left side and the slave on the right side.

The loop starts sending the information about the master device pose (position + orientation), crossing, before, a block denotes with the letter "T". This block is representative of a transformation from the master frame to the slave frame in order to make the data information usable by the slave robot. The processed information is sent through the communication block to the slave side: it represents the next pose.

The "IK Algorithm" block receives the next pose information and provides the best next joint configuration, which constitutes the necessary input for the control loop of the slave robot.

The choice of the control loop was significantly affected by the impossibility to send force/torque input to the Schunk Powerball. For that reason for the control of the slave robot, the choice fell on a PID control in voltage/velocity in Joint Space.

The PID control (Proportional-Integral-Derivative control) is a control loop feedback mechanism commonly used in industrial control system. It calculates, continuously, an error value as the difference between a measured processed variable and a desired value. In this case, a vector containing the value of each joint represents the measured processed variable, while the desired value is the next joint configuration, the output of the "IK Algorithm" block. The controller attempts to minimize the error providing a velocity input signal to the robot composed by a vector containing the velocity of each joint. The equation below is representative of the controller:

$$\dot{q}_s(t) = K_p e_s(t) + K_\iota \int_0^T e_s(t)dt + K_d \frac{d\dot{e}_s(t)}{dt} \quad where \;\; e_s(t) = q_s(t) - q_s(t-1)$$

The K_p, K_i and K_d are all non-negative coefficients for the proportional, integral and derivative terms. Each term plays a specific role [44]:

- The proportional term is the only one truly indispensable for the error minimizing. Higher is the value of its component, faster the error will tend to zero value, but too high value greater oscillations around the value zero will be observed.
- The integral term, at last, accelerates the movement of the process towards setpoint and eliminates the residual steady-state error that occurs with a pure proportional controller.
- The derivative term has the role to dampen oscillations in the transient improving settling time and stability of the system.

Following with the description of the scheme, the actual joint configuration of the slave robot is sent to the "Direct Kinematic" block, which provides its actual pose. The actual pose is submitted a frame transformation, and used on the master side to calculate the error. As on the slave side, also here a PID controller is used to minimize the error, but two features distinguishes them:

- The slave error is computed into the Joint Space; instead, the master error is computed into the Operative Space.
- The slave has a PID control in voltage/velocity; instead, the master has a PID controller in torque/force.

The PID controller on the master, so, provides a force that represents the force feedback. The force feedback is a restoring force that tends to lead the Stylus in the point of the Master device space corresponding to the position of the Slave robot. In this way, if the user moves the stylus in a proper way, with a velocity through which the slave could track the position in a faster way, the force perceived by the user is almost null, but if the velocity of the stylus imposed by the user is too high, the force feedback dampens the velocity. Besides, the restoring force is used also to recall the stylus in the point within the workspace of the slave, when it exceeds this limit. In this condition, the next pose for the slave is not refreshed and the robot remains in the last acceptable pose.

4.1.2 Software Implementation

Here will be shown and described parts of the code corresponding to the block of the scheme in figure 4.1.

```
//POSITION ERROR BASED BILATERAL CONTROL
// ForceFeedback processing
previous_error = error;
for (int i = 0; i < 3; i++){
    force_previous[i] = force[i];
}
pos(0, 0) = pos(0, 0) + delta[2];
pos(1, 0) = pos(1, 0) + delta[0];
pos(2, 0) = pos(2, 0) + delta[1];
error = pos - (previous_pos);
printf("dposition:[1] %f [2] %f [3] %f\n", error(0, 0), error(1, 0), error(2, 0));
derivative = (error - previous_error) * 0.100;
integral = integral + error * 0.100;
force[2] = (error(0, 0)) * Stiffness - (derivative(0, 0) * Damping) + (integral(0, 0) * Ki);
force[0] = (error(1, 0)) * Stiffness - (derivative(0, 0) * Damping) + (integral(0, 0) * Ki);
force[1] = (error(2, 0)) * Stiffness - (derivative(0, 0) * Damping) + (integral(0, 0) * Ki);

for (int i = 0; i<3; i++){
    if (force[i] > 3.0){
        force[i] = 3.0;
    }
    else if (force[i] < -3.0){
        force[i] = -3.0;
    }
}
```

Figure 4.15: PID control and Force Feedback generation on Master side

```
Matrix<float> SchunkArm::forwardKinematic(Matrix<float> joints)
{
    for (int i = 0; i < 6; i++) {
        (*T[i])(0, 0) = cos(joints(i,0) + (*DHParameters)(i, 0));
        (*T[i])(1, 0) = sin(joints(i,0) + (*DHParameters)(i, 0));

        (*T[i])(0, 1) = -sin(joints(i,0) + (*DHParameters)(i, 0))*cos((*DHParameters)(i, 3));
        (*T[i])(1, 1) =  cos(joints(i,0) + (*DHParameters)(i, 0))*cos((*DHParameters)(i, 3));

        (*T[i])(0, 2) =  sin(joints(i,0) + (*DHParameters)(i, 0))*sin((*DHParameters)(i, 3));
        (*T[i])(1, 2) = -cos(joints(i,0) + (*DHParameters)(i, 0))*sin((*DHParameters)(i, 3));

        (*T[i])(0, 3) = (*DHParameters)(i, 2)*cos(joints(i,0) + (*DHParameters)(i, 0));
        (*T[i])(1, 3) = (*DHParameters)(i, 2)*sin(joints(i,0) + (*DHParameters)(i, 0));
    }

    (*T06) = (*T[0])*(*T[1])*(*T[2])*(*T[3])*(*T[4])*(*T[5]);
    return (*T06);
}
```

Figure 4.16: Schunk Powerball Forward Kinematic function

The figure 4.3 describes the calculus for obtaining the forward kinematic of the Schunk Powerball expressed through the 4−by−by homogeneous transformation matrix following the process showed into the paragraph 4.2. Because of the system works in real−time, is convenient use functions as fastest as possible. The multiplication between matrix could be a process slower than multiplication and addition. For that reason, taking advantages of the "*Matlab Symbolic Toolbox*" and of the constant shape of the forward kinematic matrix depending only by the joint variables, the following function has been implemented:

```
Matrix<float> SchunkArm::fastforwardKinematic(Matrix<float> joints)
{
    float c0 = cos(joints(0,0));               float s0 = sin(joints(0,0));
    float c1 = cos(joints(1,0));               float s1 = sin(joints(1,0));
    float c2 = cos(joints(2,0));               float s2 = sin(joints(2,0));
    float c3 = cos(joints(3,0));               float s3 = sin(joints(3,0));
    float c4 = cos(joints(4,0));               float s4 = sin(joints(4,0));
    float c5 = cos(joints(5,0));               float s5 = sin(joints(5,0));
    float c12 = cos(joints(1,0) - joints(2,0)); float s12 = sin(joints(1,0) - joints(2,0));
    float c1p = cos(joints(1,0) - M_PI / 2);    float s1p = sin(joints(1,0) - M_PI / 2);
    float c2p = cos(joints(2,0) - M_PI / 2);    float s2p = sin(joints(2,0) - M_PI / 2);
    Matrix<float> TK_06(4, 4, 0.0);

    TK_06(0,0)= - s5*(c3*s0 + s3*(c0*c1p*c2p + c0*s1p*s2p)) - c5*(c4*(s0*s3 - c3*(c0*c1p*c2p + c0*s1p*s2p)) + s4*(c0*c1p*s2p - c0*c2p*s1p));
    TK_06(0,1)= s5*(c4*(s0*s3 - c3*(c0*c1p*c2p + c0*s1p*s2p)) + s4*(c0*c1p*s2p - c0*c2p*s1p)) - c5*(c3*s0 + s3*(c0*c1p*c2p + c0*s1p*s2p));
    TK_06(0,2)= s4*(s0*s3 - c3*(c0*c1p*c2p + c0*s1p*s2p)) - c4*(c0*c1p*s2p - c0*c2p*s1p);
    TK_06(0,3)= (7*c0*c1p)/20 + (9*s4*(s0*s3 - c3*(c0*c1p*c2p + c0*s1p*s2p)))/40 - (9*c4*(c0*c1p*s2p - c0*c2p*s1p))/40 - (61*c0*c1p*s2p)/200 + (61*c0*c2p*s1p)/200;
    TK_06(1,0)= c5*(c4*(c0*s3 + c3*(c1p*c2p*s0 + s0*s1p*s2p)) - s4*(c1p*s0*s2p - c2p*s0*s1p)) + s5*(c0*c3 - s3*(c1p*c2p*s0 + s0*s1p*s2p));
    TK_06(1,1)= c5*(c0*c3 - s3*(c1p*c2p*s0 + s0*s1p*s2p)) - s5*(c4*(c0*s3 + c3*(c1p*c2p*s0 + s0*s1p*s2p)) - s4*(c1p*s0*s2p - c2p*s0*s1p));
    TK_06(1,2)= - s4*(c0*s3 + c3*(c1p*c2p*s0 + s0*s1p*s2p)) - c4*(c1p*s0*s2p - c2p*s0*s1p);
    TK_06(1,3)= (7*c1p*s0)/20 - (9*s4*(c0*s3 + c3*(c1p*c2p*s0 + s0*s1p*s2p)))/40 - (9*c4*(c1p*s0*s2p - c2p*s0*s1p))/40 - (61*c1p*s0*s2p)/200 + (61*c2p*s0*s1p)/200;
    TK_06(2,0)= c5*(s4*(c1p*c2p + s1p*s2p) + c3*c4*(c1p*s2p - c2p*s1p)) - s3*s5*(c1p*s2p - c2p*s1p);
    TK_06(2,1)= - s5*(s4*(c1p*c2p + s1p*s2p) + c3*c4*(c1p*s2p - c2p*s1p)) - c5*s3*(c1p*s2p - c2p*s1p);
    TK_06(2,2)= c4*(c1p*c2p + s1p*s2p) - c3*s4*(c1p*s2p - c2p*s1p);
    TK_06(2,3)= (9*c4*(c1p*c2p + s1p*s2p))/40 - (7*s1p)/20 + (61*c1p*c2p)/200 + (61*s1p*s2p)/200 - (9*c3*s4*(c1p*s2p - c2p*s1p))/40 + (float)41/200;
    TK_06(3,0)= 0;
    TK_06(3,1)= 0;
    TK_06(3,2)= 0;
    TK_06(3,3)= 1;

    return TK_06;
}
```

Figure 4.17: Schunk Powerball Fast Forward Kinematic function

```
    while (true) {
        (*current) = getJointsCurrent();
        Matrix<float> force = force_ee((*current),(*joint_position));
        (*joint_position) = getJointsPosition();
        bool pos_reached = true;
        (*error) = (*desired_joint_position) - (*joint_position);
        derivative = ((*error) - previous_error)/(0.100);
        integral = integral + ((*error)*0.100);
        Matrix<float> vel = (*error)* 2.5 + derivative * 0.8 + integral * 0.1;
        for(int i = 0; i<7; i++){
            if(vel(i,0) > max_speed){
                vel(i,0) = max_speed;
            }else if( vel(i,0) < -max_speed){
                vel(i,0) = -max_speed;
            }
        }
        setJointsVelocity(vel);
        if (fabs((*error)(0,0))>0.001 && fabs((*error)(1,0))>0.001 && fabs((*error)(2,0))>0.001 && fabs((*error)(3,0))>0.001 &&
            fabs((*error)(4,0))>0.001 && fabs((*error) (5,0))>0.001) {
            pos_reached = false;
        }
        if(!pos_reached){
            position_reached = false;
        }else{
            position_reached = true;
        }

    }
}
```

Figure 4.18: PID control on slave side

Below is represented the implementation of the Inverse Kinematic Algorithm as described in the paragraph 3.1.3

```
Matrix<float> SchunkArm::fastinverseKinematic(Matrix<float> T, Matrix<float> actual_joint)
{
    /* CALCOLO DELLA SOLUZIONE */

    Matrix<float>* sol[8];
    for(int i=0; i<8; i++)
    {
        sol[i] = new Matrix<float>(6,1,0.5);
    }
    bool good;
    float c1 = 0.205;  //m
    float c2 = 0.350;  //m
    float c3 = 0.305;  //m
    float c4 = 0.225;  //m     /* IMPORTANT: Sono stati inseriti 150 mm -> terna sull' end effector */

    float cx0 = (T(0,3) - T(0,2)*c4);
    float cy0 = (T(1,3) - T(1,2)*c4);
    float cz0 = (T(2,3) - T(2,2)*c4);

    float* th1 = new float[2];
    th1[0] = atan2(cy0,cx0);
    th1[1] = atan2(cy0,cx0) - M_PI;
    for(int i = 0; i<4; i++)
    {
        if(th1[i]<th_min && th1[i] < -3.1416)
        {
            th1[i] = th1[i] + 2*M_PI;
        }else if(th1[i]<th_min && th1[i] >= -3.1416)
        {
            th1[i] = th1[i] + M_PI;
        }
        else if(th1[i]>th_max && th1[i] > 3.1416)

        {
            th1[i] = th1[i] - 2*M_PI;
        }else if(th1[i]>th_max && th1[i] <= 3.1416)
        {
            th1[i] = th1[i] - M_PI;
        }
    }
    float cos3 = (pow(cx0,2) + pow(cy0,2) + pow(cz0 - c1,2) - pow(c2,2) - pow(c3,2))/(2*c2*c3);
    float sin3_1 = + sqrt(1 - pow(cos3,2));
    float sin3_2 = - sqrt(1 - pow(cos3,2));
    if(fabs(cos3)>1){
        (*solution) = (*previous_solution);
        printf("Exeed Workspace Limits\n");
        printf("No solution for IK problem!\n");
    }else
    {
        float* th3 = new float[4];
        th3[0] = atan2(sin3_1,cos3);
        th3[1] = atan2(sin3_2,cos3);
        for(int i = 0; i<2; i++)
        {
        if(th3[i]<th_min && th3[i] >= -3.1416)
        {
            th3[i] = th3[i] + M_PI;
        }
        else if(th3[i]>th_max && th3[i] <= 3.1416)
        {
            th3[i] = th3[i] - M_PI;
        }
        }

        float* th2 = new float[4];
        th2[0] = atan2((c2 + c3*cos3)*(cz0 - c1) + c3*sin3_2*sqrt(pow(cx0,2) + pow(cy0,2)),-(c2 + c3*cos3)*sqrt(pow(cx0,2) + pow(cy0,2)) + c3*sin3_2*(cz0 - c1)) -
        th2[1] = atan2((c2 + c3*cos3)*(cz0 - c1) - c3*sin3_2*sqrt(pow(cx0,2) + pow(cy0,2)), (c2 + c3*cos3)*sqrt(pow(cx0,2) + pow(cy0,2)) + c3*sin3_2*(cz0 - c1)) -
        th2[2] = atan2((c2 + c3*cos3)*(cz0 - c1) + c3*sin3_1*sqrt(pow(cx0,2) + pow(cy0,2)),-(c2 + c3*cos3)*sqrt(pow(cx0,2) + pow(cy0,2)) + c3*sin3_1*(cz0 - c1)) -
        th2[3] = atan2((c2 + c3*cos3)*(cz0 - c1) - c3*sin3_1*sqrt(pow(cx0,2) + pow(cy0,2)), (c2 + c3*cos3)*sqrt(pow(cx0,2) + pow(cy0,2)) + c3*sin3_1*(cz0 - c1)) -
        for(int i = 0; i<4; i++)
        {
        if(th2[i]<th_min && th2[i] < -3.1416)
        {
            th2[i] = th2[i] + 2*M_PI;
        }
        else if(th2[i]>th_max && th2[i] > 3.1416)
        {
            th2[i] = th2[i] - 2*M_PI;
        }
        }

        (*sol[0])(0,0) = th1[0]; (*sol[0])(1,0) = th2[0]; (*sol[0])(2,0) = th3[0];
        (*sol[4])(0,0) = th1[0]; (*sol[4])(1,0) = th2[0]; (*sol[4])(2,0) = th3[0];
        (*sol[1])(0,0) = th1[0]; (*sol[1])(1,0) = th2[2]; (*sol[1])(2,0) = th3[1];
```

```
(*sol[5])(0,0) = th1[0]; (*sol[5])(1,0) = th2[2]; (*sol[5])(2,0) = th3[1];
(*sol[2])(0,0) = th1[1]; (*sol[2])(1,0) = th2[1]; (*sol[2])(2,0) = th3[0];
(*sol[6])(0,0) = th1[1]; (*sol[6])(1,0) = th2[1]; (*sol[6])(2,0) = th3[0];
(*sol[3])(0,0) = th1[1]; (*sol[3])(1,0) = th2[3]; (*sol[3])(2,0) = th3[1];
(*sol[7])(0,0) = th1[1]; (*sol[7])(1,0) = th2[3]; (*sol[7])(2,0) = th3[1];
float norm[8];
for(int i = 0; i<4; i++)
{
Matrix<float> R30 = R30_matrix((*sol[i]));
Matrix<float> R06 = T.getPart(0,2,0,2);
Matrix<float> R36 = R30 * R06;
(*sol[i])(3,0)  = atan2(R36(1,2),R36(0,2));
(*sol[i])(4,0)  = atan2(-sqrt(pow(R36(0,2),2) + pow(R36(1,2),2)), R36(2,2)) ;
(*sol[i])(5,0)  = atan2(R36(2,1),-R36(2,0));

(*sol[i + 4])(4,0)  = atan2(sqrt(pow(R36(0,2),2) + pow(R36(1,2),2)), R36(2,2));
(*sol[i + 4])(5,0)  = atan2(-R36(2,1),R36(2,0));
Matrix<float> diff = (*sol[i]) - actual_joint;
Matrix<float> diff2 = diff.getPart(0,2,0,0);
norm[i] = diff.get_norm();
diff = (*sol[i + 4]) - actual_joint;
diff2 = diff.getPart(0,2,0,0);
norm[i+4] = diff.get_norm();
}

/* ALGORITMO DI SCELTA DELLA SOLUZIONE */

int ind[8];
float norm1[8] = {norm[0],norm[1],norm[2],norm[3],norm[4],norm[5],norm[6],norm[7]};

for(int i = 0; i<7; i++)
{
int min = i;
for(int j = i+1; j<8; j++)
{
    if(norm[j] < norm[min])
    {
        min = j;
    }
}
float temp = norm[min];
norm[min] = norm[i];
norm[i] = temp;
}
for(int i = 0; i<8; i++)
{
for(int j = 0; j<8; j++)
{
    if(norm[i] == norm1[j])
    {
        ind[i] = j;
    }
}
}

good = false;
int y = 0;
while(!good)
{
if(ind[y] == 0)
{
    (*solution) = (*sol[0]);
}else if (ind[y] == 1)
{
    (*solution) = (*sol[1]);
}else if (ind[y] == 2)
{
    (*solution) = (*sol[2]);
}else if (ind[y] == 3)
{
    (*solution) = (*sol[3]);
}else if (ind[y] == 4)
{
    (*solution) = (*sol[4]);
}else if (ind[y] == 5)
{
    (*solution) = (*sol[5]);
}else if (ind[y] == 6)
{
    (*solution) = (*sol[6]);
}else if (ind[y] == 7)
{
    (*solution) = (*sol[7]);
}
if((*solution)(0,0) > th_min && (*solution)(0,0) < th_max &&
   (*solution)(1,0) > th_min && (*solution)(1,0) <= 0.9   &&
   (*solution)(2,0) >= 0.0   && (*solution)(2,0) < th_max &&
   (*solution)(3,0) > th_min && (*solution)(3,0) < th_max &&
   (*solution)(4,0) > th_min && (*solution)(4,0) < th_max &&
   (*solution)(5,0) > th_min && (*solution)(5,0) < th_max &&
   virtualWall((*solution)) == true && checkSingolarity((*solution)) == true)
{
    good = true;
}
```

```
        else
        {
            good = false;
        }

        if(y == 7 && good == false)
        {
        (*solution) = (*previous_solution);
        printf("sono a y = 7\n");
        printf("No solution for the IK problem!\n");
        }
        y++;
        }
        if(isnan((*solution)(0,0)) ||isnan((*solution)(1,0)) ||isnan((*solution)(2,0)) ||isnan((*solution)(3,0)) ||isnan((*solution)(4,0)) ||isnan((*solution)(5,0)
        {
        (*solution) = (*previous_solution);

        printf("Robot in Singularity\n");
        }
    }
    (*previous_solution) = (*solution);
    return (*solution);
}
```

Figure 4.19: IK Algorithm for the Schunk Powerball

4.2 Results

This phase could be divided in two different sub−phases:

- The trajectory test phase, used to tune the weights of the PID controller, and verify the repeatability and how the system behaves to different work frequencies, in order to find the maximum work frequency possible.
- The task simulation phase, used to verify how well the robot follows the trajectory imposed by the user while he is fulfilling the task.

4.2.1 Trajectory Test

Tuning coefficient PID control

The tuning of the coefficients is made using a manual method and holding the system online. A sinusoidal trajectory characterized by amplitude **0.3041m** and frequency **40**Hz is generated on the master side as input for the slave robot. The test was repeated using different values of K_p, K_i and K_d.

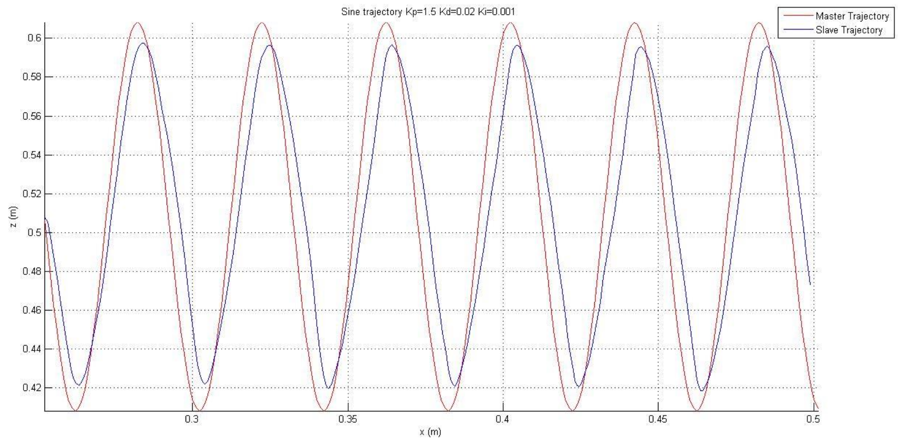

Figure 4.7: Sine trajectory $K_p = 1.5$ $K_i = 0.02$ and $K_d = 0.001$

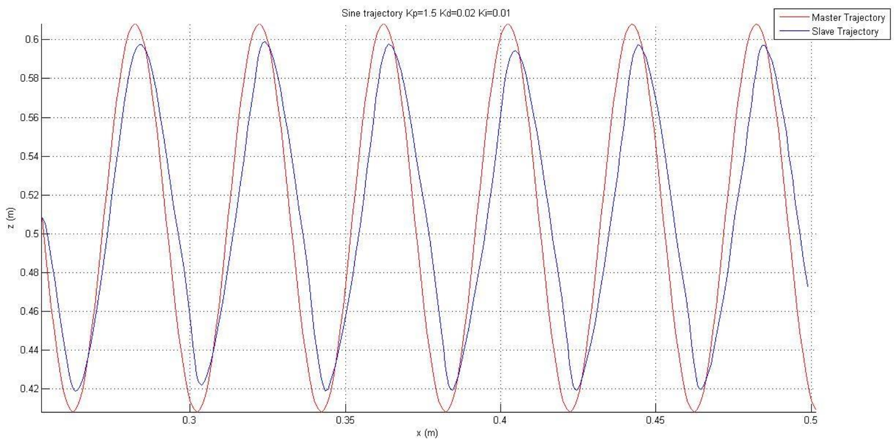

Figure 4.8: Sine trajectory $K_p = 1.5$ $K_i = 0.02$ and $K_d = 0.01$

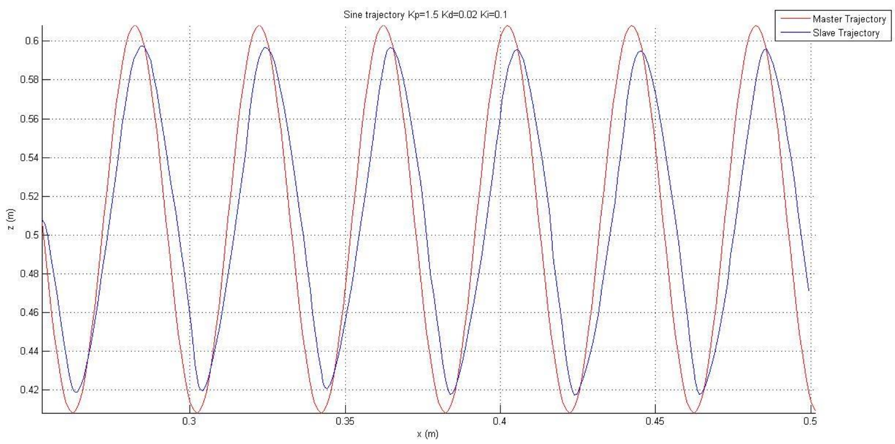

Figure 4.9: Sine trajectory $K_p = 1.5$ $K_i = 0.02$ and $K_d = 0.1$

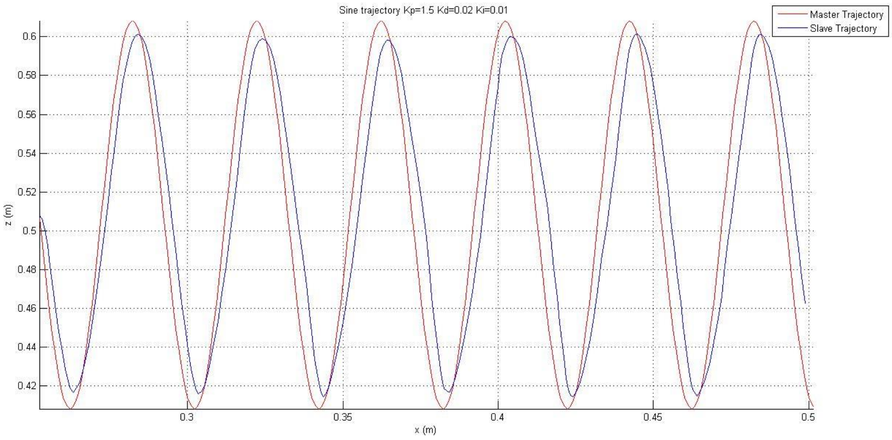

Figure 4.10: Sine trajectory $K_p = 1.5$ $K_i = 0.02$ and $K_d = 0.01$

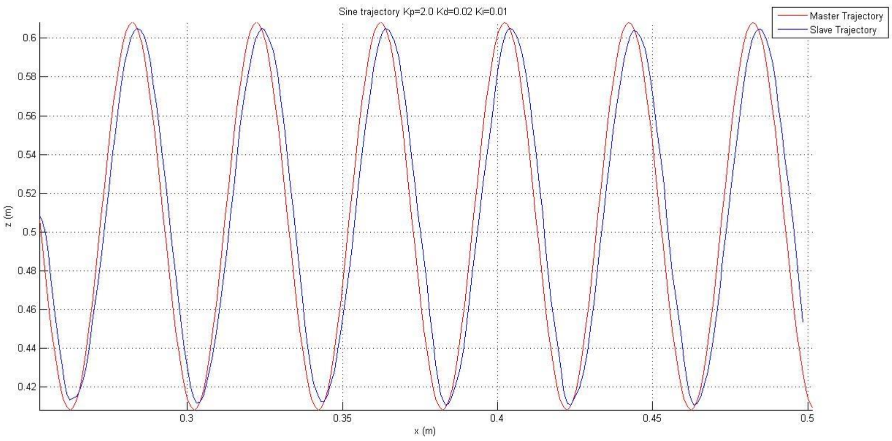

Figure 4.11: Sine trajectory $K_p = 2.0$ $K_i = 0.02$ and $K_d = 0.01$

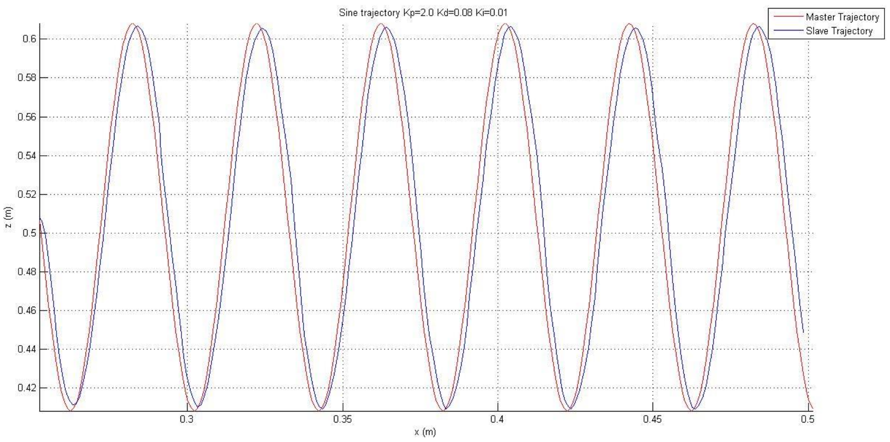

Figure 4.12: Sine trajectory $K_p = 2.0$ $K_i = 0.08$ and $K_d = 0.01$

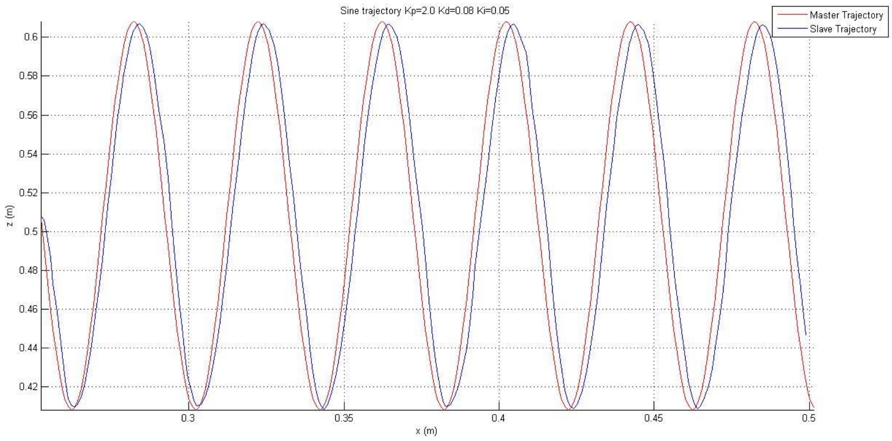

Figure 4.13: Sine trajectory $K_p = 2.0$ $K_i = 0.08$ and $K_d = 0.05$

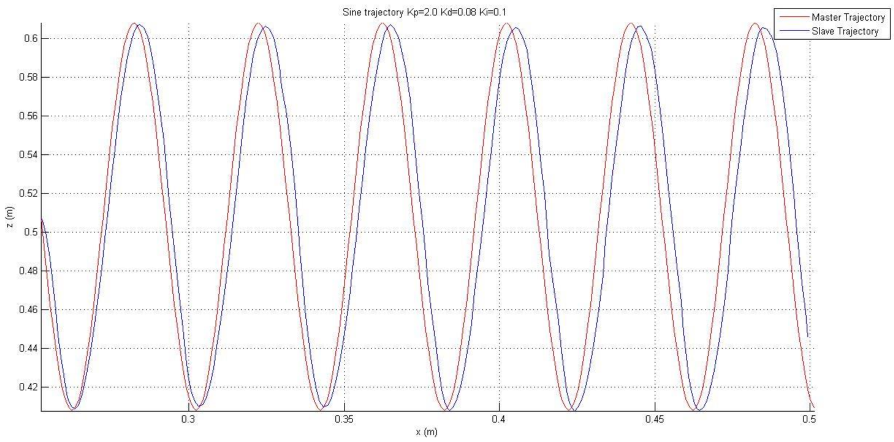

Figure 4.14: Sine trajectory $K_p = 2.0$ $K_i = 0.08$ and $K_d = 0.1$

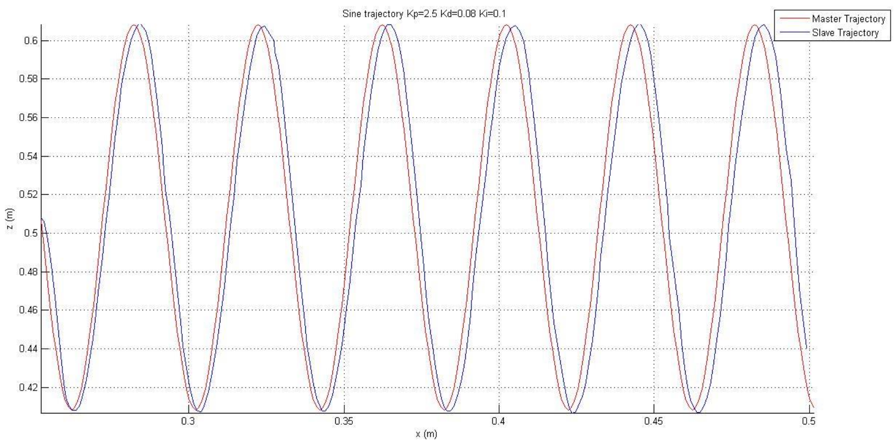

Figure 4.15: Sine trajectory $K_p = 2.5$ $K_i = 0.08$ and $K_d = 0.1$

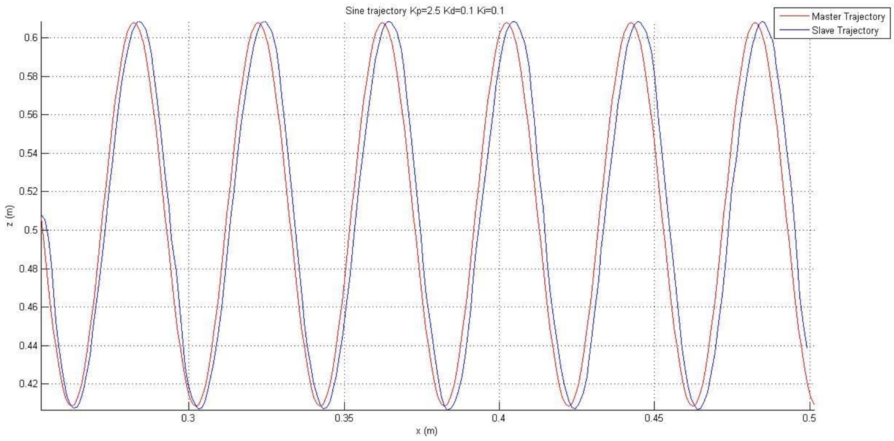

Figure 4.16: Sine trajectory $K_p = 2.5$ $K_i = 0.1$ and $K_d = 0.1$

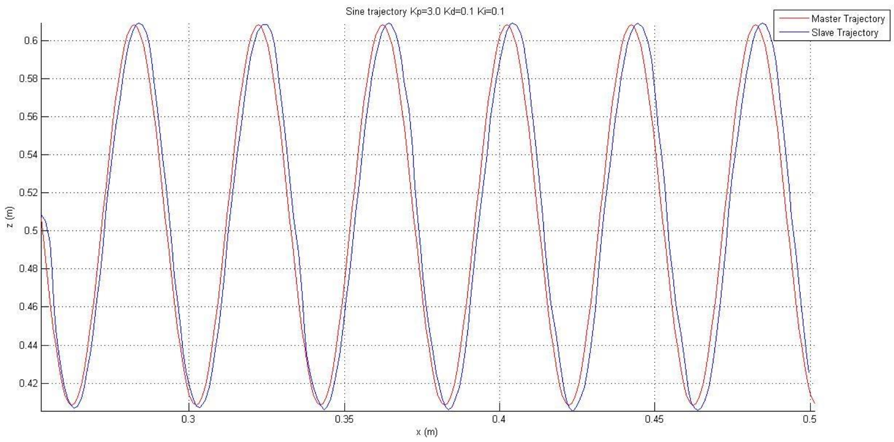

Figure 4.17: Sine trajectory $K_p = 3.0$ $K_i = 0.1$ and $K_d = 0.1$

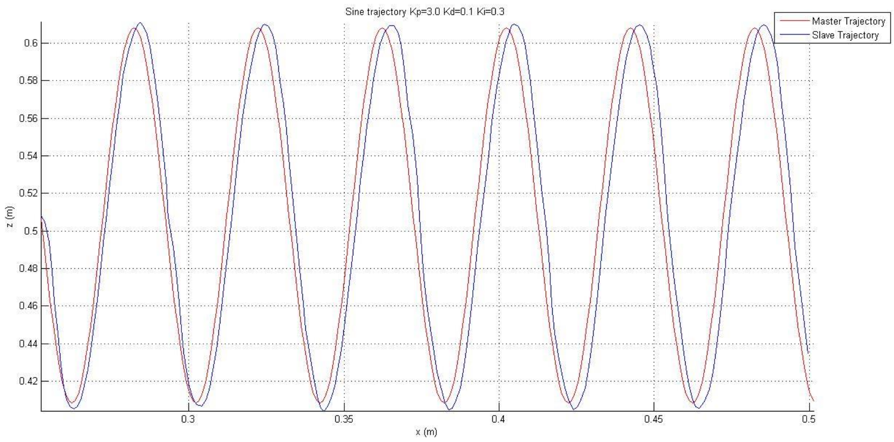

Figure 4.18: Sine trajectory $K_p = 3.0$ $K_i = 0.1$ and $K_d = 0.3$

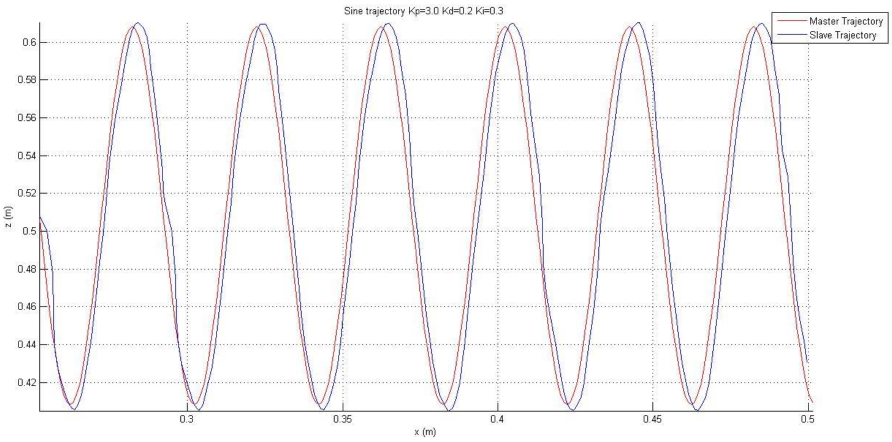

Figure 4.19 Sine trajectory $K_p = 3.0$ $K_i = 0.2$ and $K_d = 0.1$

Repeatability

Once established the best configuration for the PID parameters, a circular trajectory is generated on the master side and sent to the slave robot on the plane XZ. The test is made at different work frequencies (10Hz, 25Hz, 50Hz and 100Hz), and for each frequency the slave repeats the trajectory twenty times. The figures below represent the mean of each test (blue line) with the plots representing the error along the x and z axes.

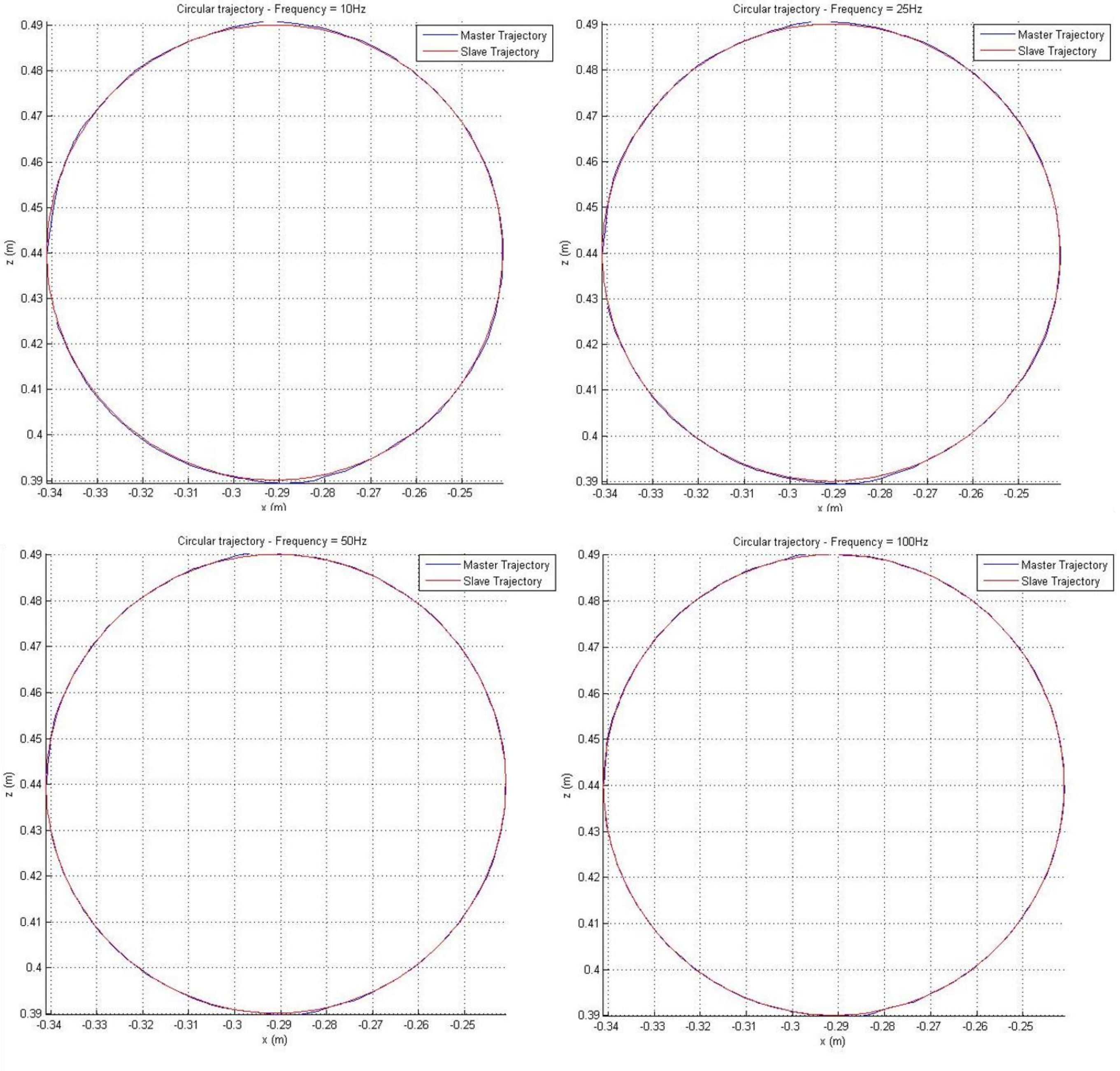

Figure 4.20: Circular trajectory at frequency 10Hz, 25Hz, 50Hz, 100Hz

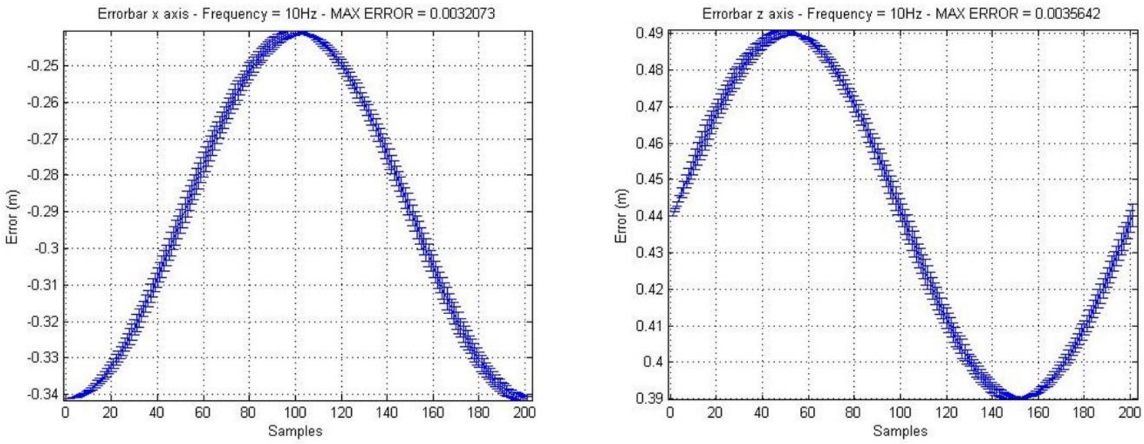

Figure 4.21: Errorbar plot at 10Hz on x and z axes

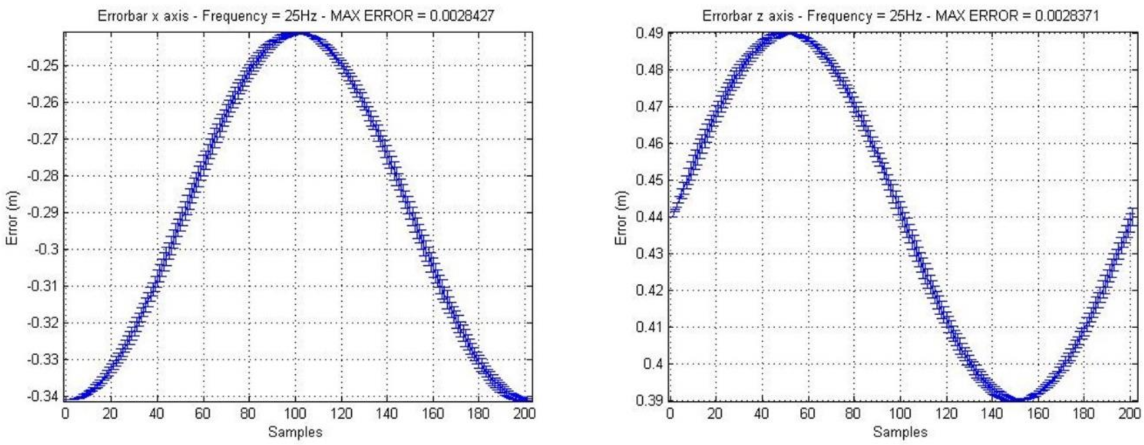

Figure 4.22: Errorbar plot at 25Hz on x and z axes

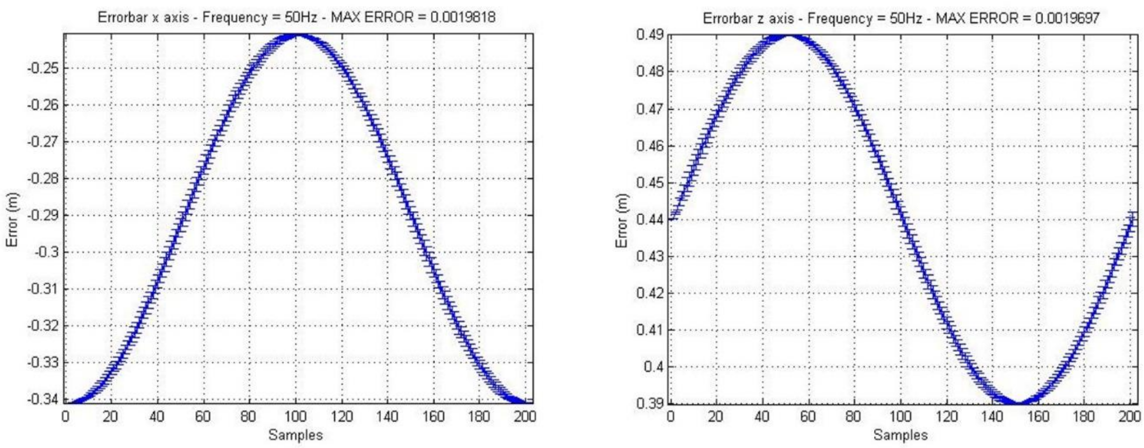

Figure 4.23: Errorbar plot at 50Hz on x and z axes

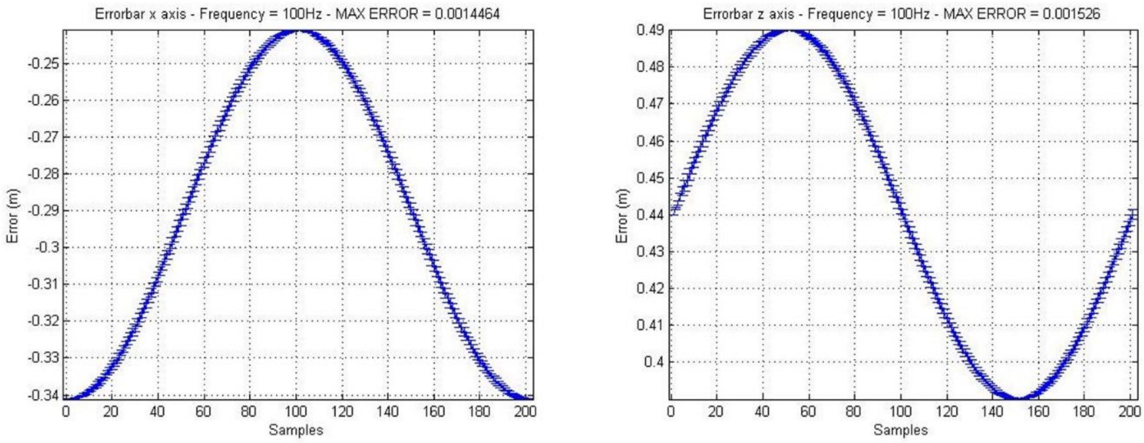

Figure 4.24: Errorbar plot at 100Hz on x and z axes

Has been observed that for frequencies greater than 100Hz, the maximum error value increases for both the axes. For that reason the work frequencies selected is 100Hz that represents the frequency at which the lowest value of the error is observed.

4.2.2 Simulation Task

One established the values of the PID control for the robot slave and the work frequency of the system, some simulation were conducted. The simulations concerned about a typical task in which the system will be called in action: the grasp and screwing of typical connector presents on the collimator system.

Information about position of the master device and slave robot are obtained, and in the figure below is represented different examples of trajectory imposed by the operator on the master device and how the robot slave track the movement imposed allowing to fulfil to the task required.

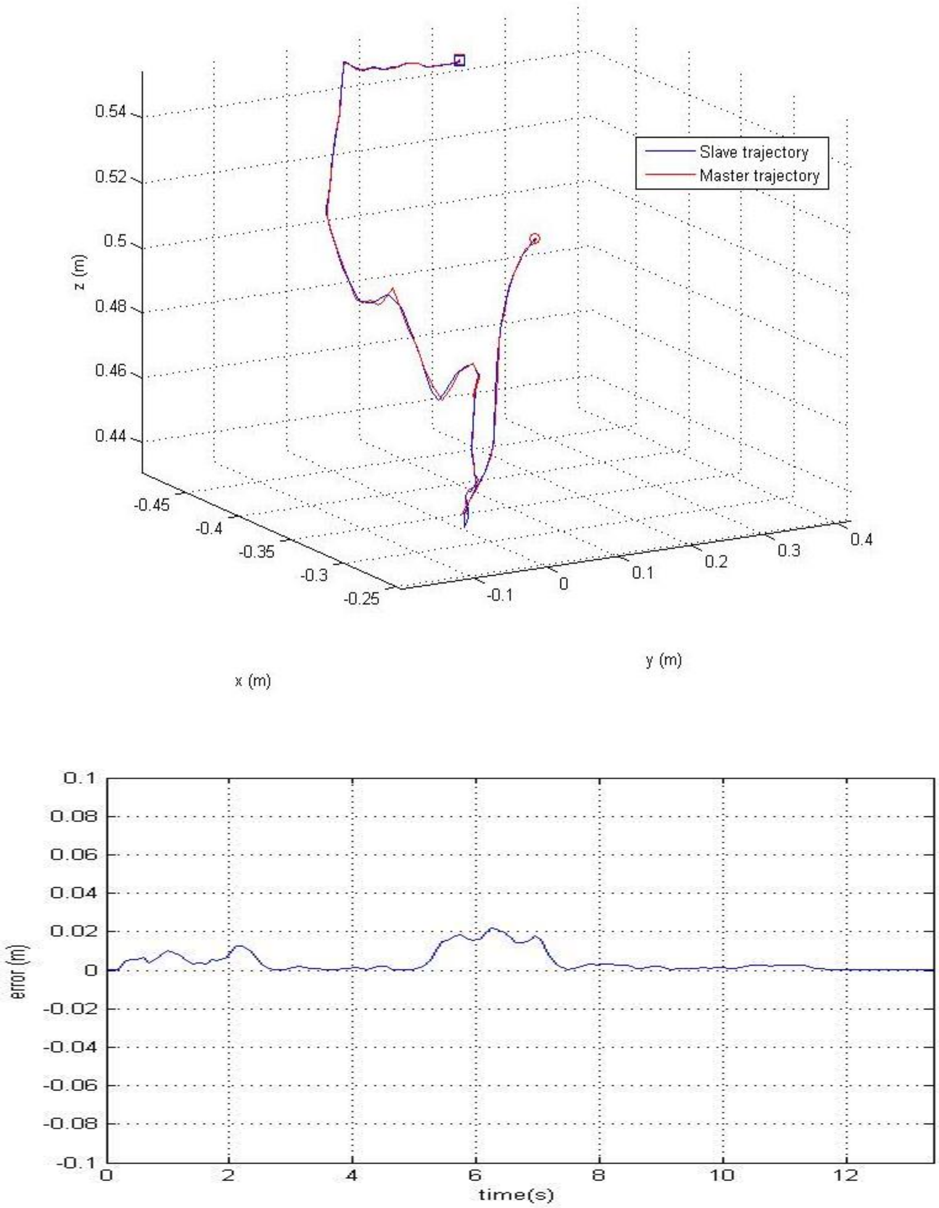

Figure 4.25: Trajectory tracking example 1 with error (m)

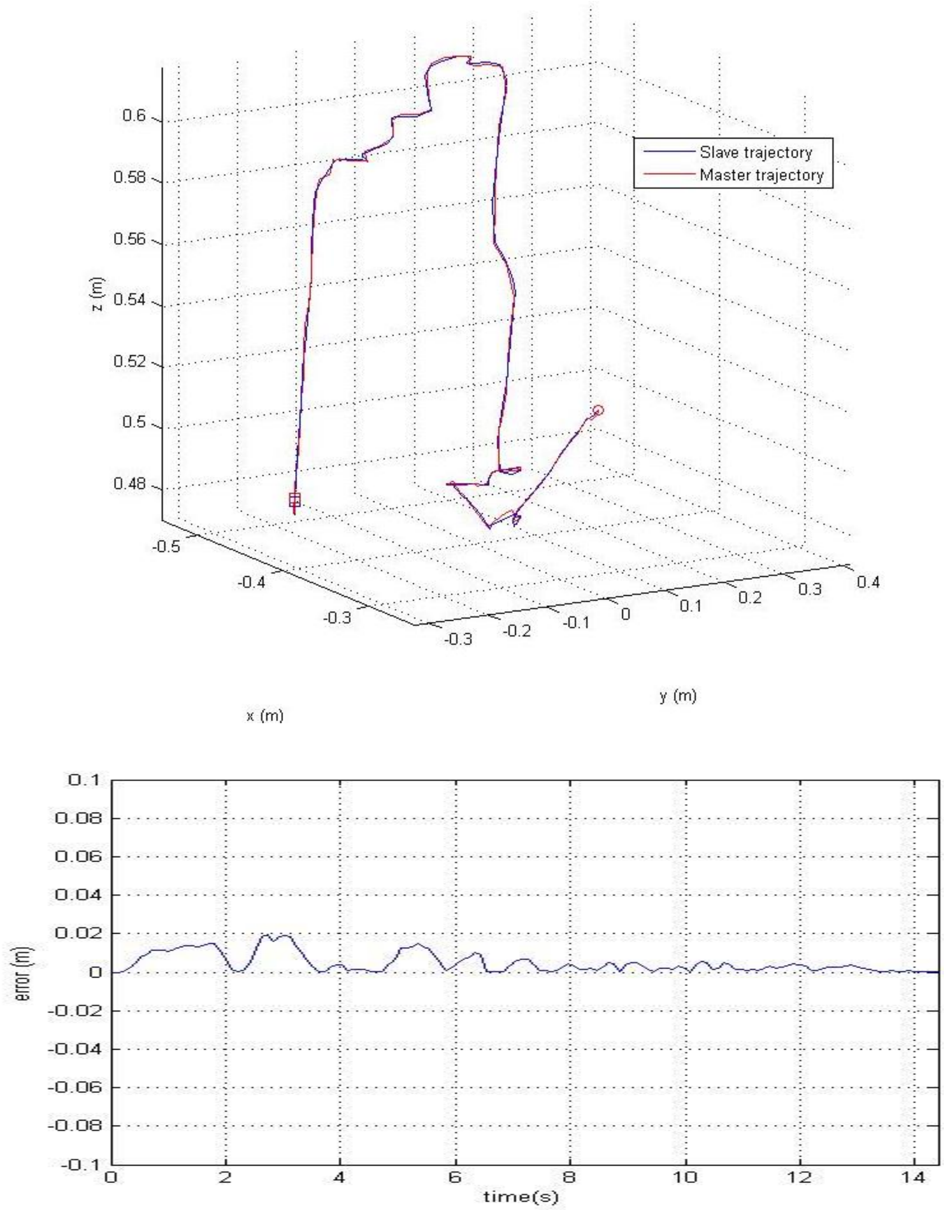

Figure 4.26: Trajectory tracking example 2 with error (m)

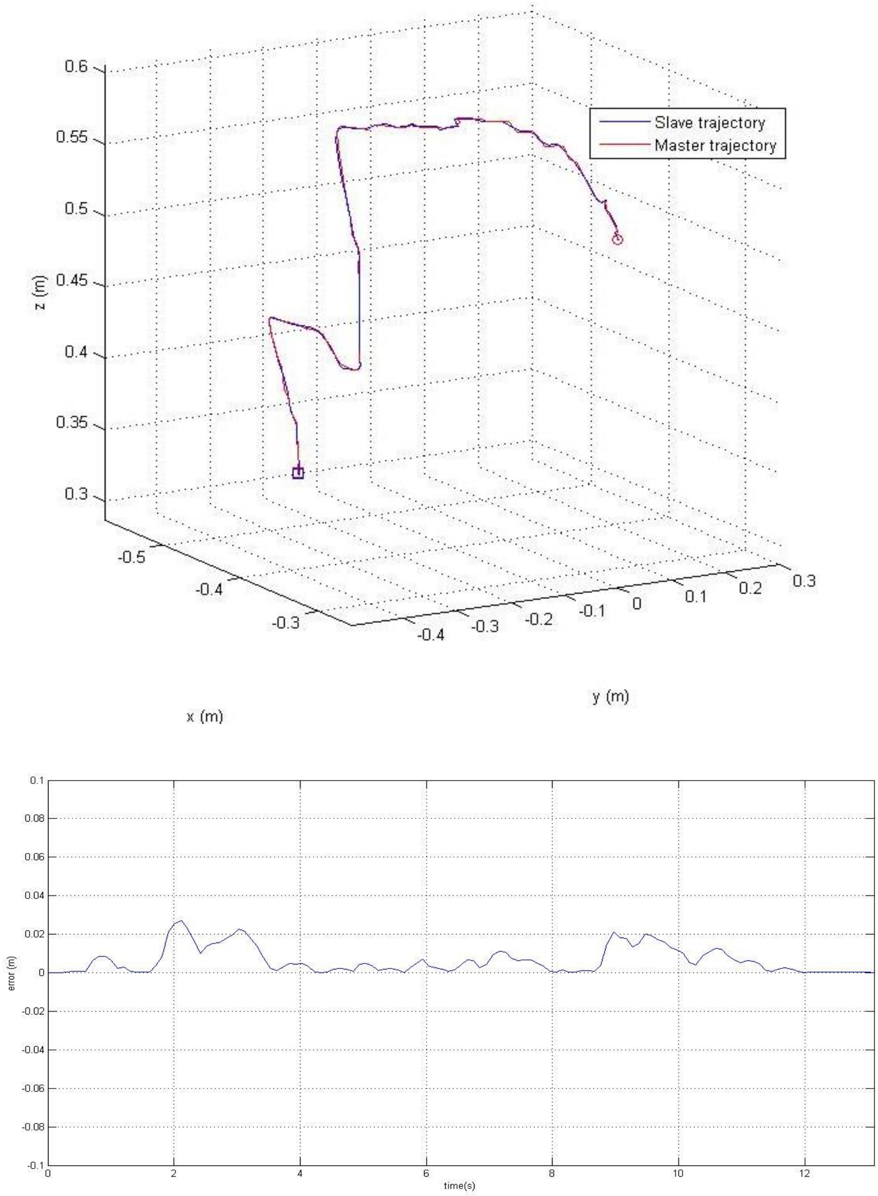

Figure 4.27: Trajectory tracking example 2 with error (m)

Important to take care, are the forces that the operator undergoes. As explained before, the force acts as a restoring force slowing the movement of the stylus, and indirectly of the user (figure 4.19), or taking back the position of the stylus in the point within the workspace of the slave robot (figure 4.18). Both behaves are described by the figures below.

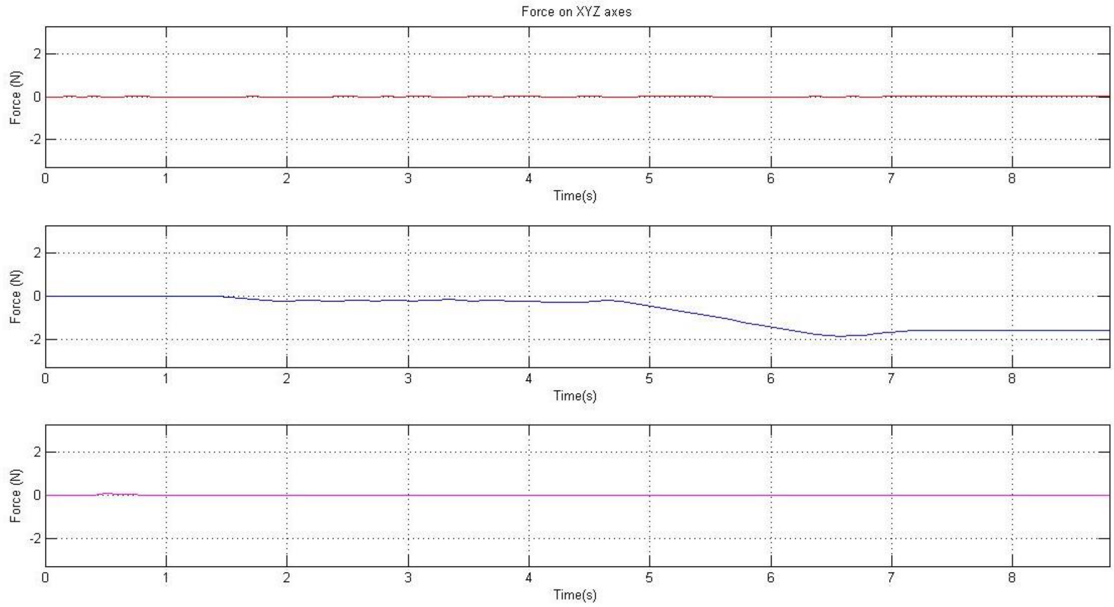

Figure 4.28: Force reaction when exceed the workspace

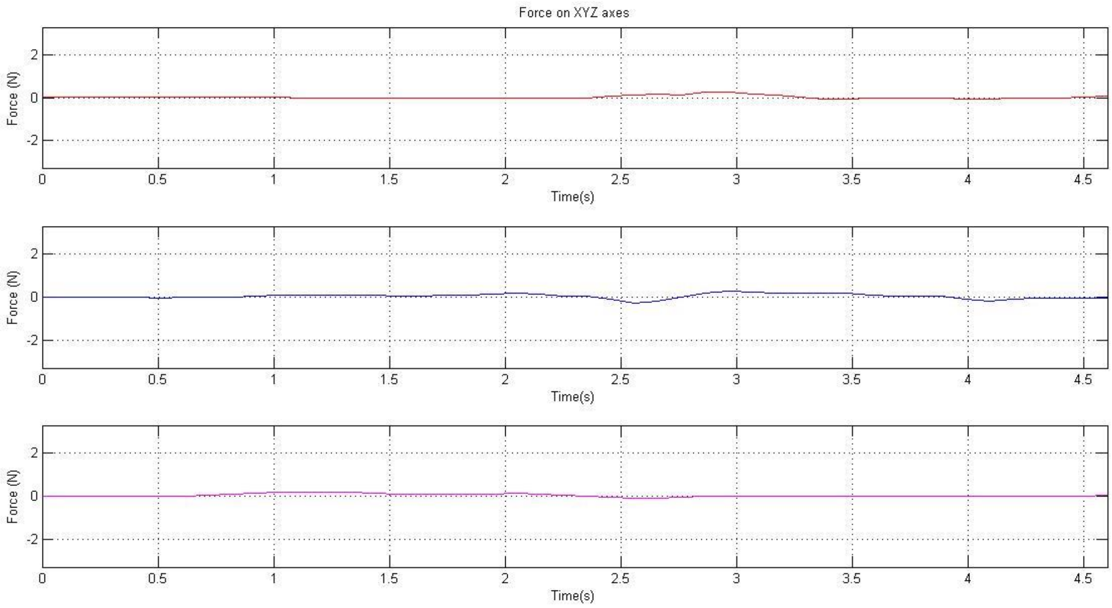

Figure 4.29: Force reaction slowing the movement of the stylus

Chapter 5

Conclusion

The use of a two-channel architecture for the bilateral control based on the error position has been used to develop a teleoperation control for telemanipulation in CERN accelerators harsh environment.

The robots on the master and slave side have been presented in detail with the control architecture and its software implementation.

Has been explained the process of tuning the PID parameters for the slave robot and fixed the work frequency.

Moreover, all the objectives have been completed:

- No problem of communication was found and use of packages allow managing information in a clear and simple way.
- The IK Algorithm has proved to work well in real time, even if the speed of calculation (~milliseconds) is not

compare with the IKFast [45] developed by OpenRAVE [46], whose speed is around two microseconds.

- The control system allows the user to control the slave robot joint-by-joint.
- The Graphical User Interface, controlled with the classic keyboard, lets to monitor all the system and stop it in dangerous case. Besides, the addition of the Schunk Powerball CAD Screen turned out to be useful to monitor continuously the configuration of the robot in the space.

In the end, it was also shown how all that allows to obtain an acceptable tracking of the master position for the slave robot, fulfilling the task of grasping and unscrewing request.

Bibliography

[1] "Introduction of CERN, LHC and the ATLAS Experiment." Available: http://zimmer.csufresno.edu/~yogao/ATLAS/CERN-LHC-ATLAS.html.

[2] "About CERN" , Available: http://home.web.cern.ch/about.

[3] "CERN," *Wikipedia, the free encyclopedia*.

[4] "Large Electron-Positron Collider," *Wikipedia, the free encyclopedia*.

[5] "The Accelerator Complex" Available: http://home.cern/about/accelerators

[6] "The Large Hadron Collider." Available: http://home.web.cern.ch/topics/large-hadron-collider

[7] "CERN LHC Guide" Available: http://cds.cern.ch/record/1092437/files/CERN-Brochure-2008-001-Eng.pdf

[8] "ALICE" Available: http://home.cern/about/experiments/alice

[9] "ATLAS" Available: http://home.cern/about/experiments/atlas

[10] "LHCb" Available: http://home.cern/about/experiments/lhcb

[11] "CMS" Available: http://home.cern/about/experiments/cms

[12] "Experiments" Available: http://home.cern/about/experiments

[13] "The collimation system: defence against beam loss - CERN Courier." Available: http://cerncourier.com/cws/article/cern/54384

[14] T. Weiler, O. Aberle, R. Assmann, R. Chamizo, Y. Kadi, J. Lettry, R. Losito, and S. Redaelli, “LHC collimation system hardware commissioning,” in IEEE Particle Accelerator Conference, 2007. PAC, 2007, pp. 1625 - 1627.

[15] “Radiation at CERN” Available: http://environmental-impact.web.cern.ch/environmental-impact/en/Radiation/Radiation-en.html

[16] Keith Kershaw, Bruno Feral, Jean-Louis Grenard, Thierry Feniet, Sven De Man, Cathelijne Hazelaar-Bal, Caterina Bertone and Ruehl Ingo. “Remote Inspection, Measurement and Handling for Maintenance and Operation at CERN” . Int J Adv Robot Syst, 2013, 10:382. doi: 10.5772/56849

[17] Horne R.A, “Teleoperator evolution at CERN” . International Symposium on Teleoperation and Control, Bristol, UK. CERN SPS/88-32 (AMR). 8p

[18] R.C.Goertz, “Fundamentals of general purpose remote manipulation” , Nucleonics, 1952

[19] R.C.Goertz, “Mechanical master-slave manipulator, Nucleonics” , 1954

[20] A.K. Bejczy, “Towards advanced teleoperation is space” , Teleoperation and Robotics is Space, American Institute of Aeronautics and Astronautics, Reston 1994

[21] Hirzinger, Gerd, et al., "Sensor-based space robotics-ROTEX and its telerobotic features." Robotics and Automation, IEEE Transactions on 9.5 (1993): 649-663.

[22] Gettman, Matthew T., et al. "Robotic–assisted laparoscopic partial nephrectomy: technique and initial clinical experience with DaVinci robotic system." Urology 64.5 (2004): 914–918.

[23] Gidaro, S., et al. "Telelap Alf–X: a novel telesurgical system for the 21st century." Surgical technology international 22 (2012): 20–25.]

[24] “Haptic Technology”
Available: https://en.wikipedia.org/wiki/Haptic_technology

[25] Brooks, T.L. and A.K. Bejczy, 1985, "Hand Controllers for Teleoperation," Technical Report JPL Publication 85–11, Jet Propulsion Laboratory, Pasadena, CA.

[26] Durlach, N.I. et al, 1992, "Virtual Environment Technology for Training," Virtual Environment and Teleoperator Research Consortium (VERTREC), MIT, BBN Report 7661.

[27] G. Raju, G. Verghese, and T. Sheridan. “Design issues in 2–port network models of bilateral remote manipulation” . In: International Conference on Robotics and Automation. IEEE Comput. Soc. Press, 1989, pp. 13161321

[28] D. A. Lawrence. “Stability and transparency in bilateral teleoperation” . In: Robotics and Automation, IEEE Transactions on 9.5 (Oct. 1993), pp. 624637.

[29] G.Niemeyer, J.E.Slotine, “Stable adaptive teleoperation” , International Journal of Oceanic Engineering 16(1)(1991)152-162

[30] W.S.Kim, “Developments of new force reflecting control schemes and an application to a teleoperation training simulator” , IEEE International Conference on Robotics and Automation, ICRA’ 92,Nice,France,May1992,pp.1412-1419.

[31] W.R.Ferrell, “Delayed force feedback”, IEEE Transactions on Human Factors in Electronics, 8(1966)449-455.

[32] P. Arcara and C. Melchiorri , "Control schemes for teleoperationwith time delay: A comparative study" , Robot Autonomous Syst. , vol. 38 , no. 1 , pp.49 −64 , 2002

[32] “Schunk Powerball LWA4.6” , Available: http://mobile.schunk-microsite.com/fileadmin/user_upload/broshures/SCHUNK_Technical_data_LWA4P.pdf

[33] “Schunk PG-plus 70” , Available: http://mobile.schunk-microsite.com/fileadmin/user_upload/pdf/SCHUNK_technical_data_PG-plus-70_01.pdf

[34] Siciliano, Bruno, and Oussama Khatib, eds. Springer handbook of robotics. Springer Science & Business Media, 2008.

[35] Niku, Saeed B. Introduction to robotics: analysis, systems, applications. Vol. 7. New Jersey: Prentice Hall, 2001.

[36] Aggiungere il richiamo alla bibliografia per il trattamento complete delle singolarita

[37] Peiper, Donald L. The kinematics of manipulators under computer control. No. CS-116. Stanford Univ Ca Dept Of Computer Science, 1968.

[38] Brandstotter, M., Angerer, A., Hofbaur, M.: Analytical solution of the inverse kinematics problem of industrial serial manipulators with an ortho-parallel basis and a spherical wrist. Proceedings of the Austrian Robotics Workshop 2014 pp. 7 - 11, 2014.

[39] J.Angeles, ” Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms,” Mechanical Engineering Series. Springer,

2007.

[40] “Geomagic® Touch™ (PHANToM Omni)” ,

Available: http://www.geomagic.com/download_file/view/3685/7839/

[41] A. J. Silva, O. A. D. Ramirez, V. P. Vega, and J. P. O. Oliver, “Phantom Omni haptic device: Kinematic and manipulability” , in Electronics, Robotics and Automotive Mechanics Conference, 2009. CERMA'09., pp. 193_198, IEEE, 2009.

[42] “Geomagic® Touch™ (PHANToM Omni) Specs”

Available: http://www.geomagic.com/en/products/phantom-omni/speci_cations/

[43] “OpenHaptic Toolkits” Available:

http://www.geomagic.com/files/9913/4858/2409/OpenHaptics_datasheet_hi.pdf

[44] Tim Wescott (October 2000). "PID without a PhD" (PDF). EE Times-India.

[45] “IKFast” , Available:

http://openrave.org/docs/0.8.2/openravepy/ikfast/

[46] “OpenRave” , Available; http://openrave.org/