Polytechnic of Turin

Master's degree: Mechatronic Engineering
A.y. 2020/2021
Degree session: 12/2021

Development of a robot control system for safe and natural Human-Robot interaction

Supervisors:
Prof. Alessandro Rizzo
Prof. Guillem Alenya
Prof. Alberto Olivares Alarcos

Candidate:
Andrea Abbate

Abstract

For many people with disabilities, simple activities such as drinking, opening a door, or pushing an elevator button require the assistance of an external system, which reduces the independence of the individual.

Assistive robotic systems could enable these people to perform this kind of tasks autonomously again and thereby increase their independence and quality of life.

The problem of adopting this kind of control system is that sometimes are non-controlled in safety. There exist a lot of normative that limits the human-robot interaction forces.

This project aims at developing a novel hybrid control interface in order to control position and impedance of a robotic manipulator, with respect also to the generated forces.

This thesis presents a Cartesian position control system for KINOVA Gen3 robotic arm, which performs a proportional-derivative control law based to the Jacobian transpose method, that does not require inverse kinematics.

A second control is proposed to change the robot's rigidity in real-time based on Adaptive admittance control system. This control allows the user to modulate the robot's impedance parameters before to perform a task.

The results demonstrate that combining the two methods presented above, the user can control robot positions with a simple software application connected to the Kinova Kortex API, adapting the robot's impedance depending on its parameters.

In the future work are reported other two methods that could optimize the performance of this control system.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

COBOT	Collaborative Robot
DoF	Degree of Freedom
API	Application Programming Interface
IRI	Institut de Robotica i Informatica Industrial
TCP	Tool Centre Point
PTP	Point to Point
IC	Impedance Controller
AC	Admittance Controller
IMF	Impact Force
CSF	Clamping/Squeezing Force
PSP	Pressure/Surface Pressing
CA	Contact Area
CC	Compression Constant

Symbols

q	Robot's joints Position
$\dot{q}$	Robot's joints Velocity
$\ddot{q}$	Robot's joints Acceleration
τ	Robot's joints Torque
M	Inertia Matrix
B	Damping Matrix
K	Stiffness Matrix
K_P	Matrix of Proportional Gain
K_D	Matrix of Derivative Gain
e	End-Effector Position and Orientation Vector
$\dot{e}$	End-Effector Linear and Angular Velocity Vector
i_χ	Element i (vector or matrix) in Cartesian Space
F	Force Vector

1. *Introduction*

This master's thesis aims to implement a control system for the collaborative robot Kinova Gen 3, which is able to ensure greater safety during human-machine operations.

In this chapter are described the motivations behind the start of this research path and then the objectives that have been achieved.

Finally, the contents of the single chapters will be briefly described in order to anticipate the structure of this work.

1.1. Motivations

Robotic systems could potentially be very useful in helping people during their daily lives as well as at work, by collaboratively or autonomously performing simple required tasks, such as drinking, opening doors, or pushing elevator buttons:

- Daily life: drinking, opening a door or pushing a lift button.

- Working environment: picking up and processing heavy materials, assisting in mass production, or doing tasks that do not bring value during a human operator's working day.

A robotic system could enable people with severe motor disabilities to achieve greater independence and thus increase their quality of life.

While on an assembly line with high production rates, a robotic arm system could be more productive in terms of quantity.

An engineering challenge has long been to establish an increasingly safe and effective man-machine collaboration in order to free humans from the heavy and exhausting works involved in quantity production, while allowing them to do the same type of work with a greater focus on product quality.

Therefore, the main motivation of this project is to define a control system for an assistance and collaborative robot that can guarantee safety during work processes or assistance for people with physical disabilities.

Furthermore, to apply the system in any situation without danger, it must give the user the ability to decide the behaviour of the robot when there are physical interactions with the environment, through a compliance control system.

1.2. Objectives

In this project the propose is to develop a novel approach that combines impedance/admittance control system with a force controlled one.

This multi-branch control system should be able to make the Kinova compliant with external applied forces that modify its trajectory to reach a desired position.

The specific objectives of this thesis are the following:

- Design of position and velocity controller in Cartesian Space.
- Implementation of an admittance control to adapt the robot's rigidity for different situations.
- Analysis of the robot behaviour with and without external applied forces.
- Analysis of the robot behaviour in home position and in movement to the desired position.
- Optimization of controller parameter in order to ensure the TS 15066 normative.
- Theoretical design of a null-space control system

1.3. Thesis outline

This document is divided in the following sections:

- Chapter 2: State of The Art

  This chapter explores the literature concerning the most common techniques used in the in the implementation of assistive manipulators. It covers examples of robotic arms, adaptive impedance control and body-machine interfaces. It further aims to describe the safety regulations that need to be adopted if a decision is made to assist or collaborate with humans by means of a mechanical arm, justifying why by presenting some examples of human-machine collaboration that have led to serious accidents

- Chapter 3: Experimental Set-Up

  The aim of this chapter is to describe the tools that were used to structure the final paper. Both the hardware tools (Kinova Gen3 Ultra Lightweight) and the software tools (Kinova Kortex Web App and Kinova Kortex API) will be described.

- Chapter 4: Theoretical background
  This section aims to explain as clearly as possible the theoretical background needed to develop the control system in question. The definition of the concept of Impedance, Admittance, the related control systems and their differences will be given. Since the aim of the paper is to implement a specific trajectory, the concept of developing a point-to-point trajectory is also explained.
- Chapter 5: Software Implementation
  This chapter first explains how the software production process takes place. Secondly, the operation of the two control systems implemented was described: Control of a Joint with Adaptive Admittance and Cartesian Position Control. Finally, since the code was written entirely in C++, the main functions useful for the correct operation of the software were described using flowcharts.
- Chapter 6: Analysis and Results
  As in any experimental thesis, the analysis and results chapter aims to describe the experiments that were carried out to demonstrate the results obtained. The relative results of two tests have been described and reported, the first will be called static (robot stationary in the initial position) the second dynamic (robot moving towards a final position).

After analysing the different behaviours, conclusions were drawn to define the best pair of values for the control system parameters.

- Chapter 7: Conclusions
  The last chapter is intended to be a summary of what has been done during this six-month period in Barcelona, reporting what could be possible future studies to increase the efficiency of the control system so far developed and implemented.

2. State of the art

The aim of this chapter is to give an introduction of what are collaborative and assistive robots where are applied and how are normalized in order to be safe for human.

Basically, it presents a definition of COBOT also explaining its different types that are implemented in industry.

Then will be given a brief description of what these robots are able to do, focusing on the assistive tasks studied by the "Institut de Robotica I Informatica Industrial" (IRI) in Barcelona.

2.1. Robotics arm

A collaborative robot (Figure1) is a machine able to work and collaborate within human during his working operation or during the everyday life tasks.

The first collaborative robot was built in 1996 by J. Edward Colgate and Michael Peshkin professors at the Northwestern University. These machines were developed without any motor, and through defined control panels the human operator was able to control its movements.

Figure 1 *ABB Collaborative Robot*

In 2002 for the first time, it was defined a safety standard ISO, updated then in 2016, the ISO/TS 15066:2016: "Robots and robotic devices Collaborative robots":

"*This Technical Specification specifies safety requirements for collaborative industrial robot systems and the work environment, and supplements the requirements and guidance on collaborative industrial robot operation given in ISO 10218-1 and ISO 10218-2.*

This Technical Specification applies to industrial robot systems as described in ISO 10218-1 and ISO 10218-2.

It does not apply to non-industrial robots, although the safety principles presented can be useful to other areas of robotics.

NOTE This Technical Specification does not apply to collaborative applications designed prior to its publication." (ISO/TS 15066:2016 - Robots and robotic devices - Collaborative robots)

The ISO/TS 15066 standard defines four collaborative robot typologies, classified by their work field.

In particular are defined as:

- *Safety Monitored Stop*: the work of the robot is carried out most of the time by itself, occasionally the human operator collaborates with it. The working area is defined by a perimeter that ensures safety since, if it is crossed, the brakes of the robot are activated, and the activity will be stopped.
- *Hand Guiding*: the robot is guided manually by the operator who defines the path. In order to make the operation more collaborative and safer, sensors are attached to the robot.
- *Speed and separation monitoring*: the working area is monitored by lasers or vision systems, which will slow down or stop the robot if the operator is too close.
- *Power and Force Limiting*: type of robot that can sense levels of force along its path. If its load is excessive it will stop, and it is programmed to dissipate forces if it hits a large surface. With these collaborative robots it is possible to perform self-learning, moving the arm manually and teaching it the movement to repeat.

2.2. Example of collaboration/assistance human-robot

As it is possible to observe in Figure 2 the collaborative robots are classified from the bigger to the smallest, their dimensions depend on the kind of work that have to do.

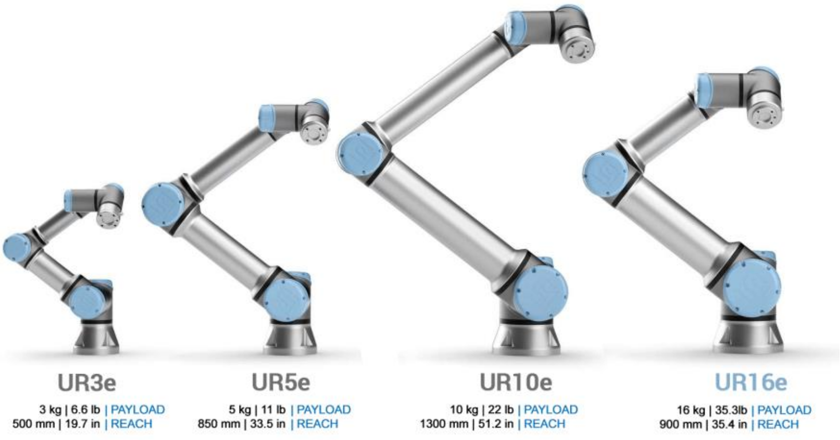

Figure 2 *Universal Robot Family*

The biggest ones are mainly used within industries that need to process heavy materials, such as automotive body parts, while smaller robots can be used both within the working world to collaborate with human operators and to assist humans in everyday tasks.

With respect to the human-robot collaboration it is possible to define the following tasks:

- *Pick and Place*: taking a product from one environment and releasing it in a different place.

- *Machine tool servicing*: working with machine tools to make their work easier and cleaner.

- *Packaging and palletising*: optimisation of storage time and operations.

- *Quality control*: through appropriate sensors the robot can control the quality of the finished product.

- *Assembly*: if appropriately programmed, the cobot can carry out assembly operations, such as screw tightening.

- *Polishing*: through a specific accessory, the collaborative robot can carry out perfect polishing operations.

On the other hand, with respect to the assistance operation it is possible to take the example of the industrial institute of robotics and informatics in Barcelona (IRI – Institut de Robotica i Informatica Industrial) where into the department of perception and manipulation the main research area is about develop control systems that allow the robot to perform assistance tasks for humans, such as:

- Learning by demonstration: Development methods to perform tasks at different levels of abstraction through object-action relationships. Object models are generated from visual and depth information, and manipulation actions are learned from human demonstrations using multimodal algorithms that combine vision and haptics.

***Figure 3** Learning by Demonstration (IRI)*

- Planning for perception and manipulation: View planning for object modelling and manipulation planning. High-level task formulations are supplemented with methods based on low-level geometry and simplified physical models. In this way, specific sequences of movement commands can be obtained.

Figure 4 Planning for Perception and Manipulation (IRI)

- Perception of rigid and non-rigid objects: Study of computer vision algorithms for interpreting and understanding scenes from images, with applications in robotics and medical imaging. In particular, the main activities concern the recovery of rigid and non-rigid shape, movement and camera pose from single images and video sequences.

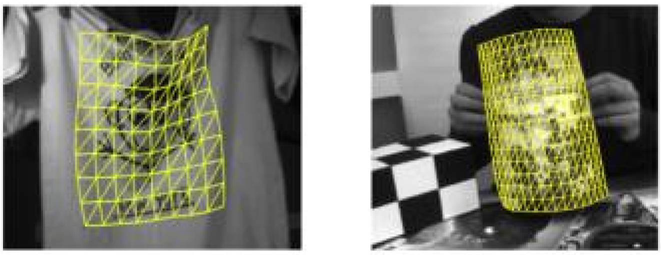

Figure 5 Perception of rigid and non-rigid objects

2.3. Robot and work accident

From the paper "Rivista Ambiente e Lavoro" it is possible to read an article "Robot e incidenti sul lavoro" written in 2017 by Renata Borgato, trainer who collaborates with the Faculty of Psychology, University of Milan Bicocca.

In this short essay Renata Borgato focuses her attention on the implementation of industrial robots into an industry, and in particular she reports some accidental events into this human-machine environment that are due not to the malfunctioning of the robot, but to human error.

The Times of India reported a fatal accident occurred at SKH Metals, in the Uttar Pradesh region, where a worker was impaled by one of the arms of a welding robot as he manually intervened in its field of action to reposition a metal sheet that had become misaligned.

She reported another injury occurred at Volkswagen in Baunatal Germany, that involved an operator who was slammed against a metal plate by a robot he was setting up. Again, the man was working inside the safety cage of the robot.

From these facts it is possible to ensure that technical prevention measures must be strengthened, but also that they are not sufficient and that the human factor must be addressed through training and education.

2.4. Injury Severity Criteria

The Institute for Occupational Safety and Health of the German Social Accident Insurance (IFA – formally BGIA) developed the TS 15066 normative, which gives the injury severity criteria.

These criteria establish the maximum force that can be applied against each part of human body in order to avoid any possible non-safety contact.

In the following table it is possible to observe the descripted criteria:

Table 1 *TS 15066 normativ*

Body model Main and individual regions with codification [a]			Maximum allowable Limit values of the injury severity criteria (CSF, IMF, PSP) and arranging factor (CC) [b]			
Main body regions		**Individual body regions**	**CSF [N]**	**IMF [N]**	**PSP [N/cm²]**	**CC [N/mm]**
1. Head with neck	1.1	Skull/Forehead	130	175	30	150
	1.2	Face	65	90	20	75
	1.3	Neck (sides/neck)	145	190	50	50
	1.4	Neck (front/larynx)	35	35	10	10
2. Trunk	2.1	Back/Shoulders	210	250	70	35
	2.2	Chest	140	210	45	25
	2.3	Belly	110	160	35	10
	2.4	Pelvis	180	250	75	25
	2.5	Buttocks	210	250	80	15
3. Upper extremities	3.1	Upper arm/Elbow joint	150	190	50	30
	3.2	Lower arm/Hand joint	160	220	50	40
	3.3	Hand/Finger	135	180	60	75
4. Lower extremities	4.1	Thigh/Knee	220	250	80	50
	4.2	Lower leg	140	170	45	60
	4.3	Feet/Toes/Joint	125	160	45	75
[a] BR - Body region with codification Regions - Name of the individual body region			[b] CSF - Clamping/Squeezing force IMF - Impact force PSP - Pressure/Surface pressing CC - Compression constant			

These limits and properties are defined as follows:

- Impact Force (IMF)

  "*The maximum permissible force acting on a body region resulting from a robot collision where the period of contact results in an elastic deformation of the soft tissue.*" An impact force occurs when the difference between the maximum force and other forces before and after the maximum is more than 5 N over a time interval of 0.5 s or less.

- Clamping/Squeezing Force (CSF)

  "*The maximum permissible force acting on a body region resulting from a robot collision where the period of contact results in a plastic deformation of the soft tissue.*" This kind of force can be detected by a spread of the force signal of not more than 5 N over a time interval of more than 0.5 s.

- Pressure/Surface Pressing (PSP)

  "*The maximum permissible partial pressure load in the case of both IMF and CSF where the contact area (CA) of the collision is small as to reduce the defined IMF and CSF limits.*" The critical contact area is defined as:

$$Elastic\ Deformation = CA = \frac{IMF}{PSP} \quad (2.1)$$

$$Plastic\ Deformation = CA = \frac{CSF}{PSP} \quad (2.2)$$

- Compression Constant (CC)

  "*The deformation constant of a body region through which the maximum compression path is established assuming linear deformation behaviour throughout the soft tissue body region.*"

In figure 6 is illustrated the factors that analyse impact between a robot part and a human arm. From the red dotted lines, the 4 safety zones are reported. From the right it is possible to observe that the first level is the human-robot distance that should be guaranteed, once this level is exceeded the second step is the impact detection by human, from this point start the body penetration until the maximum body compression. The fourth level defines the limits defined above.

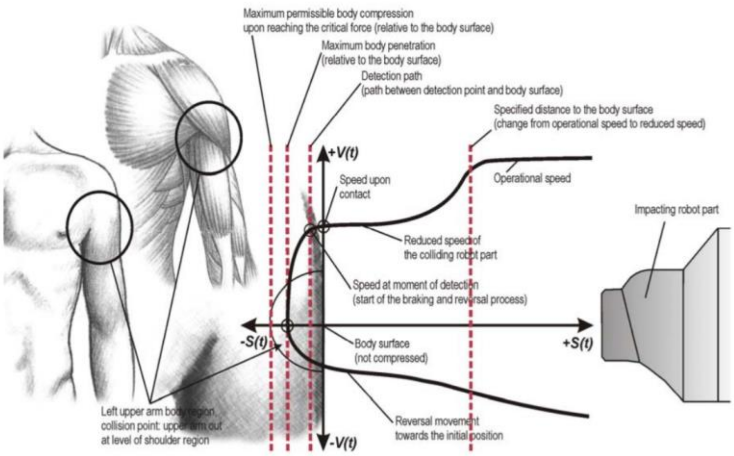

***Figure 6** Impact between a robot part and a human arm*

3. *Experimental Set-Up*

The Institute of Robotics and Industrial Informatics (IRI) in Barcelona is divided into several departments, each dealing with a different aspect of robotics research.

The perception and manipulation department offers the possibility of working with different types of robots.

In particular, an anthropomorphic robot from the manufacturer Kinova, called “Kinova Gen3 Ultra Lightweight”, was used for this project.

In this section, the hardware and software used to develop the control system will be described from a technical point of view.

3.1. Hardware: Kinova Gen3 Ultra Lightweight

The robotic KINOVA Gen3 Ultra lightweight (Figure 8) is a robotic arm with 7 degrees of freedom (DoF), composed only by spherical joints able to make the robot more versatile and to give the possibility to reach a high quantity of configuration.

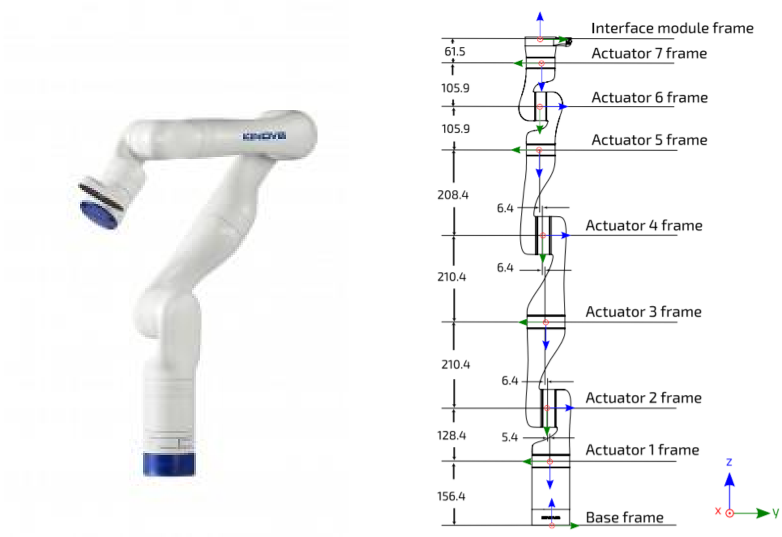

***Figure** 7 Kinova Gen3 Ultra Lightweight*

In table 2 it is possible to observe the main physical characteristics, such as maximum Payload, reachability, and velocity.

Table 2 *Physical Characteristics*

Parameter		Value(s)
Mass (without gripper and with vision module)	7 DoF	8.2 kg
	6 DoF	7.2 kg
Payload	7 DoF	mid-range continuous; no gripper: 4.0 kg
		full-range continuous; no gripper: 2.0 kg
	6 DoF	mid-range continuous; no gripper: 4.0 kg
		full-range continuous; no gripper: 2.0 kg
Maximum reach (fully extended)	7 DoF	902 mm
	6 DoF	891 mm
Maximum Cartesian translation speed		50 cm/s
Degrees of freedom		7 DoF, 6 DoF
Actuators		small: qty 3 (small)
		large: qty 4 (7 DoF), 3 (6 DoF)
Wrist interaction buttons		qty 2 (user-configurable*; default for admittance control; teaching mode controls)
Power supply voltage		24 VDC (nominal, 20 to 30 V)
Materials		Carbon fiber shell
		Aluminum
Communciations and control		100 Mbps Ethernet for real-time 1 kHz control
		100 Mbps Ethernet for Vision module / expansion

In table 3 are reported the sensors that are connected into the body of the robot.

Table 3 *Connected Sensors*

Feature	Value(s)
Sensors	current sensors (motor), temperatures (motor), voltage, torque, position

One of important feature of this hardware is the presence of a camera able to sense depth and colour following the follow feature (table 4):

Table 4 *Camera Characteristics*

Feature	Detail
Depth sensor	480 x 270 (16:9) @ 30, 15, 6 fps 424 x 240 (16:9) @ 30, 15, 6 fps FOV: 72 ± 3° (diagonal) minimum depth distance - 18 cm
Color sensor	1920 x 1080 (16:9) @ 30, 15 fps; FOV 47 ± 3° (diagonal) 1280 x 720 (16:9) @ 30, 15 fps ; FOV 60 ± 3° (diagonal) 640 x 480 (4:3) @ 30, 15 fps; FOV 65 ± 3° (diagonal) 320 x 240 (4:3) @ 30, 15 fps; FOV 65 ± 3° (diagonal) focusing range - 30 cm to ∞

3.1.1. WorkSpace

The effective workspace is the reachable region by the robot end effector.

There are two definitions of effective workspace:

1. Nominal (or reachable) workspace - the set of all three-dimensional position that are reachable by the end-effector through at least one of its position and orientation combination (figure 9).
2. Dextrous workspace - the subset of the nominal workspace in which the end effector has the full freedom to move, both in translation and in rotation.

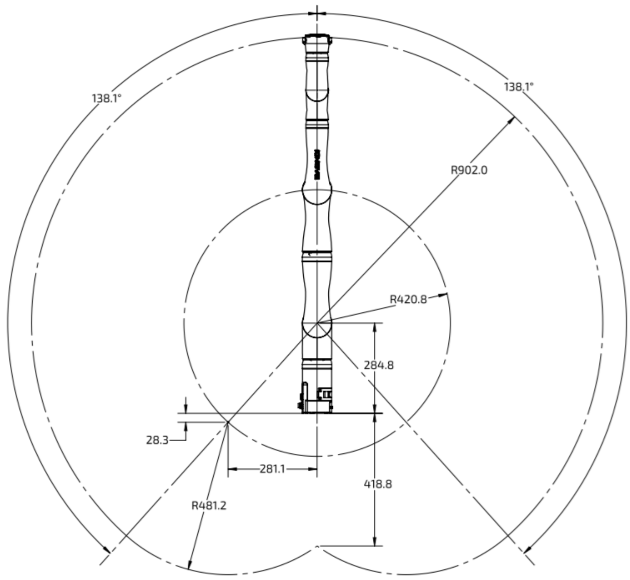

Figure 8 *Nominal Workspace*

Giving a definition of the payload concept as the maximum mass that at the end-effector the robot can hold up, generally it depends on a few factors, such as:

- Radial distance from the base – the payload is high when the end-effector is near to the basis and will decrease when it will move far from the base axis.
- Temporary vs. continuous – the maximum payload can be temporary managed. It is relevant to affirm that the time used to work with different payloads is inversely proportional to the weight of the mass.
    - High mass → temporary work ability.

- Low mass → continuous work ability.

3.2. Software

The robotic system is connected to the computer via Ethernet connection. After establishing the IP connection addresses, the Kinova is in direct connection with the machine.

Various software systems and programming languages can be chosen to program the robot through the PC.

Those that have been used in this paper refer to the web application "KINOVA® KORTEX™ Web App" for high level programming, and to libraries written in C++ provided by the Git channel "KINOVA® KORTEX™ API", for low level robot control.

3.2.1. KINOVA® KORTEX™ Web App

The Web App is a Web GUI that runs on the robot (figure 10). It allows to configure, control and monitor the robot via web from a computer connected to over by Ethernet or Wi-Fi connection.

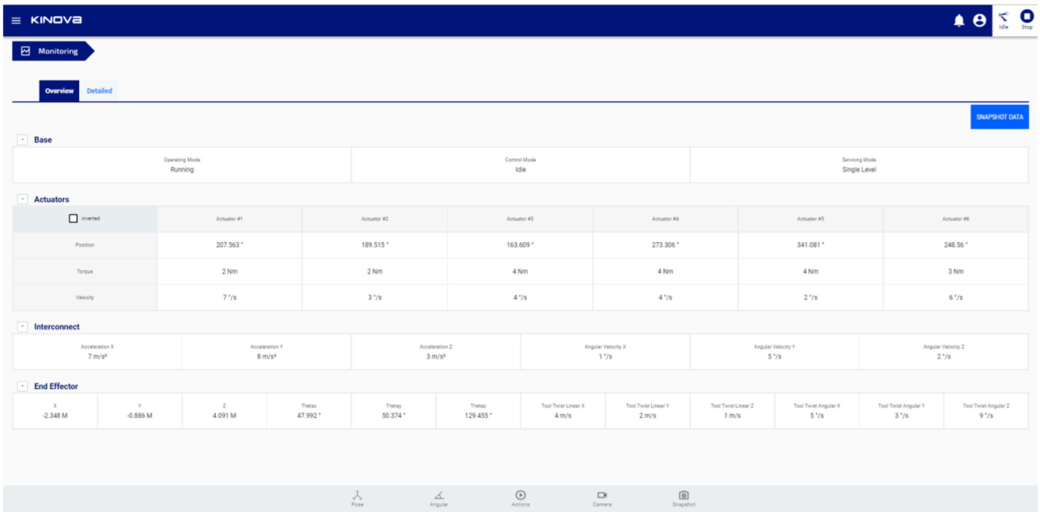

Figure 9 KINOVA® KORTEX™ Web App Overview

From this application, through an easy virtual joystick, it is possible to do a series of important task that allow the user to generate a program able to teach to the Kinova a particular work.

First of all, it is possible to set the admittance configuration, so it is possible to define if the robot has to move in cartesian, in joint or in null-space configuration. After that the user can set the safety areas that define where the robot is free to move and where it has to stop.

The definition of the points to be reached is set from a virtual joystick (figure 11) or directly moving the robot. Once every position of the work trajectory is set it is possible to simulate the movement of the robot by changing its velocity.

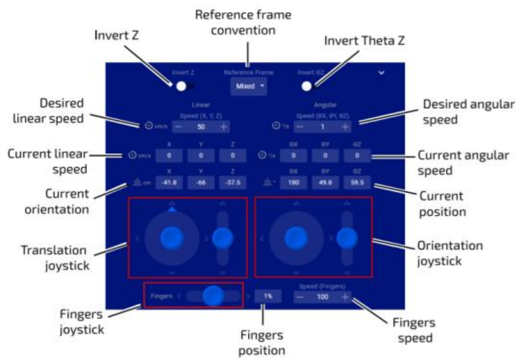

Figure 10 *Virtual Joystick Control*

In general, this web-application is used in order to generate simple program, since is not necessary to have particular programming skills.

3.2.2. KINOVA® KORTEX™ API

KINOVA® KORTEX™ API is the Kinova software framework and application development platform. This framework is useful to configure and control the robot and integrate different products into robotics applications.

APIs are currently provided for the following languages:

- C++
- Python
- MATLAB® (simplified API supporting a subset of Kortex functionality)

Kinova also offers ROS packages covering most of the same functionalities.

In particular, the API groups a series of services which define the available interfaces that can implemented on the various robot devices.

The robot consists of several devices:

- Base controller
- Actuators (each actuator is a distinct device)
- Interface module
- Vision module

"*A service consists of methods and communication exchange data structures. The devices in the robot each implement a particular set of services, some of which are available across multiple devices. The methods available as part of a service on a device are accessed via remote procedure calls (RPC).*"

Scheme of the network:

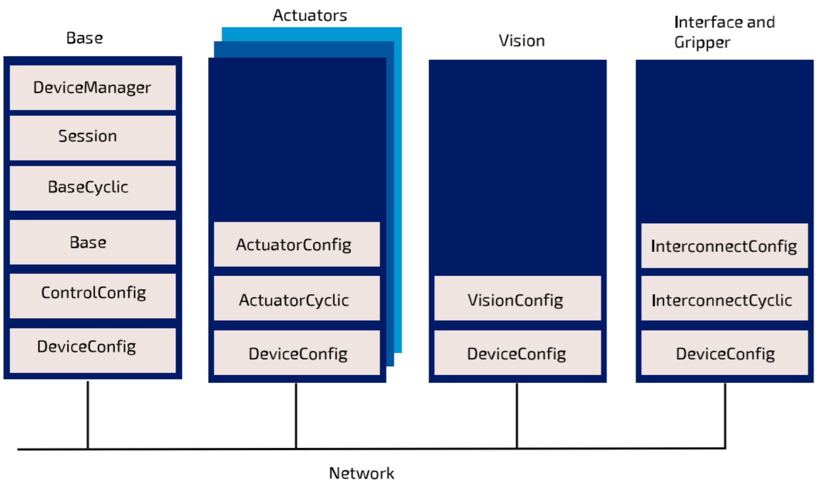

***Figure 11** Kinova API Network Structure*

4. Theoretical Background

In order to catch as well as possible the objective of this thesis it is necessary to clarify some essential theoretical concept, such as the control systems that govern the algorithm, their description, differences, advantages and disadvantages, explaining them with the respective formulas.

During this dissertation the applied control system is an admittance one merged to an impedance control system able to compute the forces applied to the end-effector.

In particular, during this thesis will be exposed the impedance and the admittance control system and how these two methods are developed.

After a brief description of the adopted control systems an explanation of the definition of a Point-to-Point trajectory will be given.

4.1. Impedance

Mechanical impedance is the measure of how much a structure is able to resist motion when is subjected to a harmonic force.

Through the ratio between the applied forces and the velocity of a 1 DoF mechanical system it is possible to define a sort of mechanical impedance, which can be expressed into a frequency domain as:

$$Z(s) = \frac{F(s)}{v(s)} \quad (4.1)$$

4.1.1. Mass-Spring-Damper System

Let's describe the robot as a mass-spring-damper system as is drawn in Figure 12.

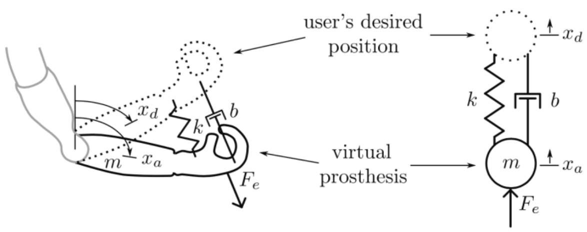

Figure 12 *Model of virtual prosthesis impedance*

The relationship between force and velocity can be expressed as:

$$F = Ma + Bv + Kx \quad (4.2)$$

The Eq. 4.2 reports acceleration and position, which are derivative and integral of the velocity respectively. Expressing Eq. 4.2 in the Laplace domain the result will be:

$$F(s) = (Ms + B + Ks^{-1})v \quad (4.3)$$

From which is defined the impedance as:

$$Z = Ms + B + Ks^{-1} \quad (4.4)$$

4.1.2. Impedance Control (IC)

From the above definition it is possible to assert that by controlling the impedance is also controlled how the robot behaves during an interaction with an external force, by defining its mass-spring-damper parameters.

The gain to control the impedance is to modify the robot's rigidity, that can be classified in two behaves:

Rigid → the robotic arm remains at a given position for any applied external force.

Compliant → deviation from the robot's equilibrium position can be deviated with respect to the applied external force.

By comparing impedance control to other control strategies, it is possible to discover some advantages and disadvantages of this control system.

Comparing impedance with the position control, in the second a certain position is defined, and the robot tries to reach the position no matter what. If the robot is not able to easily reach the position it will apply high forces which might cause damage. Through impedance control, it is possible to indirectly control the force and avoid such damaging high forces.

With respect to force control behaves, if the robot's end-effector is not in contact with another object, the forces will lead to fast movements.

In conclusion the big advantage of impedance control is the possibility to control the motion and the force of the robot's end-effector at the same time.

There are several types of impedance control, such as:

- *Cartesian Impedance Control*: is modeled on a virtual spring damper system, where it is possible to configure the values for stiffness and damping. This spring is extended between the reference and the actual positions of the TCP (Tool Center Point). This allows the robot to react accordingly to external influences.
- *Cartesian Impedance Control with Over-Strength*: specific type of the Cartesian impedance controller. In addition to conformal behavior, it is possible to superimpose constant force reference points and sinusoidal force oscillations. This controller can be used, for example, to implement force-dependent search paths and vibration movements for coupling processes.
- *Single Axis Impedance Control*: stiffness and damping values can be configured for each axis.

4.1.3. Impedance Control Application

The impedance control system has a lot of possible application, and they can be circumscribed in three macro-areas, such as:

- Avoid strong impact forces due to uncertain geometric characteristics of the environment (position and orientation).
- Adapt to the dynamical characteristics of the environment in a complementary manner.
- Mimic the behavior of a human arm (fast and stiff during a free motion, slow and compliant in "protected" motion).

In order to respect one of these usages it is necessary to tune in a right manner the parameters of the dynamic control system, as explained in the Tab 5.

Table 5 Control Parameters and their consequences

Low value of X parameter	*High value of Y parameter*	*Consequences*
M	*K*	*Low contact forces*
K	*M*	*Good trajectory tracking*
B	*//*	*Uncontrolled transient behaviour*
//	*B*	*Controlled transient behaviour*

4.2. Admittance Control (AC)

From the definition of impedance, it is possible to compute its inverse and obtain the equation that governs the admittance control system.

Admittance control can be seen as a form of indirect force control, since the objective of this algorithm is to change the trajectory of the robot with respect to the applied external forces.

This kind of control system is used for human-robot interactions in order to transform external applied forces and torques to the desired position and orientation of the end effector.

Once a desired path is set, the robot will follow the trajectory with the same orientation of the end effector even if an external force is applied, as it is explained in Figure 13.

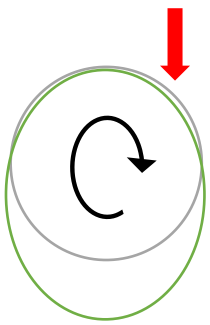

Figure 13 *Admittance Control Behaviour*

-Desired Path

-Applied External Force

-Adjusted Path

The admittance control system is applied when:

- It is not possible to access to low-level robot torque commands, so it is necessary a closed-loop control architecture.
- In order to interact with the environment sometimes is useful to have a direct connection from the contact forces to the velocity commands.

4.3. Differences (IC vs AC)

The main difference between impedance and admittance control is that the first one is able to control the force after that the trajectory is computed, the second one is able to control the trajectory after that a force is computed, so from the same parameters and under the same hardware condition the robot will have a different behavior. Considering the Eq. 4.3 the following figure presents the control scheme for impedance and admittance:

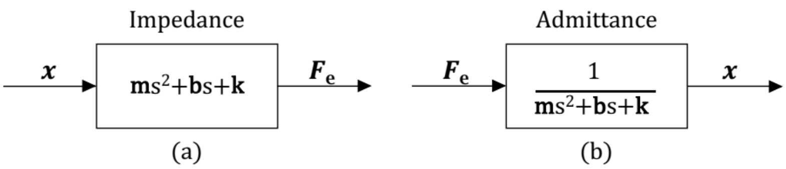

Figure 14 *(a) Impedance Control (b) Admittance Control*

Impedance control is used for manual haptic, and teleoperation displays. Admittance one is used in larger non-back drivable high-friction devices that are of the full-body type (e.g., wearable robotics) and heavy-duty type (e.g., industry).

In the following figures it is also possible to observe the generic control loop of each control system: (a) Impedance Control, (b) Admittance Control.

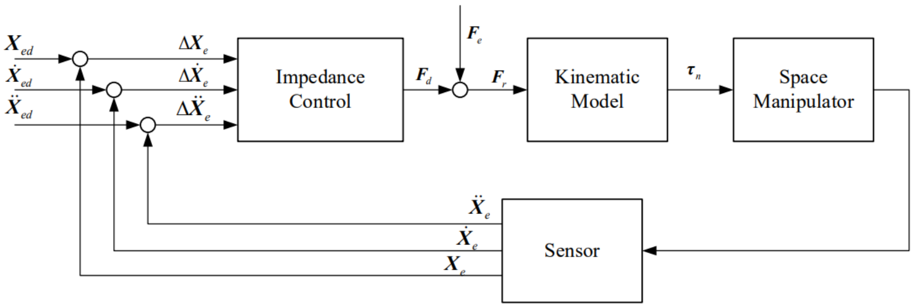

Figure 15 Impedance Control Block Diagram

The desired kinematic information is sent to the controller which will elaborate them and if an external force will be applied to the system, the kinematic model after that will response sending torques to the Manipulator that will sense it and through the sensor it is possible to compute the different kinematical errors that are sent to the impedance controller that will modify the input trajectory. Through this first block diagram, the system is defined Force-Controlled.

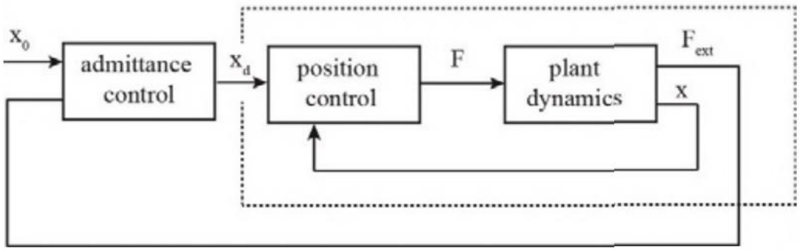

Figure 16 Admittance Control Block Diagram

The desired position is sent to the controller, which will control the position if an external force is applied to the system, this force is computed, and the initial implemented trajectory will be modified in order to reach exactly the desired position. Through this second block diagram the system is defined Position-Controlled.

4.4. Jacobian Transpose

In order to relate the robot's end-effector cartesian space with its joint space it is useful to define the robotic arm's Jacobian Transpose concept.

This matrix is able to compute the inverse kinematics problem through a unique robot configuration.

Considering the dynamics of a robotics arm system, including the total kinetic and potential energy with the absence of external force, it is possible to represent the joint space dynamic model by the following equation of motion:

$$M(q)\ddot{q} + B(q,\dot{q})\dot{q} + F_v\dot{q} + g(q) = \tau \quad (4.5)$$

Since the matrices that describe the above equation are functions of manipulator position, it can be considered as the configuration-space equation.

Thus, expressing the dynamics of the manipulator with respect to Cartesian variable, the Eq. 4.5 will be in the following general form:

$$M_\chi(q)\ddot{\chi} + B_\chi(q,\dot{q}) + K_\chi(q) = \digamma \quad (4.6)$$

Through the above expressions and the definition of the Jacobian:

$$\dot{\chi} = J(q)\dot{q} \quad (4.7)$$

It is possible to observe that the fictitious forces applied to the end-effector, can be also considered as forces applied by the actuators to the joints by using the following relationship:

$$\tau = J^T(q)\digamma \quad (4.8)$$

4.5. End-Effector Forces Calculation

The definition of Jacobian transpose gives a method for designing a trajectory-following control system that allow to the manipulator to follow a desired trajectory.

In order to optimize this computation, it is necessary to guarantee an error tending to zero between the desired and actual end-effector's position.

To ensure that condition it is sufficient to define a proportional–derivative (PD) control law considering the position and velocity vectors as feedback.

Therefore, end-effector forces obtention from the Eq. 4.8 is as follows:

$$\digamma = K_D e + K_P \dot{e} \quad (4.9)$$

4.5.1. Position Error: e

The computation of the position error from the Eq. 4.9 is not as direct as it would be in the joint space by simple subtraction. In Cartesian control, it is necessary to differentiate two different kinds of error, the error in position and in orientation. The first one is calculated as:

$$e_p = p_d - p = \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} - \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (4.10)$$

The error in orientation can be computed in different manners depending on its representation: Euler angles, axis–angle or quaternion. During this essay will be used the rotation matrix representation which can be derived for each case.

Considering $R = [n\ s\ a]$ and $R_d = [n_d\ s_d\ a_d]$ as the actual and desired rotation matrices, the error in orientation is calculated as follows:

$$e_o = \frac{1}{2}(n \times n_d + s \times s_d + a \times a_d) \quad (4.11)$$

Finally, taking Eq. 4.10 and Eq. 4.11, the position error can be expressed as:

$$e = \begin{bmatrix} e_p \\ e_o \end{bmatrix} \quad (4.12)$$

4.5.2. Velocity Error: $\dot{e}$

The end-effector linear and angular velocity error is obtained directly by the subtraction of desired and actual velocity in Cartesian space:

$$\dot{e} = v_d - v = \begin{bmatrix} \dot{p}_d \\ w_d \end{bmatrix} - \begin{bmatrix} \dot{p} \\ w \end{bmatrix} \quad (4.13)$$

Through the geometric Jacobian and the joint velocities, if the relation mentioned in Eq. 4.7 is applied the end-effector velocity can be obtained directly.

4.6. Point-to-Point Trajectory

There exist a lot of methods in order to plan the trajectory of a manipulator in order to reach a desired position. When the objective is only to reach the final position and not to follow a desired path it is common to plan the trajectory in joint space, implementing a linear path from the home position to the desired one, maintaining constrained the time law by a defined maximum velocity and acceleration values.

The PTP trajectory is obtained implementing a linear combination of the initial and final position values, such as follow:

$$\pi'(q(t)) = (1 - s(t))q_0 + s(t)q_f = q_0 + s(t)(q_f - q_0)$$
$$= q_0 + s(t)\Delta q \quad (4.14)$$

This approach allows that the motion of all joints starts and ends at the same time instants, providing a smoother motion of the system, avoiding undesired jerks that can introduce undesirable vibrations.

From the Eq. 4.14 is shown that this trajectory planning method requires only the design of the time law for the profile abscissa.

Assuming the velocity and acceleration constraints as:

$$\dot{q}(t) = \dot{s}(t)(q_i - q_{i-1}) \quad (4.15)$$

$$\ddot{q}(t) = \ddot{s}(t)(q_i - q_{i-1}) \quad (4.16)$$

The simple kinematic elements profiles are defined as:

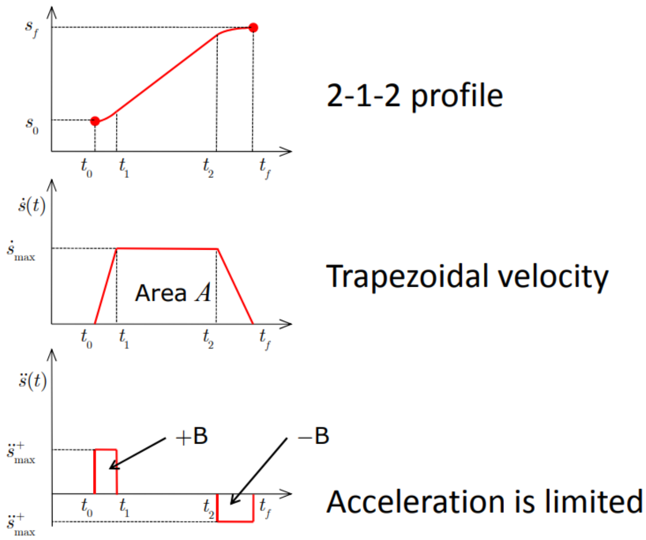

***Figure 17** Simple PTP Profile*

5. Software Implementation

The development procedure of any algorithm and/or software is structured in six steps that create a real product life cycle. The steps that must be followed are as follows:

- Requirement's specification: provide a high-level description of a software product to be developed, including its purpose, its features, key performance parameters and behaviour. As such, it essentially acts as a road map that guides the development process and keeps everyone on track.
- Feasibility study: tool for defining whether an idea or solution can be technically and legally realised, supported by the organisational and management structure of the client, as well as being economically viable.
- Analysis and Design: the first is the study of "what" the system should do considering the logical point of view. The design is the study of "how" the system is to be implemented with respect to the technical point of view.
- Implementation: the implementation may concern the individual modules that make up the system, which then have to be integrated with each other. Implementation is also called development or coding of the software product, as it is the realisation phase, which materialises the software solution through programming, i.e., the writing of programmes by programmers or developers.
- Integration and testing: the integration phase concerns the delicate step that occurs when the code developed for all components of the system is integrated to create the final product. The testing phase, in traditional development models, is planned at the end of the code development, and may serve, for example, to verify the correspondence of the software functionalities with the user's requirements, or if there are any defects that could jeopardise the correctness of the operation, or even to check if the level of usability of the programme is valid or compromises the productivity of a possible user. Software testing is essential to guarantee

the quality of the software and indispensable to ensure a satisfactory user experience.

- ❖ Maintenance: software maintenance is the process of modifying a software product after its release into service. It is necessary to:
  - ✓ to eliminate malfunctions
  - ✓ improve performance or other quality attributes
  - ✓ adapt it to changes in the operating environment

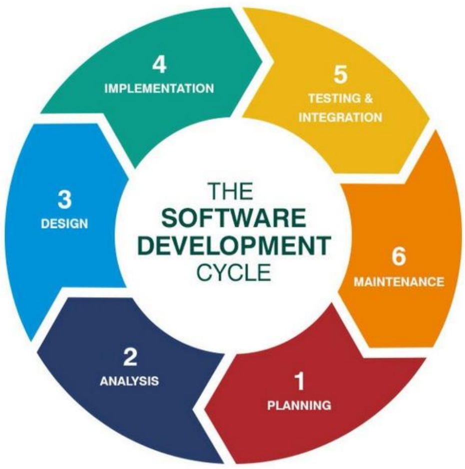

Figure 18 *Software Development Cycle*

Being an experimental study, the aim of this and the following chapter is to explain in detail the design and implementation of the software and then report the results obtained in the tests.

5.1. Control of a Joint with Adaptive Admittance

Admittance control is a system useful to handle the reaction of the robot when a contact force is applied against its end-effector. Broadly speaking is referred to as a "position-based impedance control."

Typical implementation of an admittance controller engages a force sensor attached on the end-effector, and it consists of a virtual object representing a simple dynamic, typically a damped mass element, and a high-gain position controller.

The position this virtual object is refreshed according to the force sensor measurement and the programmed force. The resultant virtual position is used as the position final position to be reached. The advantage of this controller structure is that the internal position controller overcomes the hardware dynamics, such as joint friction.

The objective is to control the position and admittance of just one joint of KINOVA Gen3, that means to program joint positions and moreover, to change the robot's rigidity in real-time.

Metodology

KINOVA is set to low-level to command actuator positions at a frequency of 1000 Hz. Joint 6 is the one controlled, chosen due to its similarity with a human elbow.

The reference position is set through the software. Robot's rigidity can be changed by applying an admittance control. The idea of adopting different stiffness for each situation requires the use of an external sensor to control its level. This project proposes an adaptive admittance control which allows adapting the robot's rigidity respect to applied external forces.

In particular, the following figure shows the block diagram proposed to control the position of joint number 6 with adaptive admittance:

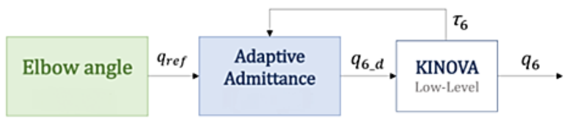

Figure 19 *Joint 6 Adaptive Admittance Control*

The adaptive admittance block employs a method exposed in the theoretical background adjusted to the thesis purpose. Its detailed control scheme is shown in the following figure:

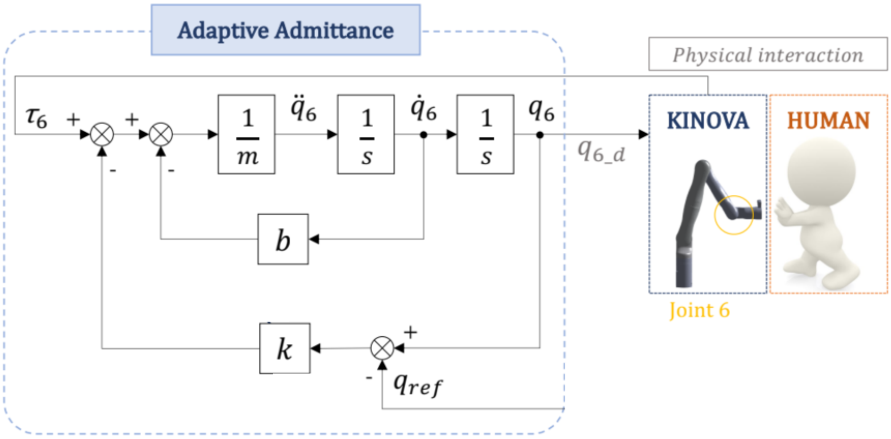

Figure 20 Detailed Adaptive Admittance Control Scheme

The diagram shows that admittance control imposes value of position resulting from an input force. In this approach, position q_{6d} is determined by the environment force τ_{6d} acting on the robot's actuator number 6.

5.2. Cartesian Position Control

Before proceeding with the explanation of how cartesian position control was implemented in the software, it is necessary to provide some preliminary system and theoretical specifications.

Cartesian space is an advantage for the end user as it is more natural for him to identify Cartesian coordinates (x, y, z) than joint displacements (q1, q2, . . ., q_n).

For this reason, it is important to describe the characteristics and properties of Cartesian space.

The analysis of Cartesian space leaving joint space begins by considering inverse kinematics, which is one of the basic functions for manipulator robot control systems.

Inverse kinematics is the process that determines the joint parameters of an object based on Cartesian position, which is described as a function f on the joint variable q:

$$x = f(q) \quad (5.1)$$

By using the definition of Jacobian written in Eq. 4.7 the joint velocity representation is obtained as follow:

$$\dot{q} = J(q)^{-1}\dot{x} \quad (5.2)$$

Finally, after some operation it is possible to relate the joint space with the cartesian space such as:

Table 6 *Joint and Cartesian Space*

JOINT SPACE	***CARTESIAN SPACE***
$\dot{q} = J(q)^{-1}\dot{x}$	$\dot{x} = J(q)\dot{q}$
$\ddot{q} = J(q)^{-1}\ddot{x} - J(q)^{-1}\dot{J}(q)\, J(q)^{-1}\dot{x}$	$\ddot{x} = J(q)^{-1}\ddot{q} + \dot{J}(q)\, \dot{q}$

As it is shown, partial derivation on the inverse kinematics models returns a relationship between the joint and the cartesian velocity.

In general, the inverse Jacobian matrix is required to study on singular position the robot manipulator.

In conclusion, when controlling specific movements, being able to give the desired end-effector orientation and position in Cartesian space is very useful.

Metodology

To implement a Cartesian position control means to determine desired robot positions in Cartesian variables: pose (x, y, z) and orientation (for example Euler angles: roll, pitch, yaw).

KINOVA's low-level control API grants a command for each actuator in the robot that allows to set position, velocity or torque joints parameters.

Achieving this control in KINOVA requires to apply the desired positions in Cartesian space as a robot's configurations in joint space.

This implementation requires to relate the robot's end-effector Cartesian space with its joint space.

Into the Theoretical Background chapter, it is explained a methodology that maps Cartesian forces acting to the robot (F) into equivalent joint torques (τ). The figure below shows graphically this relationship applied to KINOVA:

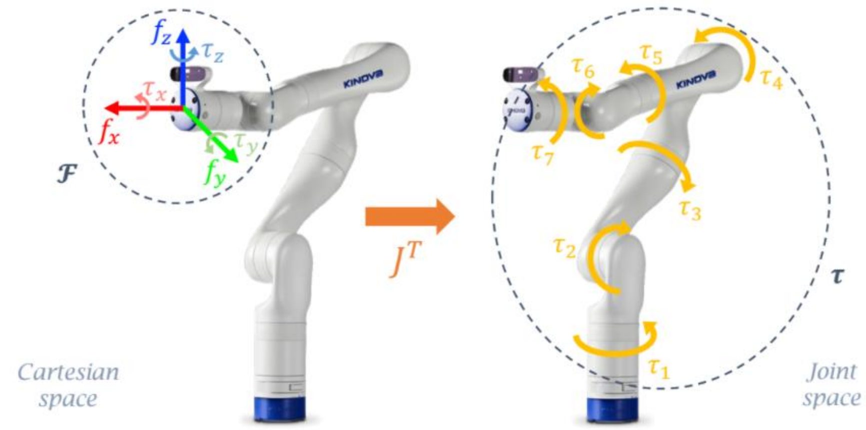

Figure 21 *Joint and Cartesian Space References applied to Kinova*

The Cartesian position control uses the Jacobian transpose (J^T) to program the robot in torque configurations (τ_d). Controlling a robot by torque commands implies to consider its gravitational compensation. The following figure shows a detailed scheme of this control's implementation:

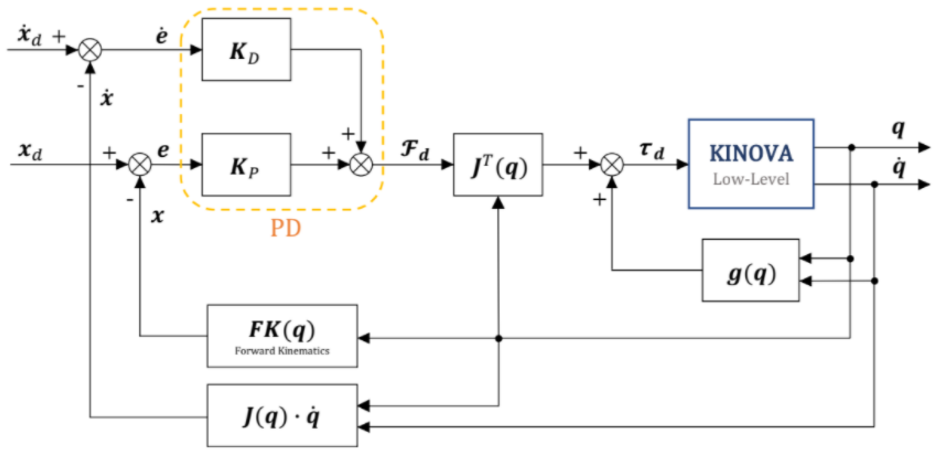

Figure 22 *Double Control System Scheme*

From the scheme, Forward Kinematics and geometric Jacobian have been computed to obtain, respectively, KINOVA's position (x) and velocity ($\dot{x}$) in the Cartesian space.

This implementation applies a proportional-derivative (PD) controller. It takes the end-effector position and velocity errors ($e, \dot{e}$) and obtains the end-effector forces (F_d) needed to follow the position reference (x_d). K_P and K_D ∈ R 6x6 are diagonal matrices, that denote the coefficients for the proportional and derivative terms respectively. The K_P gains are tuning parameters and the ones of K_D are computed from them:

$$K_D = 2\sqrt{K_P} \quad (5.3)$$

5.3. Control System Block Diagram

In order to achieve the final position under all boundary conditions, it was necessary to link the two control systems analysed above, since although they perform two apparently different roles, they maintain the trajectory of the Kinova such that it can always be positioned, albeit with a few millimetres of error, in the desired position.

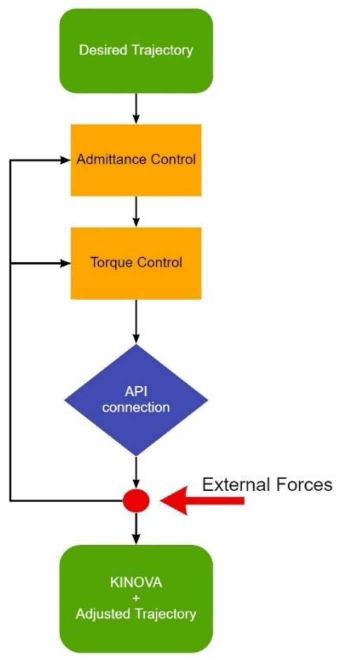

Figure 23 *Control System generics block-diagram*

After defining the trajectory to be performed, writing it in one of the many functions of the code, it will be sent to the admittance control.

$$q = desired\ position$$

$$\dot{q} = desired\ velocity$$

$$\tau = desired\ torque$$

At time t=0 the control system will keep the initial conditions unchanged, since it has not yet received any feedback from the force sensors, which as soon as they perceive an input signal from an external system send a signal that will reach the admittance control that in this case will convert the ideal trajectory into an admittance trajectory, such as:

$$q_{adm} = admittance\ position = q_{adm} + \dot{q}_{adm} * dt \quad (5.4)$$

$$\dot{q}_{adm} = admittance\ velocity = \dot{q}_{adm} + \ddot{q}_{adm} * dt \quad (5.5)$$

$$\ddot{q}_{adm} = admittance\ acceleration$$
$$= M^{-1} * (F_{ext} - B * \dot{q}_{adm} - K * (q_{adm} - q_{home}) \quad (5.6)$$

The second step is the torque control, which always receives force signals from the robot's sensors and adjusts them so that the force the robot applies to the outside is constant, keeping errors in speed and position to a minimum.

$$q_{cs} = control\ signal\ position$$

$$\dot{q}_{cs} = control\ signal\ velocity$$

$$\tau_{cs} = J^T * \left(K_p * e_p + K_d * e_d\right) \quad (5.7)$$

Finally, everything is kept connected by commands from the Kortex API, which sends these instructions to the robot, which processes them and outputs the adjusted trajectory.

The control therefore adapts perfectly to two working planes, the Cartesian plane for receiving position inputs and the joint plane for optimising the trajectory based on external forces.

The final system is illustrated in a much more technical way than in figure 24:

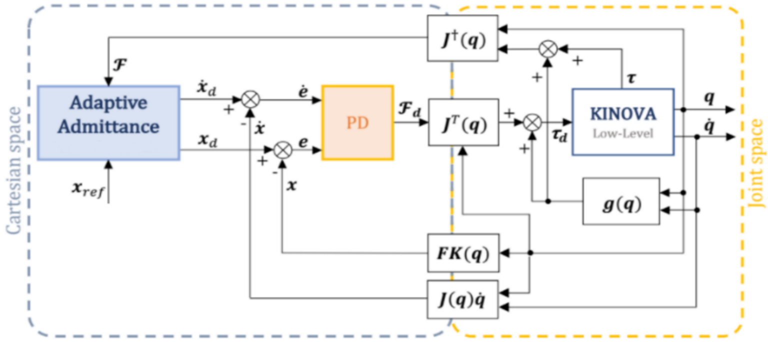

Figure 24 Cartesian and Joint Space Control System

5.3.1. Preliminary Analysis

Before proceeding with the explanation of the various functions that have been implemented and the reason why certain values have been set for certain parameters, it is necessary to report on the behaviour of the mechanical arm as the control parameters change.

In the following tab are reported the different behaviours of the Kinova by changing the admittance control parameters (M, B, K) or the torque control ones (K_P,K_D).

Table 7 Preliminary Parameter Analysis

M	B	K	K_P	K_D	Behaviour
[2,2,2]	[20,20,20]	[200,200,200]	[1000,100]	[500,30]	Difficulty by moving along the path.
[2,2,2]	[3,3,3]	[100,100,100]	[1800,50]	[130,20]	Good following of the trajectory. The gravity is not completely compensated.
[2,2,2]	[3,3,3]	[200,200,200]	[1800,50]	[130,20]	Good following of the trajectory. The gravity is completely compensated.

K_P and K_D (proportional and derivative gain respectively) are the parameters that affect the control through a control signal that is sent to the robot.

$$\tau_{CS} = J^T * \left(K_P * e_{pos} + K_D * e_{vel}\right) \quad (5.8)$$

Since these two gains are computed from other two parameters that regards the orientation and the position, a definition of these values is needed.

$$k_{iPos} = i\ gain\ element\ of\ position$$

$$k_{iOri} = i\ gain\ element\ of\ orientation$$

$$K_i = \begin{bmatrix} k_{iPos} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{iPos} & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{iPos} & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{iOri} & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{iOri} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{iOri} \end{bmatrix}$$

In particular once defined the K_P elements, as k_{PPos} and k_{POri}, the computation of the derivative gain elements is computed by:

$$k_{DPos} = 1.6\sqrt{k_{PPos}} \quad (5.9)$$

$$k_{DOri} = 0.4\sqrt{k_{POri}} \quad (5.10)$$

From the above results the optimal solution was to set the controller on the orientation as the minimum magnitude that allows the Kinova to maintain the orientation (200) of the end-effector even if the position and its controller is changing. Regarding the position, it is observed that could be relevant to analyse the values of position parameter from the magnitude of 200 to 3200.

Starting a program able to move the robot along the Y-axis by changing k_{PPos} the following results are obtained:

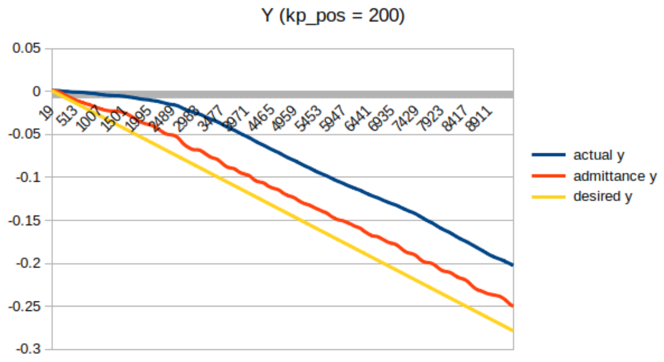

***Figure 25** Y - axis Kp Position = 200*

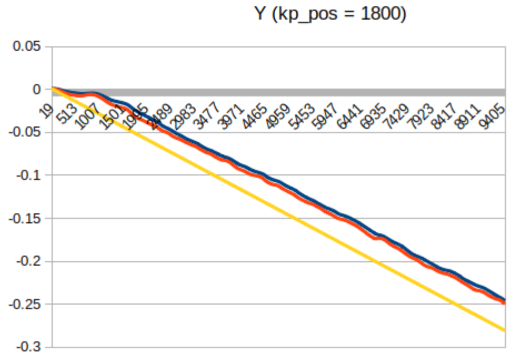

***Figure 26** Y - axis Kp Position = 1800*

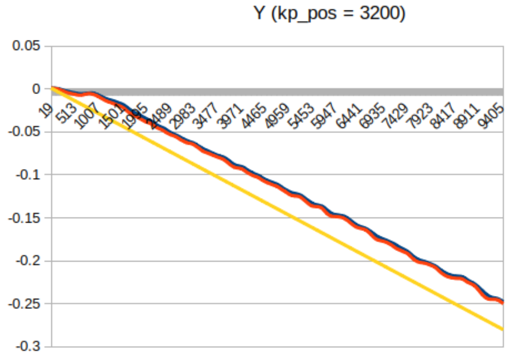

***Figure 27** Y - axis Kp Position = 3200*

From which will be defined the final value that will be set in the next step of this thesis. Since the difference between 1800 and 3200 is minimal, the $k_{PPos} = 1800$ will be used.

5.4. Adopted Functions

With the help of a flow-chart, this section will illustrate and explain the main functions that have been implemented in the code.

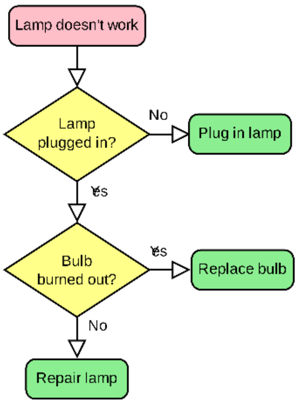

Figure 28 *Generics Flow-Chart*

In C++, after defining the libraries to be used, the absolute variables are usually inserted, in this case the connection variables (IP address, duration of the program, number of actuators of the external system).

The final program is defined within a main function, which 'calls' everything written outside it in the order in which the programmer elaborates this final method.

In order to be able to call external functions, it is necessary to write them outside the main.

It is precisely these actions that allow the robot to be controlled.

The principal function that are adopted are:

- Set torque control parameters
- Set admittance control parameters
- Set servoing mode
- Set control mode
- Admittance trajectory

5.4.1. Set Servoing Mode

"*A servoing mode is a modality through which commands are transmitted to robot devices during operation.* "

The details engaged in controlling via the API will be different with respect to the servoing mode. Kinova Gen3 API associates two servoing modes: High-level and Low-level.

High-level servoing allows to control the robot by sending it a kinematic information (position or velocity) via an API method which is sent once. The robot control library calculates inverse kinematics and applies geometrical limits (protection zones, singularity management, self-collision avoidance).

```
void set_level_servoing_mode (k_api::Base::ServoingMode level,
k_api::Base::BaseClient* base)

{
    auto servoingMode = k_api::Base::ServoingModeInformation();
    servoingMode.set_servoing_mode(level);
    base->SetServoingMode(servoingMode);

}
```

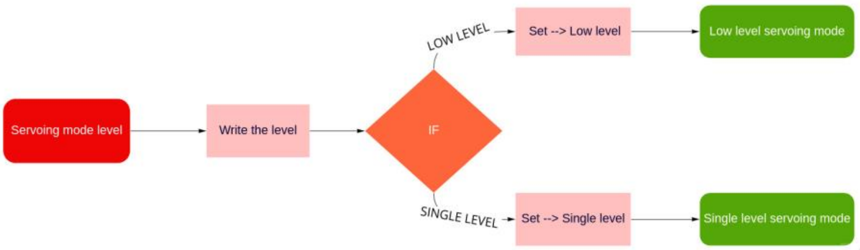

Figure 29 *Set Servoing Mode Flow-Chart*

5.4.2. Set Control Mode

Each robot is characterised by different control modes, these can be determined as certain trajectories are executed. The Kortex API offers the possibility to take advantage of different control modes such as:

- Angular mode: each axis of the manipulator is controlled separately.

- Cartesian mode: cartesian space tool's speeds control.
- Cartesian mode of admission: tool is guided by an external force.
- Force control: tool is controlled by key commands.
- Joint admission mode: joints are rotated by an external force.
- Zero clearance mode: the configuration of the robot can be changed by applying external forces to the links, but the position of the tool remains constant.
- Torque control: the robot joints trajectories are controlled by sending low-level torque commands.
- Trajectory mode: once defined an end point the robot is controlled to reach it.

```
void select_control_mode (k_api::ActuatorConfig::ControlMode
controlmode, k_api::ActuatorConfig::ActuatorConfigClient*
actuator_config, int AC)

{
 auto control_mode_message
 k_api::ActuatorConfig::ControlModeInformation();
 control_mode_message.set_control_mode(controlmode);

 for (int id = 1; id < AC+1; id++)
 {
  actuator_config >SetControlMode(control_mode_message, id);
 }
}
```

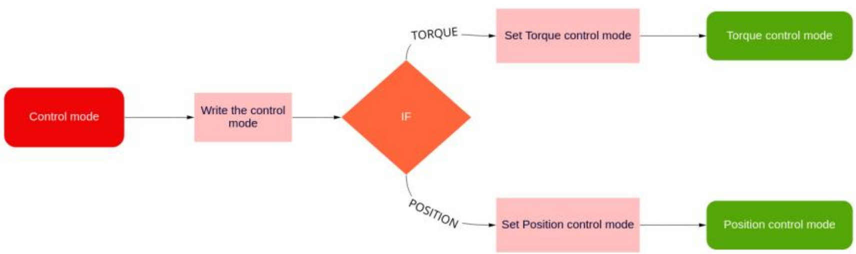

Figure 30 *Set Control Mode Flow-Chart*

5.4.3. Set Torque Control Parameters

As reported in section 5.3. in order to activate the torque control, it is necessary to define its parameters that compute the proportional and the derivative gain.

```
void set_torque_controller_parameters (Eigen::MatrixXd &Kp,
Eigen::MatrixXd &Kd)
{

    double kp_pos, kp_ori, kd_pos, kd_ori;
    Eigen::VectorXd Kp_vec(6), Kd_vec(6);
    cout<<"define kp_pos"<<endl;
    cin>>kp_pos;
    cout<<"define kp_ori"<<endl;
    cin>>kp_ori;
    Kp_vec << kp_pos,kp_pos,kp_pos,kp_ori,kp_ori,kp_ori;
    Kp = Kp_vec.asDiagonal();
    cout<<"Kp="<<Kd<<endl;
    kd_pos = 2*0.8*sqrt(kp_pos);
    kd_ori = 2*0.2*sqrt(kp_ori);
    Kd_vec << kd_pos,kd_pos,kd_pos,kd_ori,kd_ori,kd_ori;
    Kd = Kd_vec.asDiagonal();
    cout<<"Kd="<<Kd<<endl;
}
```

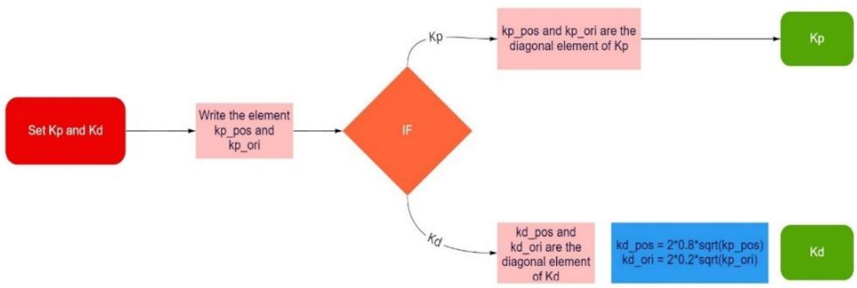

Figure 31 *Set Torque Controller Parameter Flow-Chart*

5.4.4. Set Admittance Control Parameters

As reported in section 5.3. in order to activate the admittance control it is necessary to define its parameters that compute the Mass, Spring and Damper matrix.

```
void set_admittance_control_par (Eigen::MatrixXd &M,
Eigen::MatrixXd &B, Eigen::MatrixXd &K)
   {
         double m1, m2, m3, k1, k2, k3, b1, b2, b3;
         Eigen::Vector3d m_vec, b_vec, k_vec;

         cout<<"define m_x"<<endl;
         cin>>m1;
         cout<<"define m_y"<<endl;
         cin>>m2;
         cout<<"define m_z"<<endl;
         cin>>m3;
         cout<<"define b_x"<<endl;
         cin>>b1;
         cout<<"define b_y"<<endl;
         cin>>b2;
         cout<<"define b_z"<<endl;
         cin>>b3;
         cout<<"define k_x"<<endl;
         cin>>k1;
         cout<<"define k_y"<<endl;
         cin>>k2;
         cout<<"define k_z"<<endl;
         cin>>k3;
         m_vec << m1, m2, m3;
         b_vec << b1, b2, b3;
         k_vec << k1, k2, k3;
         M = m_vec.asDiagonal();
         B = b_vec.asDiagonal();
         K = k_vec.asDiagonal();
         cout<<"M="<<M<<endl;
         cout<<"B="<<B<<endl;
         cout<<"K="<<K<<endl;
   }
```

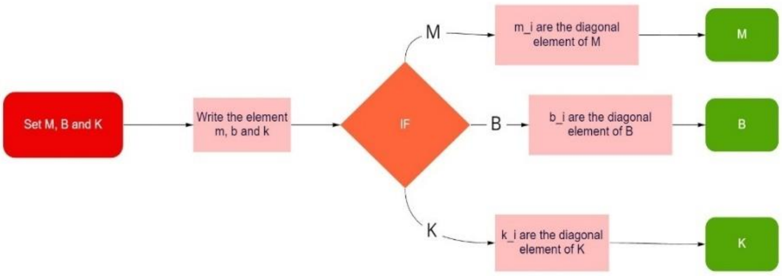

Figure 32 Set Admittance Control Parameter

5.4.5. Admittance Trajectory

From the admittance control parameters, the wrench force computed by the force sensors and the kinematic admittance information, it is possible to define the function that returns the admittance trajectory.

```
void admittance_fcn (Eigen::VectorXd wrench_force, Eigen::Vector3d
&p_d, Eigen::MatrixXd M, Eigen::MatrixXd B, Eigen::MatrixXd K,
Eigen::Vector3d &vel_d_admittance, Eigen::Vector3d
&pos_d_admittance, double dt, Eigen::Vector3d &accel_d_admittance)
{
     //ACCELERATION
    accel_d_admittance = M.inverse()*(wrench_force -
    B*vel_d_admittance - K*(pos_d_admittance-p_d));

    // VELOCITY
    vel_d_admittance = vel_d_admittance + accel_d_admittance*dt;

    // POSITION
     pos_d_admittance = pos_d_admittance + vel_d_admittance*dt;

}
```

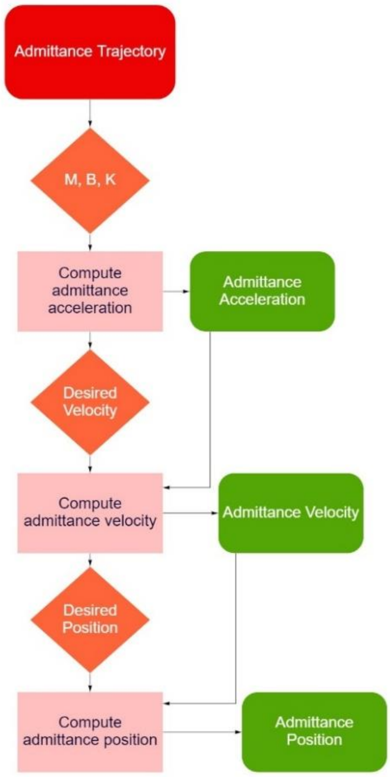

Figure 33 *Compute Admittance Trajectory*

6. *Analysis and Results*

The development of a control system capable of guaranteeing precision in reaching a given position and safety in human-robot interaction requires not only parametric analyses to determine the value of the control parameters in different situations, but also theoretical demonstrations to validate the final solutions.

In particular, this chapter has been structured in two different curricula. The first analysis, named static analysis, is based on tests developed by keeping the end effector in equilibrium with respect to its p home. The second analysis, named dynamic one, was carried out by setting the movement of the robot along an interpolated trajectory.

These two types of situations are closely related to each other, since from the static experiments it was possible to obtain the tuned values of the control parameters that make human-robot collaboration possible when the Kinova has to reach a desired position.

In order to obtain valid results, the various experiments, taking into account two control systems working simultaneously, it was necessary, as specified in the previous section, to keep the elements that govern proportional and derivative gain values constant, as shown below:

$$k_{PPos} = 1800$$

$$k_{POri} = 200$$

As already mentioned, these are the optimum values that allow the end-effector via torque control to maintain the required orientation and position.

Considering what has been said so far, it is easy to see that the parametric and analytical analyses to be carried out will be based on the optimisation of the admittance control parameters (M, B, K), thanks to which it will be possible to check whether the software is able to manage both trajectory and impact control.

6.1. Static interaction

A mass-spring-damper system controlled by a special programme can undergo variations on its three main parameters.

The aim of this first experiment is to keep the robot in its home position, applying first an instantaneous force and then a constant one, while the values of B and K are varied. This is necessary in order to study the behaviour of the robot so that optimal values can be defined to respect safety criteria and constraints on the trajectory.

The study is intended to be gradual and differentiated, so the first types of analysis were carried out by keeping one of the two parameters equal to zero, modifying the free one accordingly, while a final analysis is carried out in order to determine the optimal trio of parameters.

As it will be possible to observe later on, the inertia matrix is not considered in the analyses since the acceleration is null, therefore the behaviour of the system remains unchanged for every value of M, as long as it is greater than zero.

Figure 34 *Kinova Gen3 Ultra Lightweight*

6.1.1. Damping equal to zero (B = 0)

The first static experiment consists of modify the admittance control system by set the damping parameter equal to zero.

From the admittance control equation, it is possible to observe that the damping parameter is strictly related to the error in velocity of the robot:

$$F_{ext} = M(a - a_{des}) + B(v - v_{des}) + K(p - p_{des}) \quad (6.1)$$

The equation written above allows to make one important consideration. If B is set equal to zero and an external force is applied to the system, the velocity will be non-correctly controlled, and the system will have a weird behaviour, maybe affected by undesired oscillations.

During the experiments human applies an external force along Y-axis to move the robot from p_{home} to p_{des}. If the end-effector is controlled directly by the human hand, the manual displacement works correctly, but as soon as the hand releases the end-effector, undesired oscillations affect the trajectory, and the system will become unsafe for human-robot interaction.

The following graph show how is affected the position.

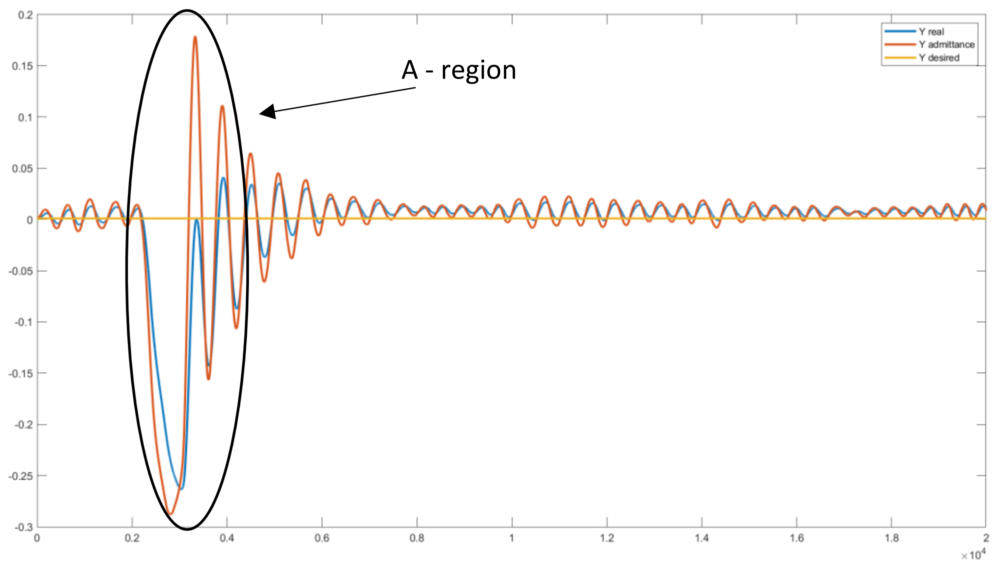

Figure 35 B = 0 Behaviour

The yellow line represents the desired position and the blue and red ones, the real and the admittance position respectively, report the oscillations that the system has when the damping parameter is set equal to zero.

As soon as the system, moved to another position by an external force, is released, the amplitude of the oscillations will increase instantaneously. This is the reason why there is a big difference in region A between the real and the admittance behaviour that is characterized by the amplitude of the oscillations.

By analysing the behaviour as K varies, it was possible to observe that the amplitude of the oscillations and the propagation time of these are closely linked to the magnitude of K.

For values of K higher than 500 the oscillations are minimal and with a short propagation time, while decreasing the value of the stiffness even by only 20% compared to the one previously set, the system will have very large oscillations for a high propagation time.

6.1.2. Stiffness equal to zero (K = 0)

The second static experiment consists of modify the admittance control system by set the stiffness parameter equal to zero.

From the admittance control equation, it is possible to observe that the stiffness parameter is strictly related to the error in position of the robot:

$$F_{ext} = M(a - a_{des}) + B(v - v_{des}) + K(p - p_{des})$$

The equation written above allows to make another one important consideration. If K is set equal to zero and an external force is applied to the system, the position will be non-controlled, and the system will define as desired position the last position of the end-effector. Thus, what will happen is that one the robot is moved from the home position to another one, it will stop there.

The stiffness parameter is also the parameter that directly defines the elastic force, so the force of the actuator. With low value of K the robot will be compliance to move from one position to other, but with high value of K, could be possible that the robot does not move from the home position to another since a high external force is required.

The phenomena described above is exactly what occurred to the Kinova without stiffness.

As an external force is applied, to move the end-effector from the home position to another one, the robot stopped and remained in the adjusted position until the application of another external force.

From the following pictures it is possible to observe that null-stiffness behaviour:

Figure 36 *K = 0 movement*

As it is reported by the figure 33 the desired position (yellow line) remains constant, since it is not affected, but the real and the admittance one will be affected and in region A and B will remain constant, since due to the null stiffness the control system is not able to return the end-effector to the home position.

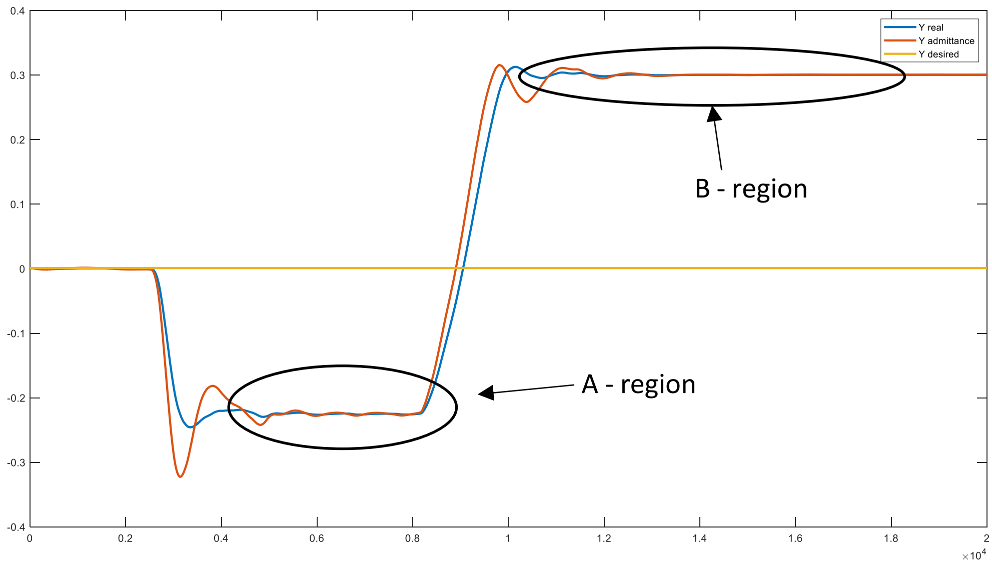

Figure 37 *K = 0 Behaviour*

From the graph it is also possible analyse the control system that is able to maintain constant reached position once the robot is moved by an applied external force, reporting a small overshoot[1] at the end of the application of the force.

In this situation, a variation of B, maintaining K = 0, does not affect the behaviour in the sense of return to the home position.

6.1.3. Constant B with different K

Up to now, parameters B and K have been analysed individually, and it has been possible to observe how the cancellation of just one of them drastically compromises the control of the system.

While setting B equal to zero implies a risky behaviour, therefore unusable under collaborative conditions, cancelling the stiffness can be advantageous if one wants to take full control of the robot in order to move it freely from one position to another.

The results obtained in section 6.1.1 and 6.1.2 allow us to understand that, in order to have a behaviour without oscillations, it is necessary to set a constant

[1] Difference between the maximum point reached by a signal and the constant value that it maintains during the time.

value of B, while in order to obtain a system in static equilibrium, the parameter K must not be equal to zero, but greater.

In this last section of static analysis, we will report the study carried out on the Kinova keeping B = 250, the optimal parameter recommended by the Kinova guide, and varying K.

The variation of K allows to analyse both the response in position, in order to maintain the desired equilibrium, and the response in force. The latter allows to understand what is the maximum value of K that can be chosen to minimise the force applied by the robot against any external force.

The test consists in moving the robot from the home position to a desired one along the Y-axis by changing K from 50 to 3000 in order to obtain results about the wrench force during this displacement.

Figure 38 *Displacement of end-effector*

As it is possible to observe from the figure 36, by increasing K, increases linearly the wrench force, as it is reported in the following table.

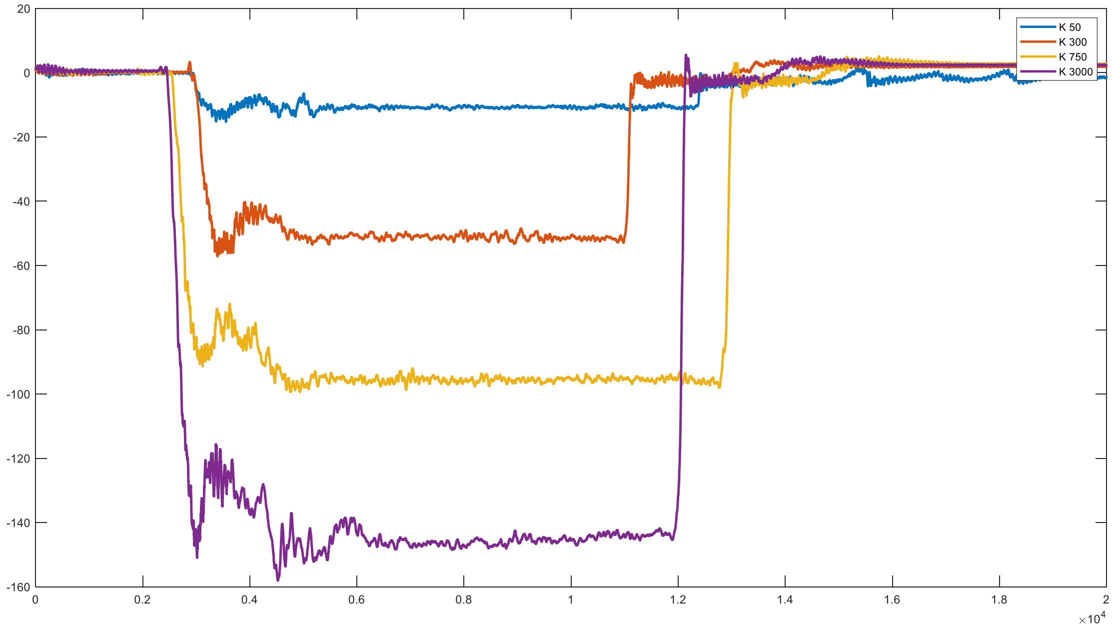

Figure 39 *Force values by increasing K*

Table 8 *Force with respect to K*

K value	**Actuator Force**
50	5
300	50
750	100
3000	150

From this table it is possible to define a range of K that is safe if applied to a control system that governs a real human-robot interaction, which range is from 50 to a value near to 300.

Excluding human-robot interaction it is also possible to observe that for values higher than 750 the force generated by the actuator does not increase so much as from K = 50 to K = 750, so in conclusion the trend resembles that of a logarithmic function.

6.1.4. Static Interaction Conclusion

From the sections 6.1.1. and 6.1.3. a definitive conclusion can be associated in order to proceed to the next kind of analysis.

First of all, the magnitude of B must be higher than zero, preferably equal to 250 as the Kinova's document suggest, after that the magnitude of K depends on what is the objective of the experiment, since for high value the error in position will be null, but the generated forces are dangerous and on the other hand for low value of K an error in position is reported but it occurs in a safety condition.

In conclusion the following table will describe every possible situation from which it is possible to choose a defined value for B and/or K with respect to the final desired behaviour.

Table 9 *Kinova's behaviour by changing B and K*

B	**K**	**Behaviour**
0	Constant	Oscillations (no safety condition)
Constant	0	Non-Controlled error in position
Constant	< 300	Safety condition with error in position
Constant	> 300	Non safety condition without error in position

6.2. Move to Position

Both collaboration and human-robot assistance require a kind of autonomy on the part of the machine, especially if it has to perform precision tasks.

Taking a pick and place operation as an example, this may require a high rate of precision if objects have to be released in a precise position, or low if the release takes place in a space with a much larger surface area than the size of the object.

Considering this example as a starting point, it was decided to develop a simple trajectory capable of positioning the end-effector from p_{home} to $p_{desired}$ as shown below (coordinates are expressed in meters):

$$p_{home} = \begin{bmatrix} 0.447 \\ 0.003 \\ 0.431 \end{bmatrix}$$

$$p_{desired} = \begin{bmatrix} 0.447 \\ -0.251 \\ 0.431 \end{bmatrix}$$

As it is possible to observe the trajectory defines an interpolation of points along Y-axis of almost 25 cm.

The objective of this last analysis is to define an optimal couple (B and K) for which the system will have a minimum error in position working in safety condition for human. Optimisation of robot control, however, also relies on factors other than position or safety, such as the operation cycle time. This is the reason why a further analysis to be carried out concerns the time taken to reach the desired position as the admittance parameters change, assuming a constant speed ($v_{desired} = 3\ cm/sec$).

6.2.1. Parameter optimization with respect to $p_{desired}$

Kortex API libraries allow the establishment of the kinematics of the robot, by setting the acceleration and speed. As the aim is to check that the desired position is reached correctly the speed is bound to the final position, such as:

$$v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Where v_i is equal to:

$$v_i = \frac{[p_{final(i)} - p_{initial(i)}]}{d} * v_{desired} \quad (6.2)$$

And d is:

$$d = \sqrt{\sum (p_{final(i)} - p_{initial(i)})} \quad (6.3)$$

Once the trajectory has been defined, it is possible to proceed with testing. In order to obtain an optimal result with regard to the attainment of the position, it was decided to proceed with a purely analytical analysis.

From the admittance control formula, it can be seen that the position error is directly related to the K parameter.

Assuming an ideal situation, i.e. absence of external forces, it is possible to calculate K as follows.

Initial remarks:

- $p_{adm} - p_{des} \approx 1mm$;
- $M\ diagonal\ elements > 0$;
- $B = 2\xi\sqrt{KM}$;
- $\xi = overshoot = 0.1$;
- $v_{des} = 3\ cm/sec$;
- $a_{adm} = 0\ cm/sec$;
- $F_{ext} = 0\ N$;

Modified admittance equation:

$$0 = \frac{1}{M}\left(-2\xi v_{des}\sqrt{KM} - K(p_{adm} - p_{des})\right) \quad (6.4)$$

Solution:

- $K\ diagonal\ elements = 9000$;
- $B\ diagonal\ elements = 25$;

Computed these parameters, a Simulink model was built in order to verify if the system will remain stable, and if the overshoot is respected.

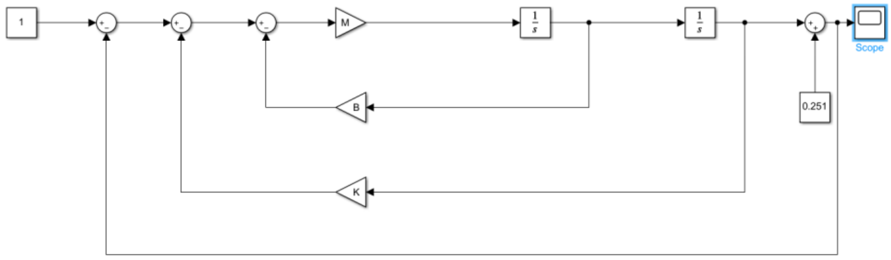

Figure 40 Simulink Mass-Spring-Damper model

From which it is possible to observe that the final position will have 1mm of position error, as set in the equation, and an overshoot less than 0.1.

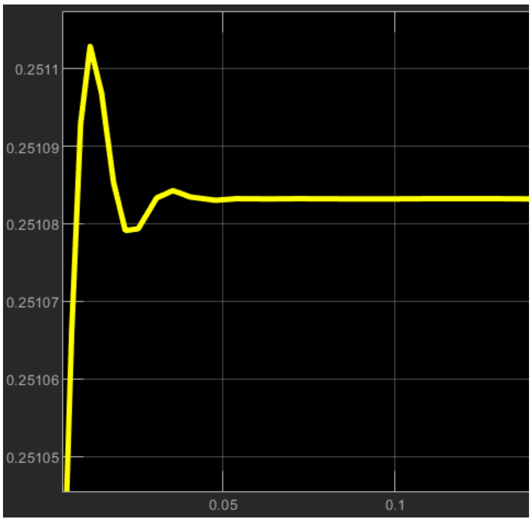

Figure 41 Final Position Result

Analytical analysis allows the parameters of the control system to be calculated without observing the actual behaviour of the robot, which is why the value of K is so high.

From a realistic point of view in order to find the optimal value, that returns a position error lower or equal than 1mm, a series of experiments was done by changing K from 9000 to lower value.

In the previous section it was observed that the relationship between K and the actuator force is similar to a logarithmic function. From this premise it can be assumed that the relationship between the value of K and the size of the position error is also of the same type. It was therefore decided to take K =

3000 as the first real analysis value, which allows the robot to be controlled with almost the same accuracy as that provided by the optimum parameter.

Proceeding in the same way as described above, i.e. with the help of the logarithmic function, it was observed that the minimum value of K that results in the maximum permissible error in position (1mm) is 1500, which is approximately six times lower than the parameter calculated analytically.

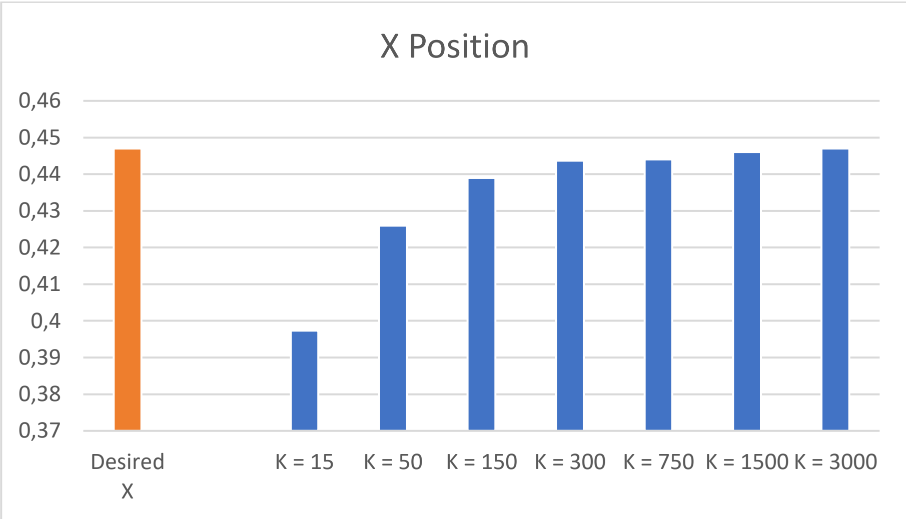

Figure 42 *X-Position by increasing K*

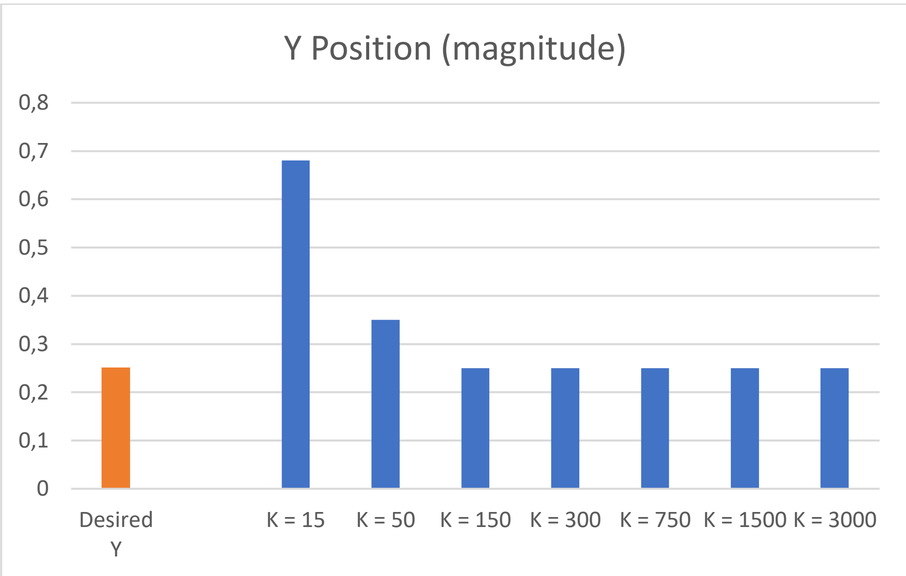

Figure 43 *Y-Position by increasing K*

Figure 44 *Z-Position by increasing K*

6.2.2. Parameter optimization with respect to Injury Criteria

In section 2.4, the maximum forces that can impinge on different parts of a human body were defined. These values are very important if a control system is to be applied in an assistive robot. For this reason, it was decided to optimise the parameters of the admittance control for these conditions as well.

In order to proceed with the experiments, it was necessary to insert an "if" loop in the code, which is able to stop the movement of the arm whenever the force sensor perceives a force greater than 35 N.

In section 6.1.3 a range of values for K was obtained which allow the control system to comply with safety regulations. This range includes all values between 1 and 300. The optimisation of this parameter with respect to the safety criteria was however carried out as in the previous paragraph, i.e. first by an analytical and then a realistic analysis.

Considering the admittance control formula it was possible to observe how, by substituting the value of $F_{ext} = 35\ N$, the equation will be equal to:

$$35 = \frac{1}{M}\left(-2\xi v_{des}\sqrt{KM} - K(p_{adm} - p_{des})\right) \quad (6.5)$$

From which two different values of K will be computed:

$K(p_{adm} - p_{des})^2 + 4\xi^2 v_{des}^2 M = 0;$ (6.6)

$K = -(35M)^2;$ (6.7)

Since K must be higher than 0, the second solution will not be considered. Thus, from the first one the value of K and consequently of B will be:

$K = 36;$

$B = 1.2;$

Ideally, these parameters should allow the robot to move along the desired trajectory, reaching its final position with millimetre precision without generating forces greater than 35 N, but it is possible to observe how parameter B greatly influences the behaviour of the mechanical arm.

As studied in static analyses, a low value of B leads to high oscillations of the robot during its movement. In this case, a simple oscillation with an amplitude greater than 1 millimetre leads to a position error at least equal to the amplitude of the oscillation.

In this case, therefore, it cannot be said that the analytical analyses led to an optimal result. This is the reason why a realistic analysis is necessary.

Leaving the analytically calculated value of the parameter K unchanged, proceeding with the variation of B, it is possible to observe that for B = 25 the robot does not present any type of oscillation, but the final position is shifted by almost 2 cm compared to the desired one.

6.2.3. Move to Position Conclusion

By means of sections 6.2.1 and 6.2.2, the optimal values of B and K with respect to both position and safety criteria were obtained through a long series of experiments and parametric tests.

As it is possible to observe from the following table, in the first case the control system allows the robot to reach the final position with a millimetric error, but with a much higher arm force than the ones allowed by the safety criteria, while in the second case, although respecting the safety values, the final position is not reached, because there is an error of a few centimetres.

Table 10 *Optimization Conclusions*

B	**K**	**Wrench Force**	**Position Error**
25	1500	1500 N	< 1 mm
25	36	36 N	> 1 cm

From the considerations made above and the results shown in the table, the process of concluding the analyses was continued in order to derive the optimal values of the parameters in order to reach the final position with as little error as possible while maintaining the safety conditions established by the relevant regulations.

As can be seen, K equal to 36 results in a position error along the three axes of more than 1 cm. The parameter was therefore changed to K = 150, a value for which the error is between 1 mm and 5 mm, i.e. less than 1 cm, and the force is not greater than 40 N.

7. *Conclusions*

Considering the starting point, remarkable results were achieved through the research. This work has enabled IRI's researchers to structure an initial software library that can be considered as a starting point for future projects that will improve the control system developed until now.

Human-Robot assistance has been improved, not so much from a performance point of view, but with respect to Injury Criteria. In addition to carrying out experiments with the dummy, once optimal results were obtained, the environment allowed to carry out experiments in direct contact with the robot, disregarding the recommended safety distances.

In spite of the excellent performance, it is mandatory to state that being a low-level control system the movements of the Kinova are acceptable for experimental and analytical operations, but not for real application. Many future studies will have to be done, both to improve the smoothness of the handling and to optimise the automation of the robot.

In the last study period were proposed by some researchers what could be considered as future improvements.

7.1. Null-Space Control System

Collaborative robots with 7 DoF have a particular feature in performing certain tasks. As each movement task requires a maximum of 6 degrees of freedom, this type of robot is called redundant in performing the operation.

The concept of redundancy in the world of robotics is certainly a very positive aspect, as the mechanical arm has the possibility to choose the best path to perform a given task.

Redundancy can be used to:

1. Avoid obstacles (in Cartesian space) or kinematic singularities (in joint space)

2. Increase manipulability in certain directions
3. Evenly distribute/limit joint speeds
4. Remaining within the limits of joint end-strokes
5. Minimise energy consumption
6. Optimise travel time, required torques, etc.
7. Increase reliability against failure.

Despite the many advantages of adopting a control system based on the concept of redundancy, there are also disadvantages that increase the complexity in developing such control system. The two main disadvantages are:

1. Structural complexity: the definition of the structure of the robot is more complex from a mechanical and actuation point of view.
2. More complex inverse kinematics and control algorithms: the solutions of an algorithm capable of finding a trajectory are infinite. In addition, there are unknown internal movements.

The implementation of such control system, as can be seen from the disadvantages listed above, requires an in-depth study to define the final algorithm. Once this will be developed, it will be possible to ensure that the robot behaves in an even more compliant manner with the external environment during the operations. For example, if a joint different from the end-effector collides with an external object, this joint, thanks to the Null-Space control system, can stop and allow other joints, that were not previously involved, to continue running.

This means that the entire system is not stopped whenever the robot starts to generate a force greater than the permissible one, but the force of the individual actuator is simply stopped, so that for each stiffness value the Injury Criteria is guaranteed.

7.2. Depth-Camera Control System

Further optimisation can be achieved by using the Kinova's integrated camera. The robot has a Depth-Camera at the end of the end-effector, which sends vision signals as it moves.

The camera can be a great advantage if you want to calibrate the maximum force exerted by the robot for each part of the body. Machine learning algorithms could be used to make the camera recognise each individual body part and modify the K parameter according to which part is closest to the end-effector. This optimisation allows both safer collaborations, since if the robot realises it is positioned too close to an obstacle, it can decide to turn back, and much more detailed collaboration due to the change of K in real time.

Bibliography

- **Alessandro De Luca,** Robot con Ridondanza Cinematica, Roma, 2008
- **Andres L. Jutinico, Jonathan C. Jaimes, Felix M. Escalante, Juan C. Perez-Ibarra, Marco H. Terra and Adriano A. G. Siqueira,** Impedance Control for robotic rehabilitation: A robust Markovian Approach, São Paulo, 2017
- **Arvid QL Keemink, Herman van der Kooij, Arno HA Stienen,** Admittance control for physical human-robot interaction, The International Journal of Robotics Research 2018, Vol. 37(11) 1421–1444
- **Christian Ott, Yoshihiko Nakamura,** Admittance control using a Base Force/Torque Sensor, Tokyo, 2009
- **IRI (Institut de Robòtica i Informàtica Industrial),** Perception and Manipulation Research areas, Barcelona, 2020 (https://www.iri.upc.edu/research/perception#area1)
- **ISO/TS 15066:2016,** Robots and Robotic devices – Collaborative Robots, https://www.iso.org/obp/ui/#iso:std:iso:ts:15066:ed-1:v1:en:en
- **Joe Falco, Jeremy Marvel, Rick Norcross,** Collaborative Robotics: Measuring Blunt Force Impacts on Humans, Gaithersburg, 2012
- **Josep-Arnau Claret i Robert,** The robot null space: new uses for new robotic system, Barcelona, 2019
- **KINOVA**. Kinova Gen3 - User Guide. url: https://www.kinovarobotics.com/
- **LADISPE (Laboratorio didattico sperimentale),** Robotics – Trajectory Planning, Torino, 2013
- **Marina Ramon Borràs,** iMove: Development of a hybrid control interface based on sEMG and movement signals from assistive robotic manipulator, Barcelona, 2020
- **Pablo Sánchez-Sánchez and Fernando Reyes-Cortés,** Cartesian Control for Robot Manipulators, Puebla, México, 2010
- **Ryo Kikuuwe,** Torque-Bounded Admittance Control Realized by a Set-Valued Algebraic Feedback, Piscataway, New Jersey, 2019

- **S. Rao, R. Carloni, and S. Stramigioli,** Stiffness and position control of a prosthetic wrist by means of an EMG interface, 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology. 2010, pp. 495–498. doi: 10.1109/IEMBS.2010.5627153
- **Sonny Tarbouriech, Benjamin Navarro, Philippe Fraisse, André Crosnier, Andrea Cherubini, et al...** Admittance control for collaborative dual-arm manipulation. 2019. ffhal-02307959
- **Suita K., Yamada Y., Tsuchida N., Imai K., Ikeda H., Sugimoto N.,** A Failure-to-safety "Kyozon" system with simple contact detection and stop capabilities for safe human-autonomous robot coexistence. IEEE International Conference on Robotics and Automation 0-7803-1965-6/95. 1995
- **Vincenzo Lippiello, Fabio Ruggiero,** Cartesian Impedance Control of UAV with a Robotic Arm, Napoli, 2012
- **Yamada Suita, Ikeda Sugimoto, Miura Nakamura,** Pain: Evaluation of pain tolerance based on a biomechanical method for human- robot- coexistence. Transactions of the Japan Society of Mechanical Engineers. No 96-0689. 1997-8