FCTUC FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Interaction with Real Environments: an Approach Based on Haptic Systems

Dissertation submitted for the Master degree on Mechanical Engineering on the subject of Production Systems

Author:

Nélio Ricardo Sebastião Mourato

Supervisor:

Joaquim Norberto Cardoso Pires da Silva

Jury

President	**Professor Cristóvão Silva** **Professor at the University of Coimbra**
Members	**Professor Altino de Jesus Roque Loureiro** **Professor at the University of Coimbra** **Professor Joaquim Norberto Cardoso Pires da Silva** **Professor at the University of Coimbra**

Coimbra, September, 2010

Acknowledgments

First of all, my special thanks to my parents for the patience and the economic effort they made to help me reach this point.

I would also like to thank Professor J. Norberto Pires, supervisor of this thesis, for the support and the facilities granted in the use of the robotics laboratory.

I would further like to thank engineer Pedro Neto for his constant availability, motivation and all the knowledge transmitted. Also, a special thanks to my classmate Ricardo Heleno and Dário Pereira for the tips he always gave me along this work.

My work on this thesis has benefited from the support and guidance of several people who work in the Industrial Robotics Laboratory of the University of Coimbra, so I wish to thanks to the engineers Nuno Mendes, Germano Veiga and Ricardo Araújo; and also to my colleagues, Pedro Malaca and Bruno Vasconcelos.

I would like to thank to my brotherhoods of G25, for the time we lived together.

I also wish to thank my former and current colleagues for creating such an easygoing atmosphere over the years and for keeping up that special work atmosphere.

I wish to extend my thanks to all the Professors of the Department of Mechanical Engineering of the University of Coimbra, for the knowledge and guidance that they have passed to me.

And finally, to Coimbra for all the moments and life lessons taught.

"Learning is its own exceeding great reward."

William Hazlitt

Resumo

Pretende-se com este trabalho demonstrar que o processo de interacção e programação de um robô industrial não se restringe somente à elaboração de um programa abstracto que define o *modus operandi* do robô. De facto, programar um robô é ainda um processo moroso e que requer conhecimentos técnicos. Assim, cada vez mais se tem procurado encontrar formas "diferentes" de interagir com robôs, mas com um objectivo bem definido: tornar a interface homem – robô mais intuitiva.

Uma área que tem sido amplamente estudada ao longo dos anos prende-se com a capacidade de um utilizador receber *feedback* de força de um robô virtual ou real, ao mesmo tempo que este executa uma determinada tarefa. Vulgarmente esta interacção homem – máquina é feita através dos denominados dispositivos haptics. Estes dispositivos apresentavam como grande desvantagem o seu elevado custo, no entanto em 2008 foi lançado no mercado um dispositivo haptics concebido para a indústria dos videojogos: o *Novint Falcon*. Este apresenta um preço cerca de 100 vezes inferior relativamente aos seus congéneres até à data existentes.

No âmbito desta tese é desenvolvido e implementado um sistema de controlo que gere todo um sistema robótico composto por um dispositivo *haptics*, um robô industrial e um sensor de força. Desse modo, um qualquer utilizador é capaz de controlar um robô e ao mesmo tempo receber feedback das forças que estão a ser exercidas no *end-effector* do robô.

O sistema foi validado através de vários testes experimentais envolvendo contacto entre o robô e diversos objectos constituídos por materiais das mais diversas características físicas.

Palavras-chave: Robótica, *Haptics*, Teleoperação, Controlo de Força.

Abstract

The aim of this work is to demonstrate that the process of programming and interacting with an industrial robot is not only restricted to the definition of an abstract robot program that defines the robot *modus operandi*. In fact, programming an industrial robot is still a hard process that requires technical expertise. So increasingly, there is a demand for "different" ways to interact with robots, but with a well defined goal: to make the interface human – robot more intuitively.

An area of knowledge that has been widely studied over the last years is related to the ability of a user to receive force feedback from a virtual or real robot whereas it performs a certain task. Commonly, this human – machine interaction is made through a special type of devices called haptic devices. These devices had as its major disadvantage their high cost, however, in 2008 was launched on the market a haptic device designed for the video game industry: the Novint Falcon. This device has a price about 100 times lower than its counterparts in present date.

This thesis presents a control system that manages a robotic platform based on haptic technology. This system is composed by a haptic device, an industrial robot and a Force/Torque sensor. Thus, using this platform any user should be able to control a robot while receiving feedback of forces being exerted on the robot end-effector.

The system was validated by several experimental tests involving contact between the robot and the most varied objects made of materials with very different physical properties.

Keywords Robotics, *Haptics*, Teleoperation, Force Control.

Contents

List of Figures

List of Tables

Symbology and Acronyms

Symbology

a – Acceleration of the mass

$\mathbf{A_m}$ – Vector average acceleration from the master device

$a_{m,x}$, $a_{m,y}$, $a_{m,z}$ – Scalars of the average acceleration vector of the master device corresponding respectively to the axes X,Y and Z

$a_{N,x,}$ – Acceleration on X axis from the master device

b – Root degree

c – Damping Coeficient

F – Force

$f(v_{nov})$ – Function of the velocity from the master device

Fc – Vector Contact Force

$\mathbf{F_{cmd}}$ – Vector contact force for the master device

$\mathbf{F_d}$ – Vector desired Force

$\mathrm{F_{máx}}$ – Scalar of maximum contact force allowed

$\mathbf{F_{vir}}$ – Vector virtual force

k – Spring Constant

$\mathrm{k_{cmd}}$ – Scale factor for the contact force for the master device $\in \mathbb{R}^+$

k_{nov} – Scale factor of the velocity magnitude from the master device $\in \mathbb{R}]\ 0, 1]$

$\mathrm{k_{vir}}$ – Constant elastic modulus of the virtual spring $\in \mathbb{R}^+$

M – Mass of the body

n – Independent variable $\in \mathbb{N}^+$

Tc – Vector Contact Torque

t_n – Time relative to instant n

v_a – velocity magnitude of the displacement for the avatar (robot)

$\mathbf{V_N}$ – Vector velocity from the master device

$\mathbf{V_{N,0}}$ – Vector of the initial velocity from the master device

$v_{N,x}$ – Velocity from the master device on X axis

v_{nov} – Velocity magnitude from the master device

X – Mass displacement

x– Independent variable $\in \mathbb{R}$

$\mathbf{X_a}$ – Vector displacement from the avatar

$\mathbf{X_{a,c}}$ – Vector displacement corrected of the avatar

$\mathbf{X_{ac}}$ – Vector of the actual position from the master device

$\mathbf{X_e}$ – Vector of actual displacement of the robot end-effector

$\mathbf{X_e}$* - Vector actual position of the robot end-effector

$\mathbf{X}_{\mathbf{e},n-1}$ – Vector position of the slave robot in time instant *n-1*

$\mathbf{X_{e,nov}}$ – Vector position of the robot end-effector in the master device coordinate system

$\mathbf{X_N}$ – Vector Position from the master device grip

x_N, y_N, z_N – Scalars of the position vector from the master device, corresponding respectively to the axes X,Y and Z

$\mathbf{X}_{\mathbf{N},0}$ – Vector position from the master device, for the initial instant

$\mathbf{X_{nov}}$ – Vector positional difference from the master device

Acronyms

2D – Two Dimensions

3D – Three Dimensions

ABB – *Asea Brown Boveri*

AOB – Active Observers

ASEA – *Allmänna Svenska Elektriska Aktiebolaget*

CAD –Computed Aided Design

clr – Common Language Runtime

CPU – Central Processing Unit

DEM – *Departamento de Engenharia Mecânica*

DLR - *Deutsches Zentrum für Luft- und Raumfahrt*

DoF – Degrees of Freedom

F/T – Force/Torque

FCTUC – *Faculdade de Ciências e Tecnologia da Universidade de Coimbra*

GUI - Graphical User Interfaces

HDAL – Haptic Device Abstraction Layer

HDI – Human Development Index

IFR – International Federation of Robotics

IP – Internet Protocol

KUKA – *Keller und Knappich Augsburg*

OSI – Open System Interconnection

PC – Personal Computer

PCI - Peripheral Component Interconnect

PUMA – Programmable Universal Machine for Assembly

RT – Real Time

SCARA – Selective Compliant Assembly Robot Arm

SDK – System Developer Kit

UDP – User Datagram Protocol

USB – Universal Serial Bus

USD – United States Dollar

1. INTRODUCTION

In the never-ending effort to "simplify" their existence, human beings are constantly searching for new concepts and technologies that may improve their living conditions. The capability of having machines doing the work of humans is undoubtedly an asset for human development. It is interesting to verify that the countries with better living conditions are those with a higher level of automation in their companies, which is a strategic issue, influencing economic success worldwide. In Figure 1 it is possible to see the estimated number of industrial robots in each country worldwide. Through its analysis it is easily observed that the countries with a better Human Development Index (HDI) are those where the number of robots is higher.

Industrial automation is described as the ability to fabricate a product without or with minor human intervention, allowing companies to produce faster, accurately, with high quality standards and more cost-effectively [Pires, 2007]. The level of automation of a company is the key element for its competitiveness, especially to face companies from regions where the costs of labour are much lower.

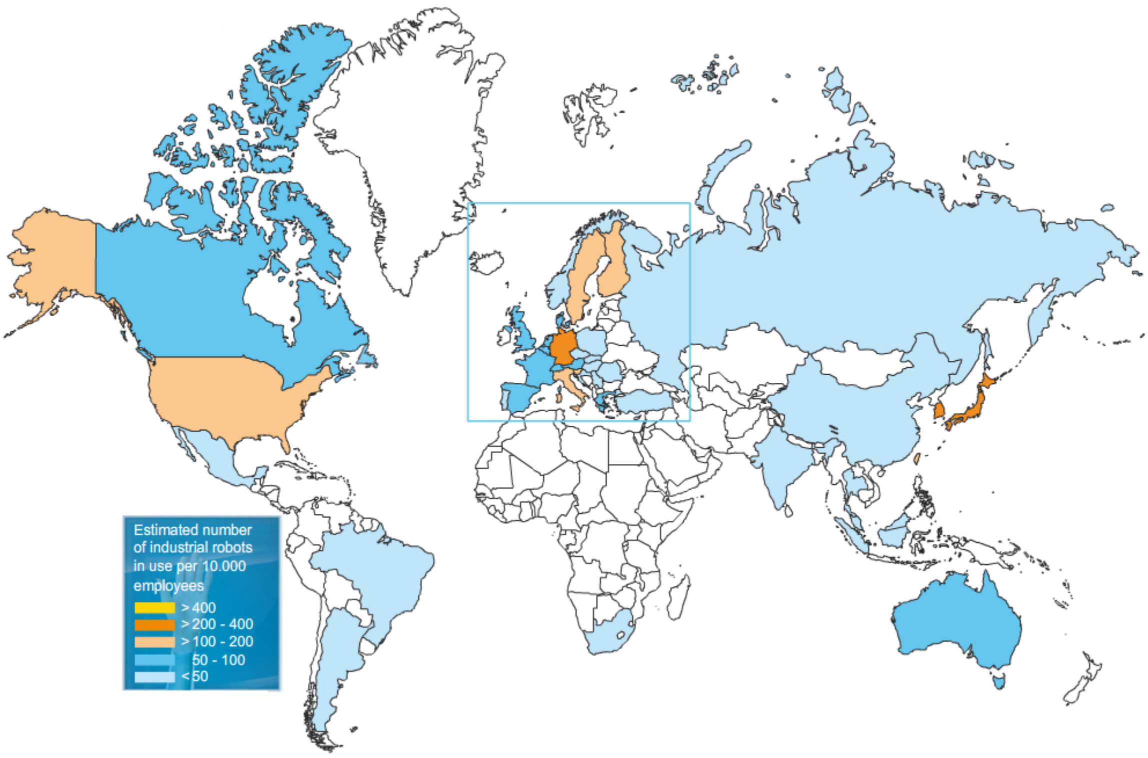


Figure 1 – Estimated number of industrial robots worldwide [IFR, 2009].

As above mentioned, industrial robots play an important role in the world of industrial automation. However, programming an industrial robot by the typical teaching process is a tedious and time-consuming task that requires some technical expertise, and hence new approaches to robot programming are required [Neto *et al.*, 2010].

The purpose of this thesis is to present a robotic system that allows users to instruct, program and receive force feedback from an industrial robot, all of this with a high-level of abstraction from the specific robot programming language. To achieve the thesis goals the potentialities of a recent technology in expansion will be explored; the haptic technology (Section 2.1). Thereunto, a new and low cost haptic device will be explored, the Falcon from *Novint* (usually designated by Novint Falcon) (Section 3.1.1). Until now, the consumer market haptic devices were fairly basic and often labeled “force feedback” controllers. Examples, we may point out the “rumble” on most modern videogames controller and joysticks like the Microsoft Sidewinder. Most of the sophisticated haptic devices are expensive and reserved for professional applications such as medical training.

The appearance of the Novint Falcon has allowed an increase of use and diffusion of haptic technology in many different fields, including industry. Since the beginning of Falcon’s production, haptics technology has been experienced by a larger audience. It is possible to say through a research, that the great advantage of the Falcon is its price. The Falcon has limited control capabilities compared to a more professional device such as *SensableOmni* or *Omega7*. However as a research tool the Falcon is a relatively good device. The developed techniques for the Falcon can easily be implemented in the other haptic devices.

Concerning the robotics field and the work here presented, the Novint Falcon will enable the control of a robot movement at the same time that the user receives force feedback through the Novint Falcon grip (the force values come from a 6 DoF Force/Torque (F/T) sensor attached to the robot wrist). Thus, any user without technical knowledge in the robotics field should be able to control the robot, save robot poses and by this way define step by step a robot program. Moreover, force feedback is a very important issue that allows a user to feel the forces exerted on the robot end-effector. This is the key point to remember: using this system the user can control and program a robot without the need to be a specialist in robotics, on the contrary, the user can focus only on the correct definition of the robotic task (welding, painting, machining, etc.).

The above mentioned system requires work on advanced robotic concepts such as teleoperation, telepresence, force control and robot pose control. All of these areas of knowledge will be subject of study in this dissertation.

Several experimental tests were carried out to verify the reliability and effectiveness of the system. These tests had shown good results and proved that the system is reliable. In a near future this type of robotic platform could be a reality in industrial environments.

1.1. Organization of the Rest of the Thesis

Chapter 2 describes haptic technology, how and where this technology has been used and current applications. Several different types of haptic devices are presented. Moreover, it is also presented a general overview about the evolution of robotics.

Chapter 3 presents an overview of the proposed system, including a description of all equipment used. It is also shown how the equipments communicate with each other.

Chapter 4 gives details about the implemented control methodologies, ballistic control, the virtual spring concept, force control and robot pose control. The architecture of the software application that manages the system is analysed.

Chapter 5 presents the experimental setup, practical tests, results and discussion of results. Four different types of practical tests are performed: free robot movement, robot end-effector contact with a sponge and with a paper box, object manipulation and a final test exploring the system capabilities in a cutting operation.

Chapter 6 concludes the thesis and discusses future directions of research in this area. Future applications are also mentioned.

2. STATE OF THE ART

2.1. Haptics

Haptics is a term derived from the Greek verb "haptesthai" that means "to touch", referring to the act of detection and manipulation through touch. The word haptics is also pointed as the science of the touch, which is devoted to study and to simulate the pressure, texture, vibration and other biological sensations related with the touch [Eid *et al.*, 2007]. In relation to humans, this sense (the touch) possesses two independent components: cutaneous and kinetic. The cutaneous component is linked to the "sensors" located in the surface of the skin, which are responsible for sensations as pressure, temperature, vibration and pain. The kinetic component is connected to the sensors located in the muscles, tendons and joints. This component is related to the sensations of movement and force [Carneiro, 2003].

Through the use of special input/output devices (joysticks, data gloves, haptic devices, etc.) the human beings can receive feedback from computer applications in the form of felt sensations in the hand or other parts of the body. For example, when combined with computer graphics, haptic technology can be used to train people for tasks requiring hand-eye coordination such as surgery or space ship maneuvers [Salisbury *et al.*, 2004].

Haptic technology embraces multiple disciplines such as biomechanics, neuroscience, psychophysics, robot design, robot control, mathematical modelling, simulation and engineering software. This extensive variety of disciplines could instigate a complex bibliographic research, so it is useful to define sub-areas of knowledge in the haptics field: human haptics, machine haptics, computer haptics and multimedia haptics:

(I) **Human haptics** refers to the study of human sensing and manipulation through tactile and kinaesthetic sensations. It comprises human haptic perception, cognition and neurophysiology brought together to contribute to the study of human touch and physical interaction with external environments.

(II) **Machine haptics** involve design, construction and development of mechanical devices that "replace" or augment human touch.

(III) **Computer haptics** is an emerging area of research that concernes the development of algorithms and software to generate and render the "touch" of virtual environment objects – somehow analogous to computer graphics. Essentially, computer haptics deals with modelling and rendering virtual objects for real-time display. It includes the software architecture for haptic interactions and the synchronisation with other display modalities such as audio or visual media.

(IV) **Multimedia haptics** consists in integrating and coordinating the presentation of haptic interface data and other types of media in multimedia applications, allowing the utilization of gesture recognition and receive force feedback.

2.1.1. History

Haptics was introduced at the beginning of the 20th century through research in the field of experimental psychology, aiming the understanding of human touch perception and manipulation.

In order to better understand the importance of haptics, one can draw an analogy between the concept radio-television and video-haptics. In the heyday of radio it was unimaginable that someday the television would come to replace the radio, but in fact that was what happened. Since the sense of touch contains much more information than an image or a video by itself, it is expected that in the future we may have a much more strong presence of haptic systems in our lives. Haptic technology will not appear by itself but associated with video technology and several types of machines [Eid et al., 2007; Salisbury et al., 2004; Gillespie, 2005].

In the 1970s and 1980s, significant research efforts were conducted in the robotics field, including studies in manipulation and perception by touch. These studies were very important since many of them were intended to be part of a long-term goal, the construction of autonomous robots inspired by human abilities. Soon researchers found that the development and construction of these types of machines was much more complex and subtle than their initial hopes had suggested. Meanwhile, terms such as teleoperation, telepresence and telerobotics became common in the robotics community. From those terms, two were especially important to the development of haptic technology, Teleoperation and Telepresence:

(I) **Teleoperation** refers to the extension of a person's sensing and manipulation capabilities to a remote location.

(II) **Telepresence** can be described as a realistic way that an operator can feel physically present at a remote site.

(III) **Telerobotics** is the area of robotics concerned with the control of robots from a distance. It is the combination of Teleoperation and Telepresence.

In the early 1990s, the use of the word haptics in the context of computer was introduced. Much like computer graphics, computer haptics is concerned with the techniques and processes of generating and displaying haptic stimulus to the user in an interactive manner [Srinivasan and Basdogan, 1997]. However, computer haptics uses a display technology through which objects can be physically palpated. This new modality provides information to the user's hand or other parts of the body, by exerting controlled forces through the haptics interface. These forces depend on the mechanical contact and are delivered to the user according to the physical properties of the objects that can be perceived.

Recently, haptic technologies have been integrated with high-end workstations for Computer-Aided Design (CAD) and on home PCs and consoles, expanding the human-computer interaction. Effectively, this implies opening a new mechanical channel between humans and computers so that data can be accessed and manipulated through haptic interfaces.

Nowadays, with the evolution of computers, haptic systems can "display" objects with sophisticated complexity and behaviour. This is possible due to the availability of high-performance force-controllable haptic interfaces and affordable computational modelling tools.

The commercial availability of haptic interfaces, software toolkits and haptics-enabled applications gave this field an experiencing exciting and exponential growth.

2.1.2. Haptic Interaction and Devices

In this section, some of the most relevant haptic interaction techniques, interfaces and devices will be presented, some of them still under development.

Through the physical simulation of a virtual environment it is possible to calculate in real time the contact forces between the various virtual objects. These forces can then be sent to the mechanical actuators of a haptic device (typically motors and vibrators), creating in the user the sensation of being in contact with a real object. Depending on the actuators, haptic devices can be reactive or tactile [Eid *et al*., 2007; Salisbury *et al*., 2004; Anderson *et al*., 2007].

Haptic technology has been implemented through different types of interaction with haptic devices. These interaction types are usually classified into the following different types of touch sensations: force feedback, tactile feedback, proprioception and kinaesthesia.

(I) **Force feedback**: a human can feel the forces applied to its body through the movements of a haptic device; these movements are sensed primarily through musculoskeletal forces and also through the skin that touches the physical interface of the haptic device. This is often accomplished by a user's hand grasping a handle connected within the haptic device, for example, 3D haptic devices (like the Novint Falcon), 2D haptic devices and force-feedback joysticks [Anderson et al., 2007], Figure 2. Nevertheless, these touch sensations can also be achieved through the use of a haptic glove (Figure 2–b), through the vibration of the motors of a haptic device that the user is holding (for example, a game controller or a force-feedback mouse), through the vibration or movement of an object where the user sits or through any other mechanical system that can give adequate sensations of force and/or touch to the user.

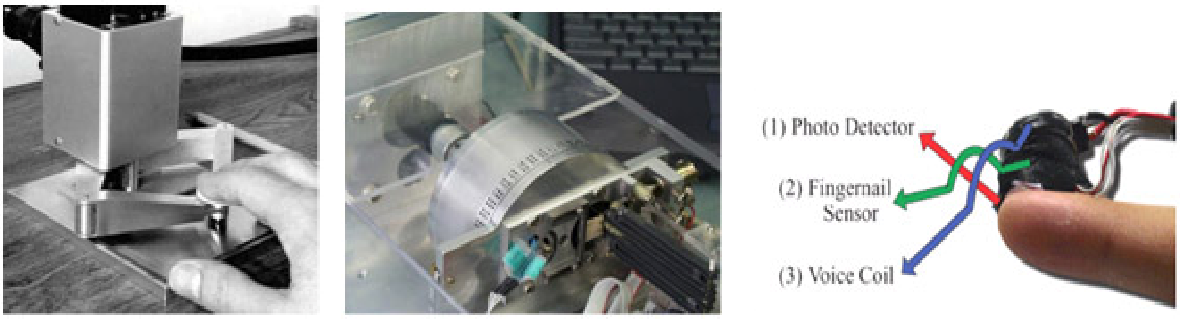
Haptics is often accomplished through the use of electrical motors, although there are other methods and techniques to create force and/or touch sensations:

(i) Pneumatic devices (air controlled).
(ii) Hydraulic devices (fluid controlled).
(iii) Piezoelectric materials (expand or contract with electric current).
(iv) Electric stimulation (sending electric currents directly to a user's skin or nervous system).

Figure 2 – Haptic devices. (a) SideWinder Joystick; (b) DataGlove Cyberglove; (c) WingMan Mice; (d) Omega.7 from Force Dimension; (e) Delta Haptic devices (Force Dimension Delta and Novint Falcon); (f) Phantom from SensAble.

(II) **Tactile feedback**: The forces applied directly on the human skin are detected by "sensors" within the skin called mechanoreceptors. Tactile feedback can also be sensed by a user through electrical currents applied on the skin or using objects that can vary in temperature when touching the skin [Anderson *et al*., 2007].

In general, tactile sensations include pressure, texture, puncture, thermal properties, softness, wetness and friction-induced phenomena such as slip, adhesion and micro failures. Local features of objects such as shape, edges, embossing and recessed features also may cause tactile sensations. For example, tactile feedback can be accomplished with pin arrays on a haptic device that the user places a hand or finger on it [Carneiro, 2003]. The pins within the pin array can slightly raise or lower as the haptic device moves, giving the sensation that the user's finger or hand is moving across a virtual object with texture. These devices are related only with the cutaneous component of the touch. As example of such devices, it can be mentioned the pantograph, Braille displays and the Smart Finger (Figure 3).



(a) (b) (c)

Figure 3 – Tactile feedback devices. (a) Pantograph; (b) Rotary Braille display; (c) Smart Finger.

(III) **Proprioception**: This is the sense that refers to the body's ability within joints and joint position. For example, if you move your arm out to the side, even if your eyes are closed you know where it is. The human sense of proprioception derives from the force that our muscles exert within our body.

Generally, force feedback has a proprioceptive component, as the user's movements on a haptic device in correlation with an application that create the forces he feels. Even computer input devices that are generally not considered haptic devices use our sense of proprioception, such as mice and keyboards [Anderson *et al.*, 2007].

(IV) **Kinaesthesia**: This is similar to proprioception but in this case other internal feelings are included, for example the feeling of a full stomach.

2.1.3. Applications

Haptics research and development has been focused on designing and evaluating several prototypes of different characteristics and capabilities, especially for applications to interact with virtual environments. Applications of haptic technology have been spreading rapidly in several areas: video game industry, multimedia publishing, scientific discovery and visualization, arts and creation, edition of sound and images, vehicle industry, engineering, manufacturing, telerobotics and teleoperations, education and training, as well as medical simulation and rehabilitation [Eid *et al.*, 2007; Salisbury *et al.*, 2004], Figure 4. Recently, some haptic based prototypes have become commercially available.

(a)

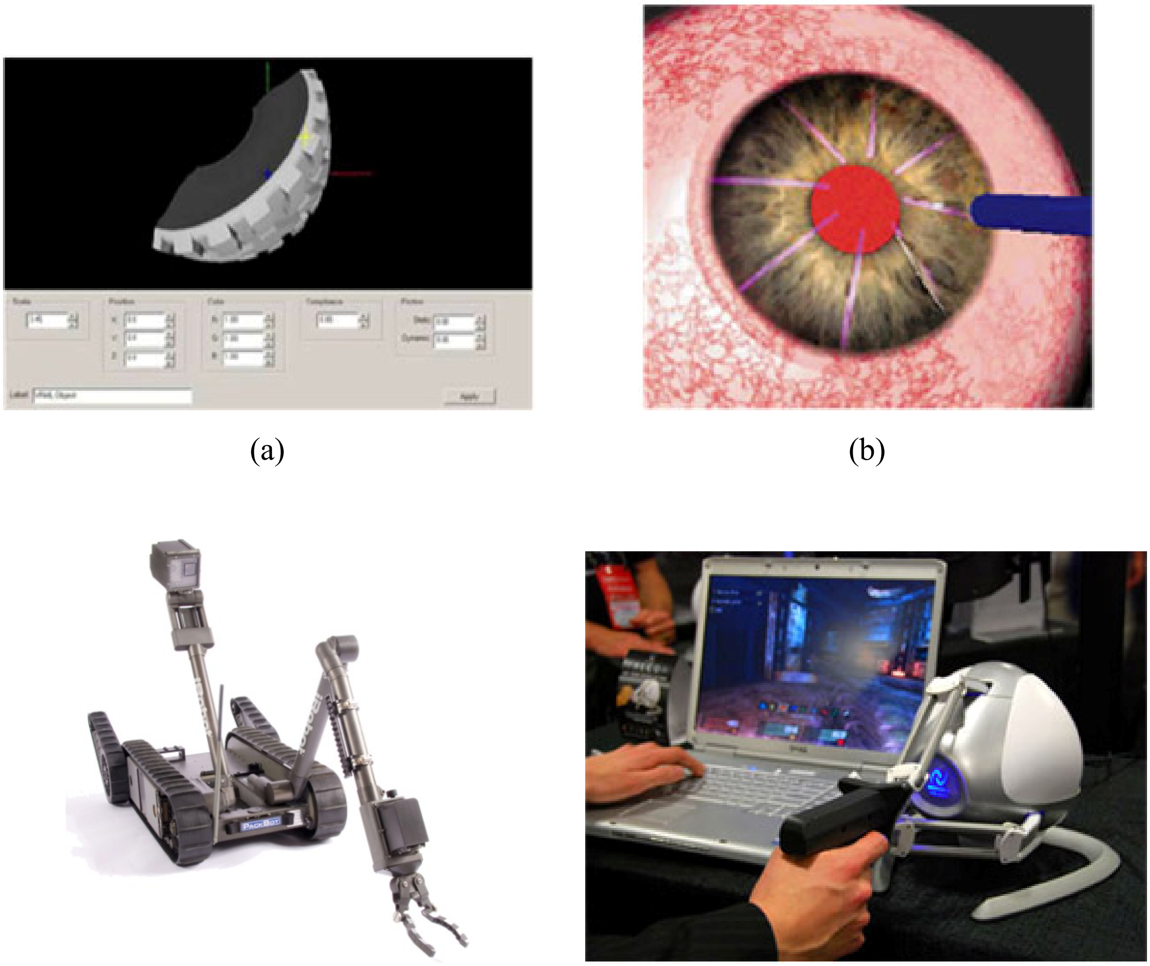
(b)

Figure 4 – (a) Sensable Phanton used for the training of dentists; (b) *DaVinci*, the medical and teleoperated robot.

The haptics application's spectrum is quite vast and its trend of expansion is promising to increase. Nevertheless, haptic interfaces are not yet ready to become a regular device such as computer in today's society. These interfaces confront computational challenges that become considerably demanding, as the realistic experience has to result from the collaboration of three processes: the coordination of the visual system, the position tracking and the update of the forces that actuate the haptic device (these forces can be delivered or simulated).

The Novint Falcon is being applied primarily in the video game industry, but its applications do not stop here. It can also be applied in medical training, as a design and architectural tool, in automotive modelling, tire modelling, programming and control of manipulators, etc. [Anderson *et al.*, 2007; Chotiprayanakul, 2009; Schill *et al.*, 2008; El Far *et al*, 2008], Figure 5. An interesting work presents the use of the Falcon as a force feedback teleoperation device whilst present a mechanical linkage and additional software that allow the Falcon to be used as a pantograph for robot-assisted repetitive motion training [Palsbo *et al.*, 2008; Streng, 2008].

As mentioned, nowadays, haptic technology is applied mainly to interaction with virtual environments (Figure 5-a, 5-b and 5-d), however, exists a development tendency in researching to use this technology for interaction with real environments (Figure 5-c).



(a) (b)

(c) (d)

Figure 5 – (a) Tire modeling; (b) Eye surgery training; (c) Control of military robots; (d) video game industry.

2.2. Industrial Robotics

Robotics is the engineering science and technology of robots, their design, manufacture, application and structural disposition. Robotics is related to electronics, mechanics and software.

The term "robotics" is derived from the word "robot" introduced by the Czech writer Karel Capek in 1920. To put it in a simple way, a robot is a machine capable of independent actions; it means that a robot is a machine which could execute a specific task without human supervision [Hägele *et al*, 2008]. In fact, robots have revolutionized the industrial workplace and today thousands of manufacturers rely on the productivity, high-performance and savings provided by modern-day industrial automation where the robot is a key element.

The history of industrial robotics begins in 1954 with the invention of the first programmable robot by George Devol (considered the father of robotics). In 1956 Devol and Joseph F. Engelberger created the first robot company, the UNIMATION [Hunt, 1983; Nof, 1999]. The first industrial robot was online in a General Motors automobile factory in 1961, New Jersey. It was a Devol and Engelberger's UNIMATE (Figure 6-a).

In 1969 Victor Scheinman at Stanford University invented the Stanford arm (Figure 6-b), which was the first electrically powered computer-controlled robot arm with 6-axis articulated.

(a) (b)

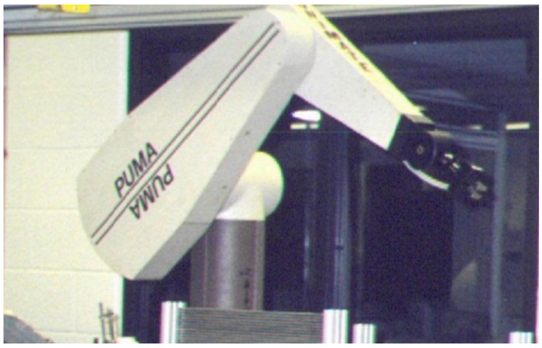
Figure 6 – (a) Unimate robot; (b) Stanford arm.

In 1973, the Swedish company ASEA (now ABB) introduced the first microcomputer-controlled all-electric industrial robot, the IRB-6 (Figure 7-a), which allowed continuous path motion, a precondition for arc-welding or machining robotic operations [Hägele *et al*, 2008].

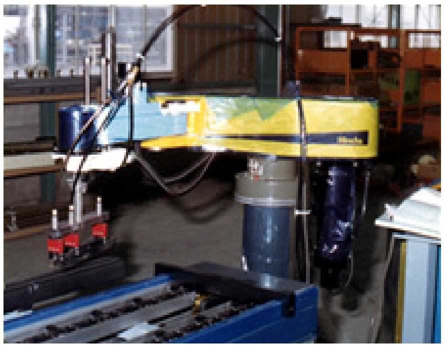
In 1978, the selective compliance assembly robot arm (SCARA) (Figure 7-c) was invented by Hiroshi Makino of Yamanashi University, in Japan. The ground-breaking four-axis low-cost design was perfectly suited for small parts assembly as the kinematic configuration allows fast and compliant arm motions.

The 6 axis robot PUMA (programmable universal machine for assembly) came close to the dexterity of a human arm (Figure 7-b). After its launch in 1979 by UNIMATION it became one of the most popular arms and was used as a reference in robotics research for many years.

(a)



(b)



(c)

Figure 7 – (a) ASEA IRB-6; (b) Puma robot arm; (c) Hirata Scara Robot.

In modern days, two handled dexterous manipulation can be critical for complex assembly tasks, simultaneous handling, processing of work pieces and the handling of large objects. The first commercial robot for synchronized, two-handed manipulation was introduced by MOTOMAN in 2005 (Figure 8-a). This dual-arm robot imitates the reach and dexterity of human arms and can be put on a place that previously accommodated human workers, reducing labour costs. It features 13 axes of motion, six axes per arm plus a single axis for the base rotation.

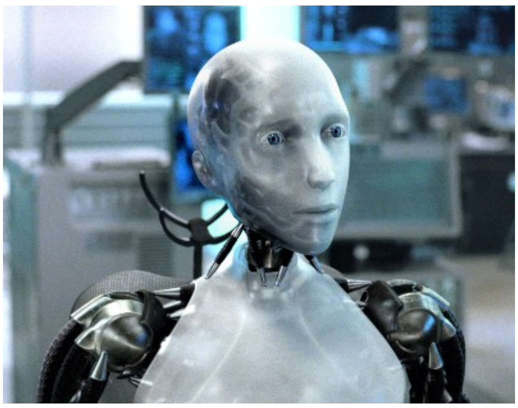
In 2006 the German company KUKA Robotics release a compact 7 DoF lightweight robot arm with advanced force control capabilities (Figure 8-b). The joint speed and weight of this robot have led to a new kinematic and transmission design. The 7 axis arm which is suited for human-robot cooperation imitates the dexterity of a human arm.

(a)

(b)

Figure 8 – (a) MOTOMAN DA20; (b) KUKA Light-Weight robot arm (7 DoF).

It is expected that in the future the robots we commonly see in science fiction can be a reality (Figure 9-a). This can be evidenced by the humanoid robot Justin, equipped with two lightweight arms and two four-finger hands (Figure 9-b). In future, humanoid robots are envisioned in household applications as well as in space environments [Borst, 2008]



(a)

(b)

Figure 9 – (a) Robot model NS-5 from the science-fiction film "I Robot"; (b) A futuristic-real robot Justin made by DLR and KUKA.

3. SYSTEM ARCHITECTURE AND EQUIPMENTS

All the equipments belonging to the robotic platform, developed by the author, are presented in this chapter with an adequate level of detail. Moreover, the way the equipments interact with each other will also be analysed.

The interaction with the robot is carried out through a haptic device, the Novint Falcon, allowing to teleoperate and monitoring the forces being exerted on the robot end-effector. The position data sent by the haptic device are received in a computer running a software application that manages the entire platform. After being analysed and treated these data will be used to control the robot (teleoperation). Simultaneously, force and torque data sent by the F/T sensor is received by the application interface, treated and then sent the appropriate force commands to the haptic device (Figure 10).

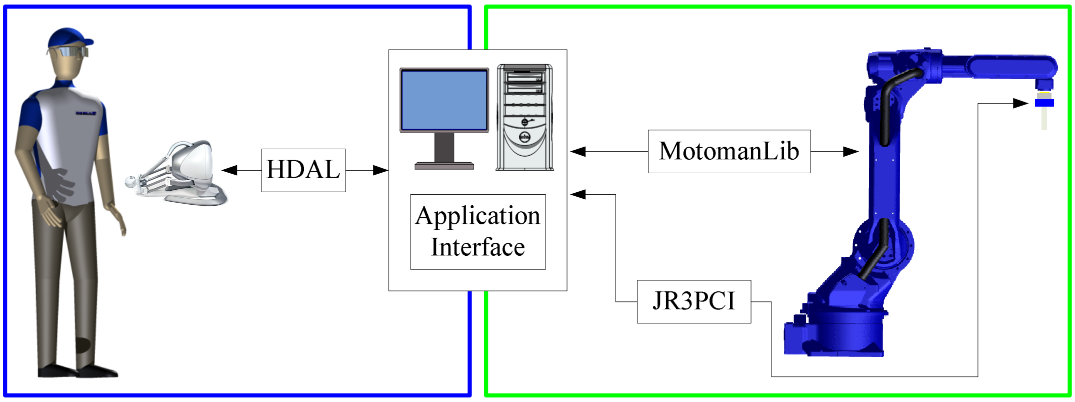


Figure 10 – System Architecture.

3.1. Equipments

3.1.1. Novint Falcon

Novint Falcon is a relatively inexpensive 3 DoF haptic device, which has a configuration similar to a delta-robot [Clavel, 1989] (Figure 11). This specific type of robot (delta configuration) has proven itself to be an excellent platform for high-speed pick-and-place operations due to the mechanism's low actuated inertia, high power to weight ratio, high stiffness and high payload capability when compared to serial counterparts [Olsson, 2009].

Novint Falcon was the first haptic device that made high-fidelity three-dimensional interactive touch possible and practical for consumer computing applications at a low cost. It is essentially a small robot that lets users feel weight, shape, texture, dimension, dynamics and force effects when playing games. Using the Falcon, users experience a full range of realistic touch sensations that allows them to interact with an environment (real or virtual) more naturally and intuitively.

This device was initially used as a computer game controller however its application was rapidly expanded to other areas. The Falcon provides highly accurate tracking and extremely realistic force feedback sensation, allowing a user to interact and learn muscle memory, as much as games and sports are played in real life [Anderson *et al*, 2007].

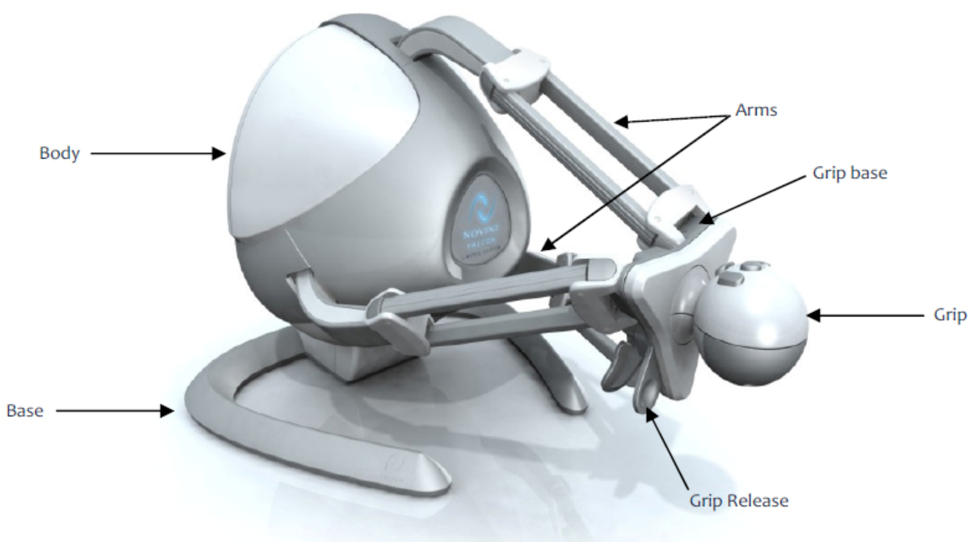


Figure 11 – Novint Falcon.

The product name comes from one of nature's best flying predators, the falcon. Because like the falcon, this controller can move freely in the three dimensional space, providing a realistic 3D sense of touch and an immersive experience that surpasses existing point and click technology. So, it is natural that Novint Falcon has been described as the predator of the computer mouse.

The Novint Falcon was launched in 2007 and pioneered a new category of touch products for the consumer market. In the past, the cost of 3D-touch haptic hardware made the technology impractical for consumer applications (from tens of thousands to hundreds of thousands of USD). The Novint Falcon costs less than 200 USD (Table 1) and can be compared with other similar devices [Inition, 2010].

Table 1 – Comparison of Novint Falcon with other haptic devices and game controllers; RL – Right-Left, FB – Front-Back, UD – Up-Down.

	Novint Falcon	Omega	Delta	SensAble Phanton	Nintendo Wii remote	PS3
Degrees of Freedom	3 + 3 with special handle	3 + 3 with special handle	3 + 3 with special handle	6	6	3
Movements Directions	RL FB UD With special handle: Roll Pitch Yaw	RL FB UD With special handle: Roll Pitch Yaw	RL FB UD With special handle: Roll Pitch Yaw	RL FB UD Roll Pitch Yaw	RL FB UD Roll Pitch Yaw	Roll Pitch Yaw
Rumble/Vibration Force Feedback	Yes	Yes	Yes	Yes	Yes	No
3D shape exploration	Yes	Yes	Yes	Yes	No	No
High-fidelity Force Feedback, Textures, and Interactions at a consumer price point	Yes	No	No	No	No	No
Real Dynamics Modeling	Yes	Yes	Yes	Yes	No	No
Workspace [mm]	100 x 100 x 100	160 x 160 x 120	360 x 360 x 300	160 x 120 x 70	-	-
Forces Máx [N]	8,9	12	20	3,3	-	-
Price USD	<200$	33000$ to 53000$	40000$ to 60000$	2000$ to 10000$	50$	>200$

3.1.1.1. Interface

The Falcon uses a USB interface to send and receive data to/from a computer application. Received data is interpreted by an onboard firmware and sensory data from encoders is transmitted back to the controlling computer. Novint has released a Windows SDK to make easier the process of interaction with the device.

This USB interface uses a 1kHz update rate with commanded forces maintained by the firmware for 10Hz unless overwritten [Martin and Hillier, 2009]. Some authors refer that the haptics force rates must exceed 1kHz to obtain acceptable results [Choi and Tan, 2004]. It was found that a 1kHz update rate was unable to be sustained over the USB interface and typically missed commands or reads resulted in a real-world communication rate between 800Hz and 1kHz, depending on the controlling computer's load. This interface also resulted in a noticeable (2-5 samples) delay between force commands being issued and changes in the encoder measurements being received by the controlling computer. The exact cause of this delay is still unknown. It is possible that some of these delays may be eliminated with custom firmware on the Falcon's internal controller chip [Martin and Hillier, 2009].

3.1.1.2. Workspace

The most common criticism of the delta-robot configuration is its limited workspace. The three limbs of the Falcon work in kinematic concert to actuate the end-effector, but each leg is limited by the reach of the connected linkages. This results in a workspace bounded by warped tri-hemispherical regions overlapping along the longitudinal Z axis. To quantify this volume several tests have been done. In such way, a number of random points were generated in cartesian space and tested to see if they were kinematically realizable. As the numbers of random points were increased, so a better estimate of the workspace volume was formed.

Similarly, by simple application of limit theory, it is also possible to estimate the enclosing volume, approximately $7.90 \times 10^{-5} m^3$. Figure 12 shows the resultant calculated plots of the workspace [Martin and Hillier, 2009].

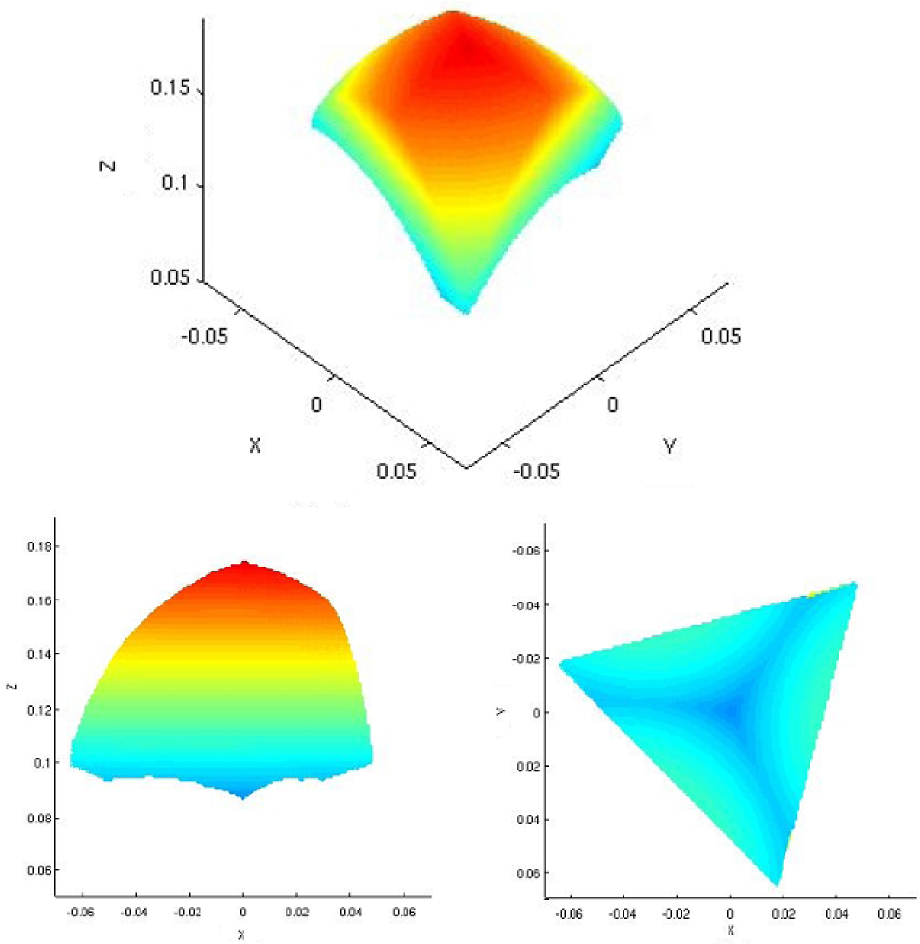


Figure 12 – Novint Falcon's achievable workspace. The units are in meters [Martin and Hillier, 2009].

3.1.1.3. Force Models

The manufacturer of Falcon haptic devices provides some general specifications on workspace size and forces that the device is able to support, but they do not present detailed values. Concluded that the torques required to produce a certain force increases as the actuators are near to the end of their stroke. The largest forces can be produced in the centre of the workspace [Martin and Hillier, 2009].

Haptics calculation is mainly based on applied physics. There are only two physics relationships needed to provide a huge percentage of haptic effects, mass-acceleration model (1) and basic damped spring (2).

$$F = M \times a \tag{1}$$

$$F = k \times X + c \times V \tag{2}$$

Where F is the force, M the mass, a the acceleration, k the spring constant, X the mass displacement, c the damping coefficient and V the mass velocity.

The mass-acceleration model can be used for example when we are shoting a basketball, in this case the primary force calculation is based on the mass of the basketball and the force required to produce the desired acceleration. The other model, spring-mass-damper (Figure 13), finds application in very easily visualized examples, such as a ball bouncing at the end of a rubber band.

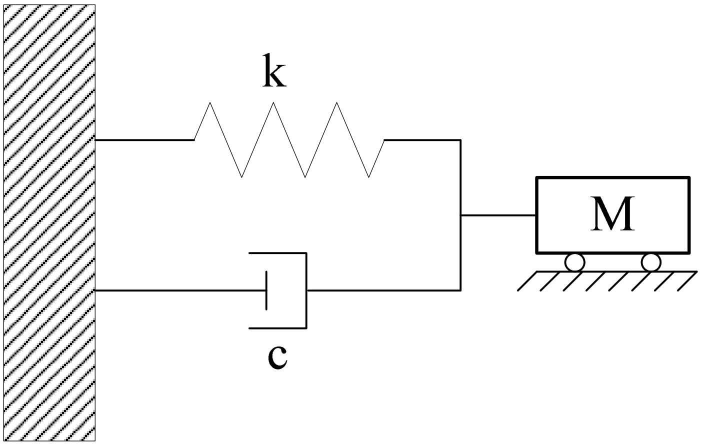


Figure 13 – Basic spring-mass-damper model.

3.1.2. Robot Manipulator

The selected robotic arm is a MOTOMAN HP6 (Figure 14) equipped with an NX100 controller also from MOTOMAN. This is a high speed robotic arm that presents high reliability and repeatability. This robot has been applied in different industrial applications, such as material handling and packaging. The characteristics of the robot are given in Table 2.

Table 2 – Robot Characteristics.

MOTOMAN HP6	
Controlled Axes	6
Robot Mass	130 kg
Payload	6 kg
Vertical Reach	2403 mm
Horizontal Reach	1378 mm
Repeatability	±0,08 mm

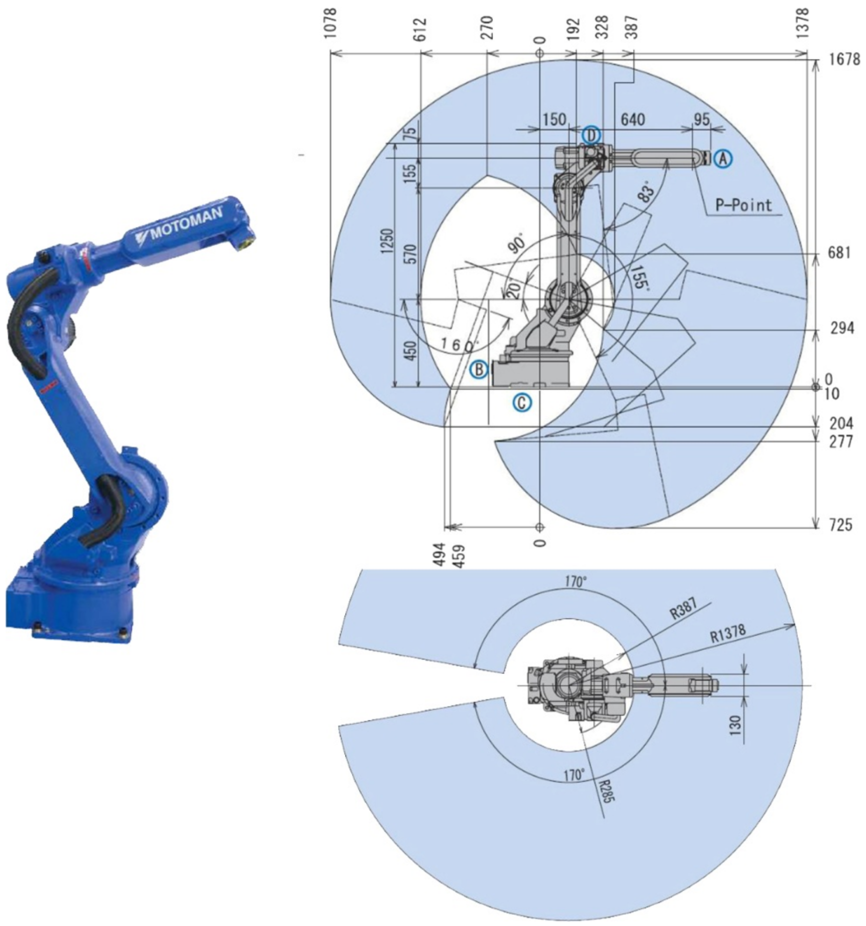


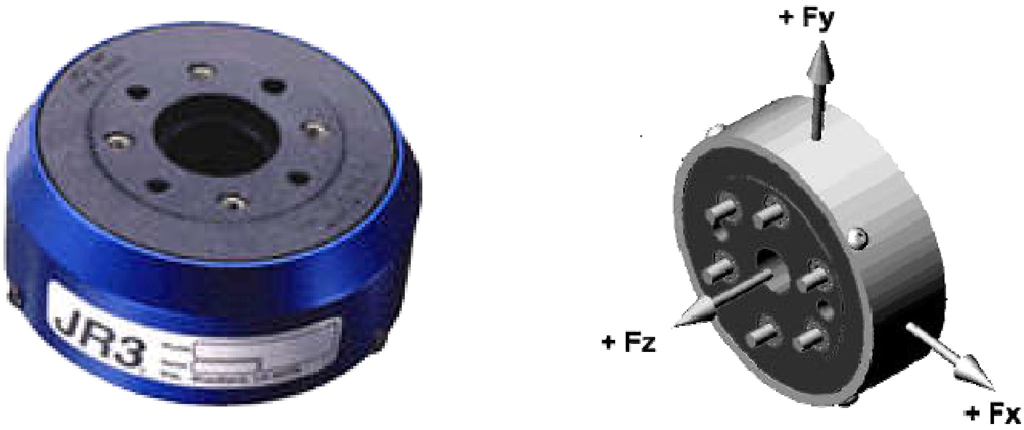
Figure 14 – Reaching volume of *MOTOMAN HP6*.

3.1.3. Force Sensor

The selected F/T sensor, 85M35A from JR3 (Figure 15-a), is instrumented with metal foil strain gages which sense the loads imposed on the sensor. The strain gage signals are amplified and combined to become an analogue representation of the force and torque loads along the three axes of the sensor. In most models, the analogue data is converted to digital format by electronic systems contained within the sensor [*JR3* Manual's].

The coordinate system on standard JR3 sensors is oriented with the X and Y axes in the plane of the sensor body and the Z axis perpendicular to X and Y (Figure 15-b). The reference point for all loading data is the geometric centre of the sensor.

The selected sensor is 85mm in diameter and 35mm in thickness, and load rating 63N [*JR3* Datasheets].



(a) (b)

Figure 15 – (a) F/T sensor JR3 6 DoF; (b) Sensor coordinate system.

3.1.4. Computer

The computer used in this work served both to develop the interface application that manages the whole system and to run that application. The main features of the computer are listed in Table 3.

Table 3 – Main features of the computer.

Processor	Intel® Core™2 Duo CPU E8400 @ 3,00 GHz
Memory	1,75 GB of RAM
System	Microsoft Windows XP Professional

3.2. Data Transfer

One of the main tasks of this work was to put all the hardware to communicate with each other. The way the various components of the system communicate and interact with each other is represented in a simplified scheme in Figure 16.

It was developed an application interface which controls/manages the whole system, in other words, the application acquires data, interprets it (robot pose control, force control and haptic interface) and sends control commands to the robot and Novint Falcon. This application interface was implemented on the development platform *Visualstudio* from *Microsoft* and coded in C++ [Soulié, 2007].

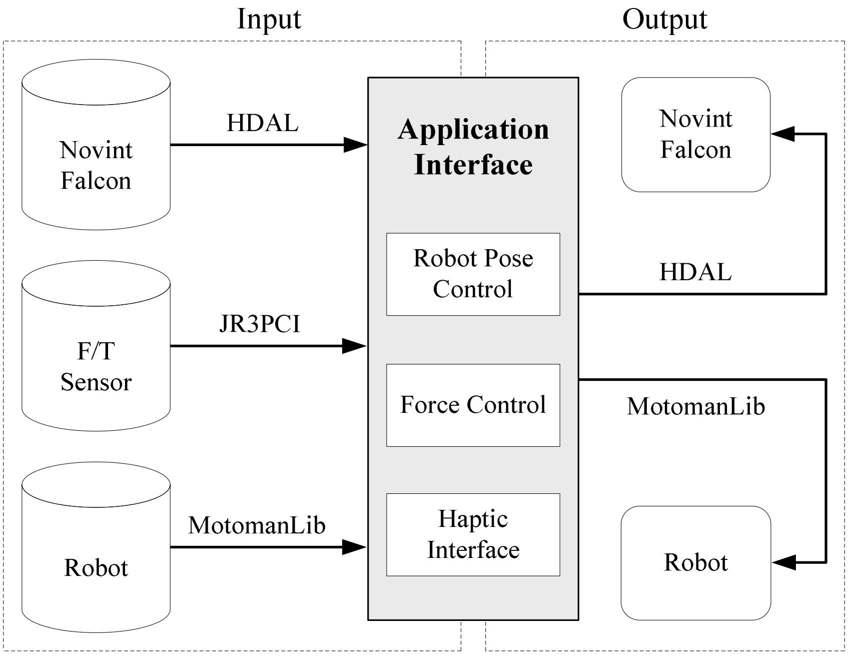


Figure 16 – System data transfer.

In order to acquire motion data from the Novint Falcon it was used a free tool named HDAL SDK [Novint Technologies Incorporated, 2008]. This motion data represents the position vector of the Novint Falcon's grip, $\mathbf{X_N} = (x_N, y_N, z_N)$.

The data from the F/T sensor is acquired using an ActiveX named JR3PCI, these data include forces, $\mathbf{F_c} = (f_{cx}, f_{cy}, f_{cz})$, and torques, $\mathbf{T_c} = (t_{cx}, t_{cy}, t_{cz})$.

Adding the MotomanLib component to the application interface it is possible to control and monitor the robot, in other words, it will enable the application to receive information about the pose of the robot end-effector $\mathbf{X_e}^* = (x_e^*, y_e^*, z_e^*, rx_e^*, ry_e^*, rz_e^*)$, and send pose commands to the robot $\mathbf{X_a} = (x_a, y_a, z_a, rx_a, ry_a, rz_a)$.

Finally, the Falcon receives force commands, $\mathbf{F_d} = (f_{dx}, f_{dy}, f_{dz})$, from the application interface through the HDAL SDK.

4. TELEOPERATION APPROACH BASED ON HAPTICS

Haptic teleoperation provides telepresence since it allows a user to remotely operate a slave robot through a master device and at the same time “fell” the remote environment. The potential of this type of system is enormous, but the connection of master/slave stations in a coherent way is still a challenging task. Whereas the master station is controlled by a human operator, the slave station often interacts with an unknown and dynamic environment. The nature of this interaction greatly influences overall system performance [Park and Khatib, 2006; Chotiprayanakul *et al*, 2008].

In this study, the teleoperation approach is realized by integrating three different components: a virtual spring to connect the master and the slave, positional robot control (ballistic approach) and force control (robot reaction and Falcon force feedback).

To a better understand of the applied concepts it will be described how to acquire force data on the robot wrist. Then, with this data it is possible to limit the movement of Falcon and therefore the movement of the robot.

4.1. Robot Position Control (Ballistic Control)

Concerning to haptics, position control is one of the most common control paradigms. Usually, it refers to a mapping in which the displacement of an object in physical space directly dictates displacement of another different object. In our specific study, the displacement of the Falcon grip will produce a displacement on the robot end-effector. It is important to note that the workspace of a haptic device is normally different from the workspace of a robot. The volume ratio between a haptic workspace and a robotic arm workspace is usually less than one tenth [Chotiprayanakul, 2009].

In this study the robot accuracy depends on the robot motion which in turn is controlled by the speed with which the user’s hand moves the Falcon grip (ballistic approach). Generally speaking, the ballistic approach makes the mapping between the displacement produced on the haptic device and the resulting robot motion. Ballistics refers to the technique of varying the scaling between the motion of a physical device (the Falcon) and the motion of a displayed avatar (in this case a real robot) depending upon the velocity of the device in its workspace [Conti and Khatib, 2005]. Making an analogy, one can consider that the real robot end-effector behaves as an avatar. The ballistic approach provides an accurate control of the avatar, videlicet, performing the same hand trajectory the user could control the avatar with a large or small movement (depend on the speed of the hand movement). More specifically, if the user wants the avatar travel a great distance he needs to perform a rapid movement (moving the grip of the Falcon), on the contrary, if the user is performing small movements the avatar will travel a short distance (Figure 17). This last situation can be useful when it is necessary to control the robot with precision.

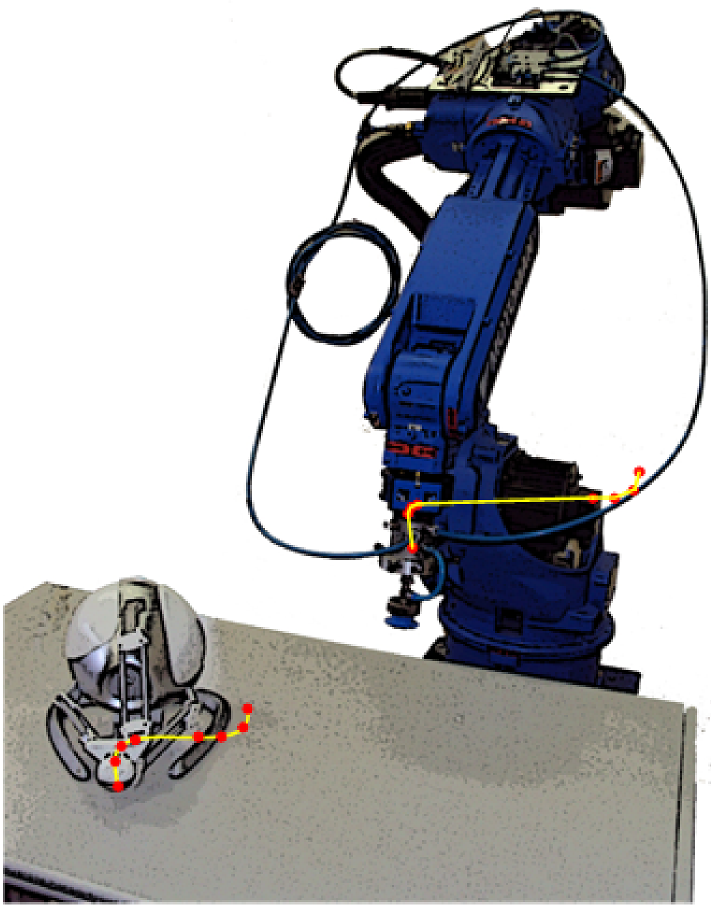


Figure 17 – Ballistic control; Novint Falcon's grip large motions are translated into larger displacements on the slave robot end-effector.

4.1.1. Ballistic Control I

It is necessary to achieve the motion to send to the robot: displacement $\mathbf{X_a}$=(x_a,y_a,z_a,0,0,0) and velocity v_a. These data come from the motion produced by the user on the Falcon's grip. Thus, $\mathbf{X_a}$ can be calculated as:

$$\mathbf{X_a} = \mathbf{V_{N,0}} \times \Delta t + \mathbf{A_m} \cdot \frac{\Delta t^2}{2} \tag{3}$$

Where $\mathbf{V_{N,0}}=(v_{N,x,0}, v_{N,y,0}, v_{N,z,0},0,0,0)$ is the initial velocity which the Falcon is moved along the three Cartesian axes, $\Delta t=t_n-t_0$ is the time variation from the beginning of the Falcon movement to the end of the movement (time instant n); and $\mathbf{A_m}=(a_{m,x}, a_{m,y}, a_{m,z},0,0,0)$ is the average acceleration with which the Falcon is moved. So, to calculate $\mathbf{X_a}$**,** first it is necessary to calculate $\mathbf{V_0}$ and $\mathbf{A_m}$. In order to simplify the explanation of the process it will be analyzed only motion along the X axis. For Y and Z axes the process is similar. The system is constantly acquiring motion data from the Falcon, an example is presented in Figure 18, where four different points (along X axis) acquiring at four different time interval are represented.

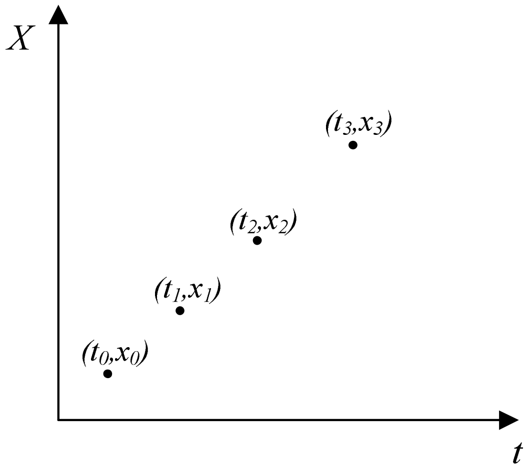


Figure 18 – Position points acquired along the X axis from the Falcon.

From the acquired points it is possible to calculate velocity and acceleration of Falcon's grip. In basic physics velocity is the change rate of position in an interval of time, $v=dx/dt$. In the same way, acceleration is the change of velocity over time, $a=dv/dt$. Figure 19 explains how to calculate velocity and acceleration from the motion data acquired from the Falcon.

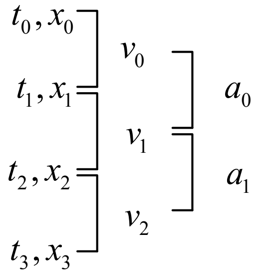


Figure 19 – Calculating velocity and acceleration from motion data.

The velocity along the X axis for a generic interval of time *n* can be calculated as:

$$v_{N,x,n-1} = \frac{x_{N,n} - x_{N,n-1}}{t_n - t_{n-1}} \tag{4}$$

In the same line of thought and from (4), acceleration can be calculated as:

$$a_{N,x,n-2} = \frac{v_{N,x,n-1} - v_{N,x,n-2}}{t_n - t_{n-2}} \tag{5}$$

From (5), $\mathbf{A_m}$=($a_{m,x}$, $a_{m,y}$, $a_{m,z}$,0,0,0)where the average acceleration along the X axis can be calculated as:

$$a_{m,x} = \overline{a_{N,x,n-2}} = \frac{1}{n-1}\sum_{i=0}^{n-2} a_{N,x,i} \tag{6}$$

The above mentioned method was implemented and tested (Table 4). It was done six different tests, extracting different number of position data from the Falcon. It is important to refer that each test was executed with different motions, calculating the normal displacement of the Falcon grip and the displacement using the ballistic approach.

Table 4 – Experimental tests for Falcon grip displacement calculation, normal displacement and displacement calculation using the ballistic approach.

BALLISTIC APPROACH I

Test 1 - 5 position elements

x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
-2,46	-38,10	-12,47	1765
2,46	-33,81	-7,62	1875
9,14	-27,64	-2,26	1968
14,81	-22,00	1,37	2078
16,1036	-15,265	2,1336	2171

Normal Displacement ($X_n(4)– X_n(0)$)		
Δx_N [mm]	Δy_N [mm]	Δz_N [mm]
18,57	22,83	14,61

Ballistic control		
x_a [mm]	y_a [mm]	z_a [mm]
14,01	20,36	13,05

Test 2 - 5 position elements

x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
-6,50	-37,54	27,61	6703
12,70	-33,32	28,55	6812
29,90	-30,58	24,59	6921
37,54	-23,32	20,04	7015
40,79	-12,19	16,10	7125

Normal Displacement ($X_n(4)– X_n(0)$)		
Δx_N [mm]	Δy_N [mm]	Δz_N [mm]
47,29	25,35	-11,51

Ballistic control		
x_a [mm]	y_a [mm]	z_a [mm]
53,13	25,57	-2,41

Test 3 - 4 position elements

x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
-35,00	2,90	3,61	3093
-36,37	-12,12	6,83	3203
-27,76	-31,09	12,57	3312
-11,48	-38,30	22,10	3406

Normal Displacement ($X_n(3)– X_n(0)$)		
Δx_N [mm]	Δy_N [mm]	Δz_N [mm]
23,52	-41,20	18,49

Ballistic control		
x_a [mm]	y_a [mm]	z_a [mm]
17,69	-35,18	17,66

Test 4 - 4 position elements

x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
-47,02	7,52	-7,87	17078
-45,64	-1,09	-2,97	17171
-41,61	-15,04	2,51	17281
-37,69	-26,37	5,18	17390

Normal Displacement ($X_n(3)– X_n(0)$)		
Δx_N [mm]	Δy_N [mm]	Δz_N [mm]
9,32	-33,88	13,06

Ballistic control		
x_a [mm]	y_a [mm]	z_a [mm]
7,14	-30,45	13,28

Test 5 - 3 position elements

x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
0,51	29,16	-15,98	13968
-9,14	25,25	-16,00	14078
-20,07	18,72	-12,04	14171

Normal Displacement ($X_n(2)– X_n(0)$)		
Δx_N [mm]	Δy_N [mm]	Δz_N [mm]
-20,57	-10,44	3,94

Ballistic control		
x_a [mm]	y_a [mm]	z_a [mm]
-20,83	-10,73	4,30

Test 6 - 3 position elements

x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
28,63	7,72	-16,61	10109
27,18	18,03	-24,69	10218
20,96	28,45	-31,80	10312

Normal Displacement ($X_n(2)– X_n(0)$)		
Δx_N [mm]	Δy_N [mm]	Δz_N [mm]
-7,67	20,73	-15,19

Ballistic control		
x_a [mm]	y_a [mm]	z_a [mm]
-8,07	20,85	-15,20

Throughout the tests, it was observed that the ballistic approach complies with the planed, but its results were not commensurate with the expected. In reality the obtained values do not give a concrete result, for example, in test number four, when we have a medium variation of displacement along the Y axis the displacement calculated by the ballistic control decreases and for a small variation along Z axis its value augments. The more acceptable results were obtained from the tests when was extracted five position measurements but even so those values diverge from the desired.

Finally, it is necessary to calculate the velocity magnitude for the robot displacement v_a, from (3):

$$v_a = \sqrt{\frac{x_a^2 + y_a^2 + z_a^2}{(t_n - t_0)^2}} \tag{7}$$

This method unfortunately does not give the expected results, but it has much potential, for example to differentiate the movements of the controller, videlicet, with a good algorithm it is possible by a similar method to send the information to the robot about the motion executed by the user. In these cases the robot will know if it has to execute a straight line or a curvilinear motion.

4.1.2. Ballistic Control II

Since the ballistic approach presented in previous section does not give the expected results it is necessary to implement a new approach that can provide the desired position variation. To achieve the purpose of this dissertation it was created a new method (Figure 20) where $\mathbf{X_a}$ depends on the velocity magnitude from the master device v_{nov} and the positional difference in the Falcon grip $\mathbf{X_{nov}}$ in two consecutive time interval, t_n and t_{n-1} is given by:

$$\mathbf{X_{nov}} = \mathbf{X}_{\mathbf{N},n} - \mathbf{X}_{\mathbf{N},n\text{-}1} \tag{8}$$

This means that for the same position variation of the master device the movement of the slave robot will directly change with the master velocity v_{nov}; if it increases the displacement for the slave robot will also increase, and vice versa.

$$\mathbf{X_a} = f(v_{nov}) \times \mathbf{X_{nov}} \tag{9}$$

From equations (8) and (9):

$$\mathbf{X_a} = f(v_{nov}) \times \left(\mathbf{X}_{\mathrm{N},n} - \mathbf{X}_{\mathrm{N},n-1}\right) \tag{10}$$

The approximate scalar velocity magnitude of the master device can be calculated as:

$$v_{nov} = \left\|\mathbf{V}_{\mathrm{N},n-1}\right\| = \frac{\sqrt{(x_{N,n} - x_{N,n-1})^2 + (y_{N,n} - y_{N,n-1})^2 + (z_{N,n} - z_{N,n-1})^2}}{t_n - t_{n-1}} \tag{11}$$

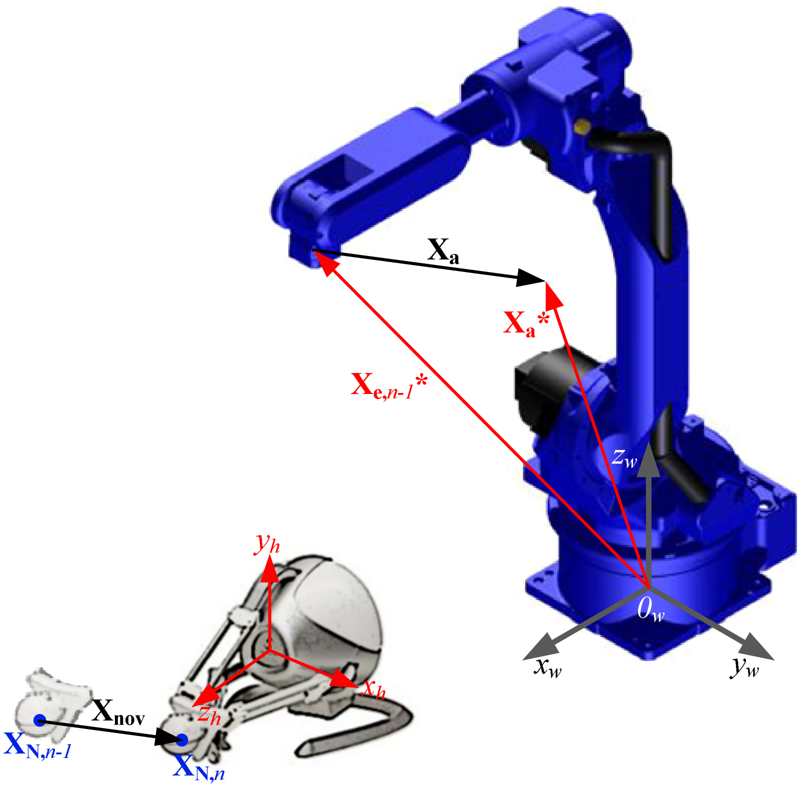


Figure 20 – Teleoperation approach. The robot displacement, $\mathbf{X_a}$, depends on the positional difference from the Falcon grip, $\mathbf{X_{nov}}$ and its velocity v_{nov}

In order to select what type of velocity function is needed some tests were carried out (Table 5). It was done six different tests extracting different number of motion data from the Falcon. Each test was executed with different motions. For each test was calculated the velocity magnitude v_{nov}.

Table 5 – Experimental tests for Falcon grip velocity calculation using a ballistic approach.

Ballistic Approach II

Test 1			
x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
-20,06	18,72	-12,04	14171
-27,58	8,33	-8,33	14281
v_{nov} [mm/s]= 121,36			

Test 2			
x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
-2,46	-38,10	-12,47	1765
2,46	-33,80	-7,62	1875
v_{nov} [mm/s]= 73,99			

Test 3			
x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
49,05	-11,13	6,05	6703
33,21	-6,02	2,38	6812
v_{nov} [mm/s]= 156,28			

Test 4			
x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
-79,32	-0,66	-15,42	19140
-79,37	-0,43	-15,57	19250
v_{nov} [mm/s]= 2,54			

Test 5			
x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
34,57	24,54	6,96	8765
7,98	41,63	2,56	8875
v_{nov} [mm/s]= 290,11			

Test 6			
x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
-78,89	12,24	-18,49	11671
33,21	13,13	-18,87	11781
v_{nov} [mm/s]= 1019,181			

By analysing results it is possible to observe that the achieved velocity values are relatively high when compared with the desired values for robot motion. By this way it is necessary to achieve a function of the velocity magnitude of the master device $f(v_{nov})$ to compensate this situation. In a more detailed analysis of Table 5, there are two tests that overhang (test 4 and 6), with completely different speed values. A slow movement on test 4 and a quick movement on test 6. The goal is to avoid high speeds. Therefore, some methods were studied to "achieve" a function $f(v_{nov})$ which complies with the desired values (moderated speeds). After some research, it is got an approach that gave good results, using roots functions. These functions when applied to data reducing the high values considerably and for the low values the change is not so high. Figure 21 graphically represents five root functions $x^{1/b}$ where b is the degree of the root.

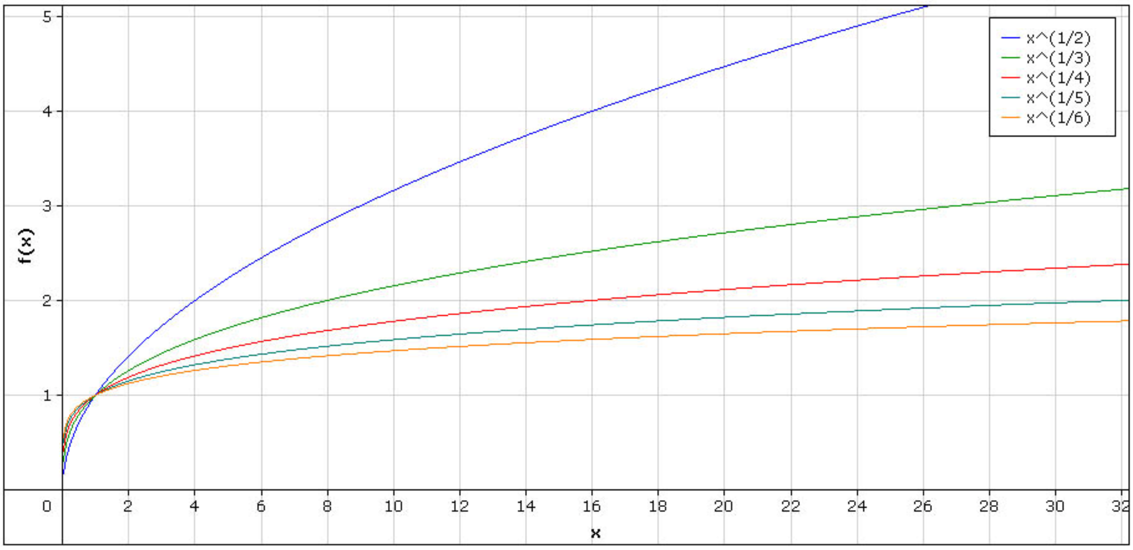


Figure 21 – Graphics of the square, cubic, fourth, fifth and sixth roots functions.

The proposed function is:

$$f(v_{nov}) = k_{nov} \times \sqrt[b]{v_{nov}}\ ,\ k_{nov} = cta \in \mathbb{R}\ ]0,1] \qquad (12)$$

Where, k_{nov} is a Falcon velocity scale factor. Some tests are made with k_{nov}=1 and changing b between 2 and 8. For the case of the square root the results were not the best because the movements of the slave robot were large and fast, giving a poor control. The cubic, fourth and the fifth root have provided a good control of the robot (these three values give the best results). After the fifth root the robot control did not change much, but did not cease to be able to be used.

After this implementation it was done a table identical to the Table 5 (Table 6), with the same test, but with the function of the velocity and the displacement for the slave robot. These test were done with k_{nov}=1 and b=4. Analysing the table results it is possible to conclude that the implementation of function $f(v_{nov})$ smooth $\mathbf{X_a}$ data.

Table 6 – Experimental tests for Falcon grip velocity calculation using a ballistic approach.

Ballistic Approach II

Test 1			
x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
-20,07	18,72	-12,04	14171
-27,58	8,33	-8,33	14281
v_{nov} [mm/s]= 121,36			
$f(v_{nov})$ = 3,32			
x_a [mm]	y_a [mm]	z_a [mm]	
-24,95	-34,48	12,31	

Test 2			
x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
-2,46	-38,10	-12,47	1765
2,46	-33,81	-7,62	1875
v_{nov} [mm/s]= 73,99			
$f(v_{nov})$ = 2,93			
x_a [mm]	y_a [mm]	z_a [mm]	
14,45	12,59	14,23	

Test 3			
x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
49,05	-11,13	6,05	6703
33,21	-6,02	2,39	6812
v_{nov} [mm/s]= 156,28			
$f(v_{nov})$ = 3,54			
x_a [mm]	y_a [mm]	z_a [mm]	
-55,98	18,05	-12,93	

Test 4			
x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
-78,89	12,24	-18,49	19140
33,21	13,13	-18,87	19250
v_{nov} [mm/s]= 3,88			
$f(v_{nov})$ = 1,40			
x_a [mm]	y_a [mm]	z_a [mm]	
0,11	0,53	-0,25	

Test 5			
x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
34,57	24,54	6,96	8765
7,98	41,63	2,57	8875
v_{nov} [mm/s]= 290,11			
$f(v_{nov})$ = 4,13			
x_a [mm]	y_a [mm]	z_a [mm]	
-109,73	70,55	-18,14	

Test 6			
x_N [mm]	y_N [mm]	z_N [mm]	*Time* [ms]
-78,8924	12,2428	-18,4912	11671
33,2133	13,1318	-18,8722	11781
v_{nov} [mm/s]= 1019,18			
$f(v_{nov})$ = 5,65			
x_a [mm]	y_a [mm]	z_a [mm]	
633,42	5,02	-2,15	

Finally the robot velocity magnitude v_a is given by:

$$v_a = \sqrt{\frac{x_a^2 + y_a^2 + z_a^2}{(t_n - t_{n-1})^2}} \tag{13}$$

4.2. Virtual Spring

In the proposed teleoperation approach, a virtual spring connects the master and slave devices. When the positions of the master and slave do not match, the virtual spring produces a force proportional to the difference in positions. This force acts as a desired contact force which will be affected by local force control. Therefore, this approach provides the human operator with contact forces within the bandwidth of the force sensor. Even in the free space operation of the slave system, the controller assumes that the robot is in contact with a very compliant environment [Park and khatib, 2006; Colgate *et al*, 1995].

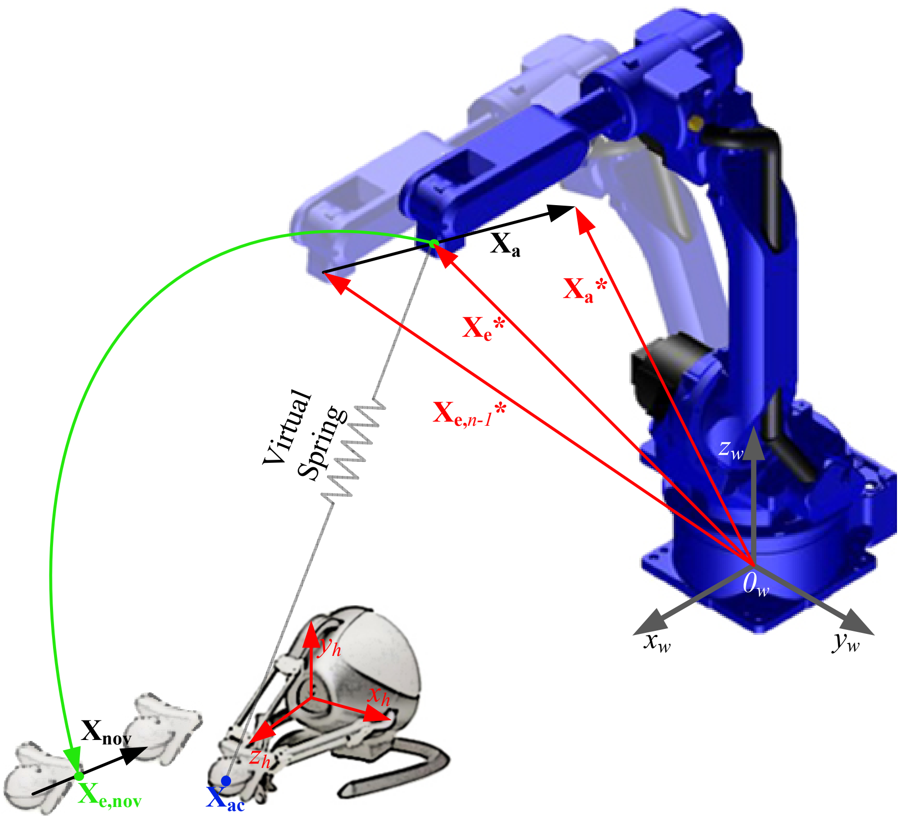


Figure 22 – Teleoperation approach with virtual spring. The robot end-effector position, $\mathbf{X_e}$*, is transformed to Falcon coordinate system, $\mathbf{X_{e,nov}}$.

The virtual force ($\mathbf{F_{vir}}$) come from a virtual spring with a constant elastic modulus k_{vir} and is proportional to a distance between the actual vector position of the master device $\mathbf{X_{ac}}$ and the slave robot end-effectors position in the master device coordinate system ($\mathbf{X_{e,nov}}$). Thus, the virtual force could be described by the following equation:

$$\mathbf{F_{vir}} = k_{\text{vir}} \times (\mathbf{X_{ac}} - \mathbf{X_{e,nov}}) \quad (14)$$

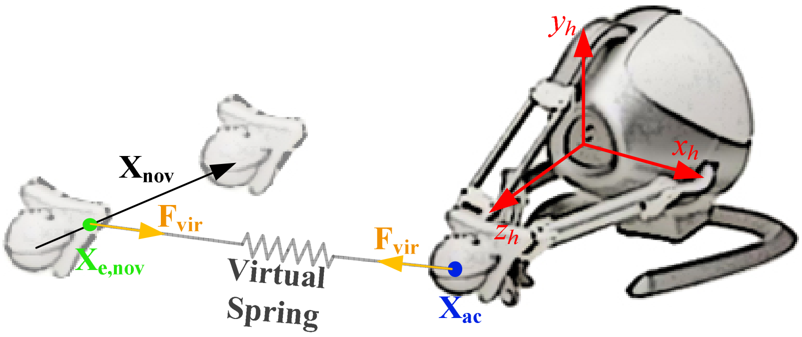


Figure 23 – Teleoperation approach with virtual spring. The virtual force, $\mathbf{F}_{\mathbf{vir}}$, is based on the position difference between the master grip and the slave robot end-effector.

The vector position of the slave robot end-effectors in the master device coordinate system $\mathbf{X}_{\mathbf{e,nov}}$ could be achieved by (15). To achieve it, there are two distinct cases; the first is when the user begins the operation to move the robot, in this case we do not have a pair of position vector measurements. Therefore the function of velocity magnitude of the master device $f(v_{nov})$ cannot be calculated and in this case the vector position will be equal to the first vector position of the measurements $\mathbf{X}_{\mathbf{N},0}$. The other case is a global one, where the coordinate system of the slave robot was transformed to the master device coordinates.

$$\begin{cases} \mathbf{X}_{\mathbf{e,nov}} = \dfrac{\mathbf{X}_{\mathbf{e}}{*} - \mathbf{X}_{\mathbf{e},n-1}{*}}{f(v_{nov})} + \mathbf{X}_{\mathbf{N},n\text{-}1}, \text{when } n-1>0 \\ \mathbf{X}_{\mathbf{e,nov}} = \mathbf{X}_{\mathbf{N},n-1} \text{ , when } n-1=0 \end{cases} \tag{15}$$

The actual robot displacement, $\mathbf{X}_{\mathbf{e}}$, is the difference between the actual vector position of the slave robot, $\mathbf{X}_{\mathbf{e}}$*, and the robot vector position $\mathbf{X}_{\mathbf{e},n-1}$* corresponding to the position vector of the Falcon, $\mathrm{X}_{\mathbf{N},n-1}$.

$$\mathbf{X}_{\mathbf{e}} = \mathbf{X}_{\mathbf{e}}{*} - \mathbf{X}_{\mathbf{e},n-1}{*} \tag{16}$$

The concept implemented in this study can be designated as a “fake” virtual spring, that is, in the concept of virtual spring exist a pair of forces action-reaction, in other words, the master device receives a force to pull the operator hand to the robot's position and the robot receives a force in the opposite direction. In this case this last force does not exist, because the robot does not allow that situation.

The virtual force gives the sensibility to the operator feel "where the robot is", and with this the operator has a better idea about the robot motion. Although, this force cannot be too high or it could be confuse with a real contact force. With the virtual spring the connection human-machine become more than an abstract and empty operation, and gives the robot a more human aspect.

4.3. Force Control

The local force control is a very important part, otherwise the more important when we are dealing with haptic interaction with real environments. So the forces sent by the F/T sensor are used to create reactions in the master device and in the slave robot.

4.3.1. Robot Reactions

The present study aimed that the robot changes its displacement vector according to the local force, in other words, when the slave robot comes in contact with an object it is intended that the displacement of the robot is corrected. This assures to the user that the robot movements will be short and with precision, guarantee that he could be focus on the task. Starting from this concept the corrected robot displacement vector, $\mathbf{X_{a,c}}$, decrease with the augment of the contact force $\mathbf{F_c}$ and when the contact force reaches a maximum value $F_{máx}$ it is pretended that the slave robot stops its movement and, of course, when there are no contact forces the robot must execute the normal displacement, $\mathbf{X_a}$ (Figure 24).

$$\begin{cases} \mathbf{F}_\mathrm{c} \nearrow \ \Rightarrow \ \mathbf{X}_\mathbf{a,c} \swarrow \\ \mathbf{F}_\mathrm{c} = F_\mathrm{máx} \Rightarrow \mathbf{X}_\mathbf{a,c} = 0 \\ \mathbf{F}_\mathrm{c} = 0 \Rightarrow \mathbf{X}_\mathbf{a,c} = \mathbf{X}_\mathbf{a} \end{cases} \tag{17}$$

From (17) and having only two boundary conditions, it was selected a simple method to calculate $\mathbf{X_{a,c}}$:

$$\mathbf{X}_{\mathbf{a,c}} = \mathbf{X}_{\mathbf{a}} - \frac{\mathbf{X}_{\mathbf{a}}}{F_{máx}} \times \mathbf{F}_{\mathbf{c}} \tag{18}$$

Follow by the condition:

$$\left|\mathbf{X}_{\mathbf{a,c}}\right| < \left|\mathbf{X}_{\mathbf{e}}{}^{*} - \mathbf{X}_{\mathbf{e},n-1}{}^{*}\right| \tag{19}$$

If the condition (19) is true then the slave robot stops its movement imediatly.

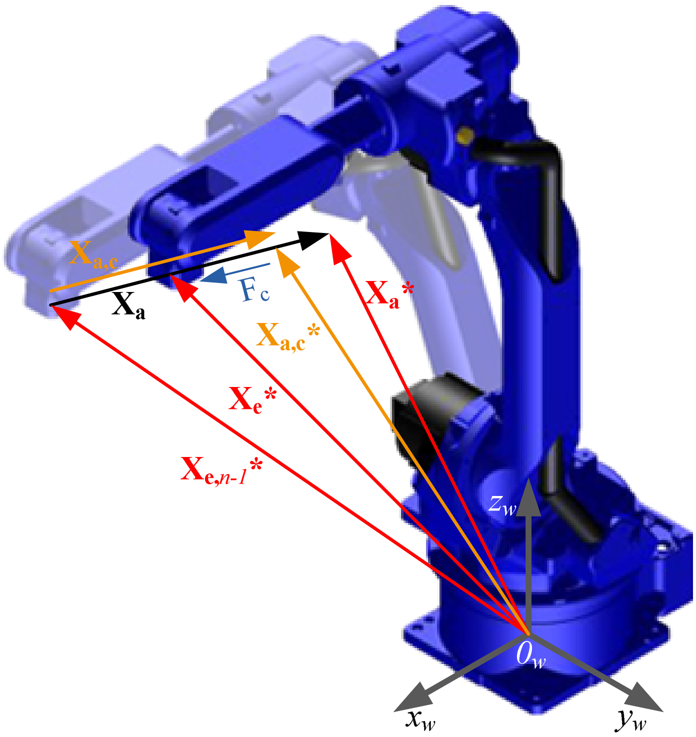


Figure 24 – Teleoperation approach with contact force. The contact force, $\mathbf{F}_{\mathbf{c}}$, will decrease the robot displacement, $\mathbf{X}_{\mathbf{a}}$, to a new and corrected robot displacement, $\mathbf{X}_{\mathbf{a,c}}$.

4.3.2. Novint Falcon Force Feedback

The Falcon force feedback system based on the third Newton law (action-reaction). In this method the contact forces vector $\mathbf{F}_{\mathbf{c}}$ from the F/T sensor, are multiplied by a scale k_{cmd}, to achieve a contact force vector to be applied on the master device $\mathbf{F}_{\mathbf{cmd}}$:

$$\mathbf{F}_{\mathbf{cmd}} = \mathbf{F}_{\mathbf{c}} \times k_{cmd} \tag{20}$$

As mentioned before, the method of the virtual spring is implemented to provide the user to always stay in contact with forces know by his senses "where the robot is". So in this effect, the desired force $\mathbf{F_d}$ sent to the master device could be described as the sum of the virtual force $\mathbf{F_{vir}}$ and the contact force to the master device $\mathbf{F_{cmd}}$:

$$\mathbf{F_d} = \mathbf{F_{vir}} + \mathbf{F_{cmd}} \tag{21}$$

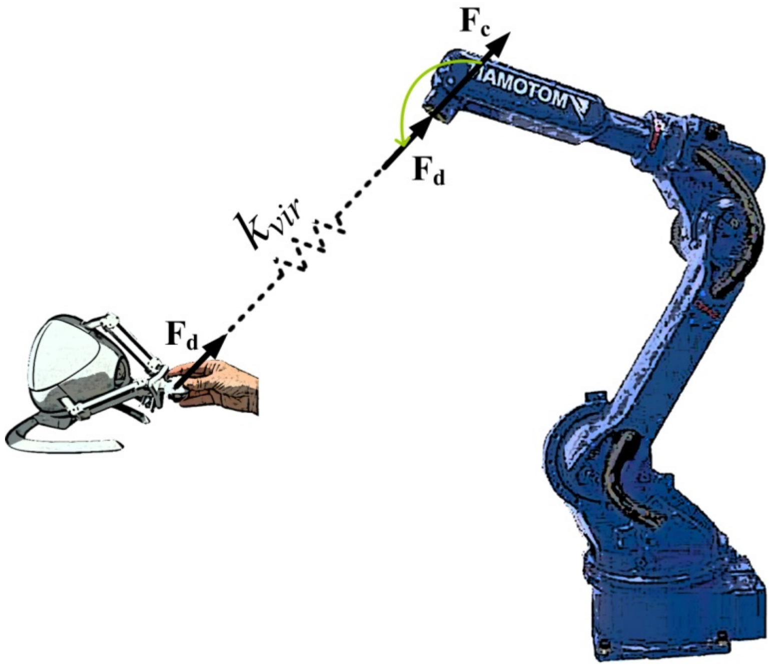


Figure 25 – Teleoperation approach with a virtual spring and force control. The desired force, $\mathbf{F_d}$, is produced by the virtual spring based upon the position difference between the master and slave robot end-effectors. $\mathbf{F_d}$ was enforced by the contact force, $\mathbf{F_c}$.

4.4. Control System

A block diagram representing the control methods applied on the system is present in (Figure 26). This diagram has been widely studied on this disertation and contains contributions from recent and largely reported papers in literature [Chotiprayanakul and Liu, 2009; Park and khatib, 2006; Cortesão *et al*., 2006; Cortesão *et al*, 2003].

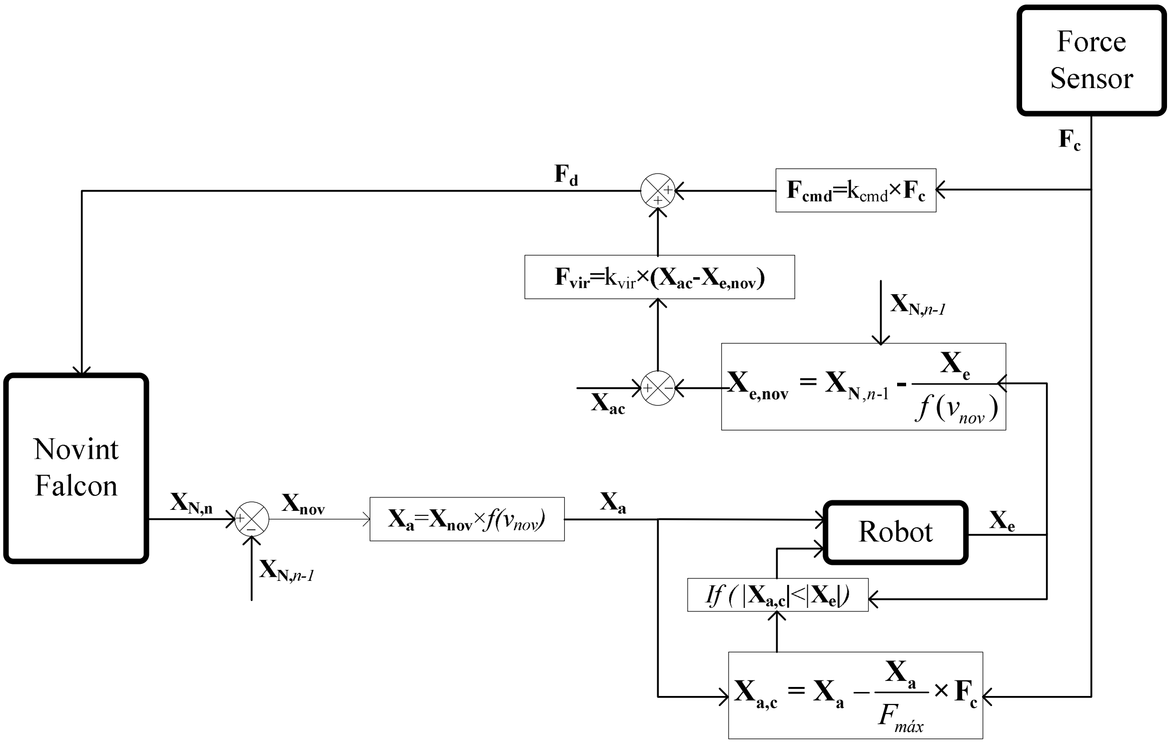


Figure 26 – A block diagram of the teleoperation control system.

4.5. System Interaction

In this section, it will be analysed "how all the system interact", in other words, how the system was implemented in practice. Advanced programming concepts like thread and multithreading were applied on the development of the software application that manages the system.

A thread (or thread of execution) is the smallest unit of processing that can be scheduled by an operating system. It generally results from a split of a computer program into two or more running tasks [Fraser, 2006].

The multithreading concept, is also named as multitasking, is the concurrently running threads. The easy way to explain is by an example, imagining a program executing itself, the program has two options:

(I) The program runs itself in one thread of execution. In this method the program follows the logic of the program from start to end in a sequential fashion. It possible to see this method as a single threaded.

(II) The program can break itself into multiple threads of execution or, in other words, split the program into multiple segments (with beginning and end points) and run some of them concurrently (at the same time).

The multithread system work by split the code and run each part at different time, in other words, the processor switch between different threads, it is like when executing two concurrently threads the processor first runs a part of the first thread, next it runs a part of the second, subsequently run the next part of the first thread, going on a loop [Fraser, 2006].

The multithreads gives good results when it is necessary to have some routines running at the same time, but like all methods it has advantages and disadvantages. The biggest concern of using simultaneous running threads is because each thread divides the clock processor, that means, each thread is running according is priority and as the process are always switching, could turn all the system slow.

Through the concepts explained previously it was built a first version of the system interaction (Figure 27).

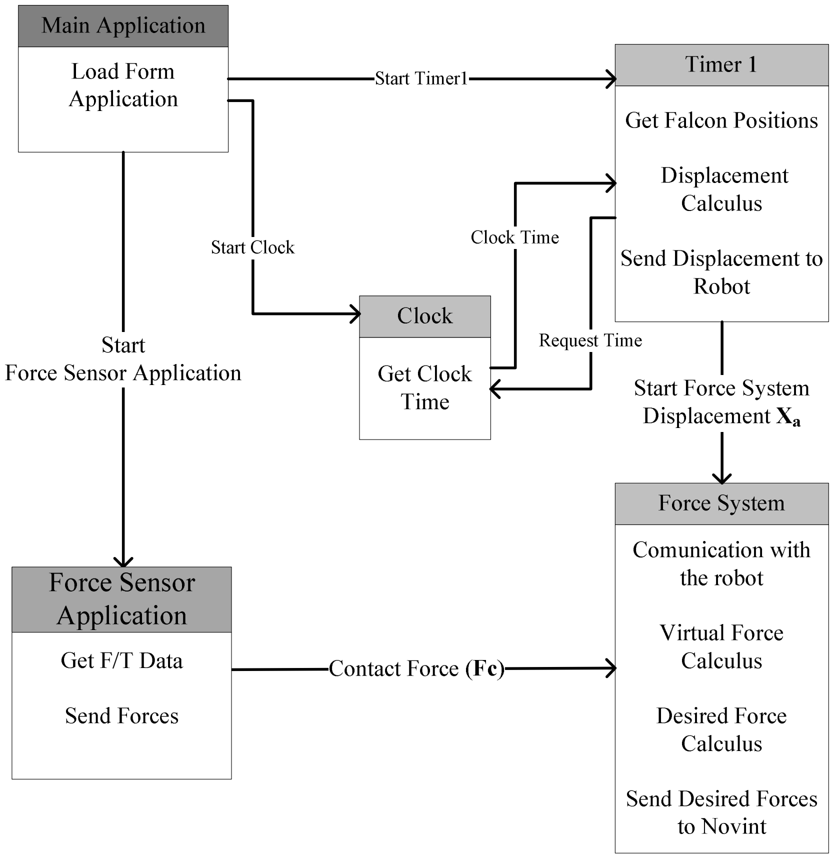


Figure 27 – System interaction (first version).

In order to a better understand the implemented system on Figure 27, it will be explained how the system interacts in a simple manner. The main application initialises the form application, and with its commands also starts the Clock, Timer 1 and Force Sensor Application. The Clock is a functionality to get the CPU clock time, this way it is possible to acquire concrete times from the CPU. The Timer 1 is responsible for activating the force system, to get the position of the Falcon, to calculate the displacement $\mathbf{X_a}$ and furthermore send it to the robot. The Force Sensor Application communicates with the force sensor and send information to the force system. The Force System has the heaviest part of the job, communicate with the robot in order to know in what position it was, do the calculus of the virtual force and the desired force and send this information to the Falcon. Moreover the force system also adjusts the robot position.

Unfortunately, this first version does not work well, mainly because the Falcon and the force sensor have different and incompatible C++ compilers. The Novint Falcon uses a compilation model /clr (Common Language Runtime), and the force sensor uses a compilation model /clr:pure. Normally it is possible to mix compilation models but these two models cannot be combined [Heege, 2007].

After some study about the problem, it was decided to use a socket UDP client/server connection, allowing by this way to have the F/T sensor to interact with the Falcon. A socket represents a single connection between two network applications. These two applications normally run on different computers, but sockets can also be used for inter process communication on a single computer. Applications can create multiple sockets to communicate with each other. Sockets are bidirectional, meaning that either side of the connection is capable of both sending and receiving data [Mitchell, 2010]. The UDP (User Datagram Protocol) is a simple OSI (Open System Interconnection) transport layer protocol for client/server network applications based on Internet Protocol (IP). In the client/server application there really is not much difference between the server and the client except that the server is waiting for packages from someplace, and a client is initiating the conversation and expecting some type of action from the server [Fraser, 2006].

In the first version of the developed interaction method exists another problem, related to the communication with the robot, lets rephrase, in the first run the application send the vector displacement to the robot and it executed the movement without any problem, but the time communication between the two displacements was too high for the proposal approach. Because of this it existed an empty time between robot displacements, which turned all the operation slow (for example sometimes the time-delay is superior to 300ms). Then there is a need to try a new approach to lose the empty times. One of the approaches that gave better results was using two threads (timers) to acquire the Falcon data and send the displacement vector to the robot. In this method the first timer acquire the position, does the calculus and before it begins the communication with the robot it sends information to the second timer to begin the cycle (the second timer is identical to the first timer). After established the communication with the robot, the first timer stay waiting for an instruction of the second timer to repeat the cycle.

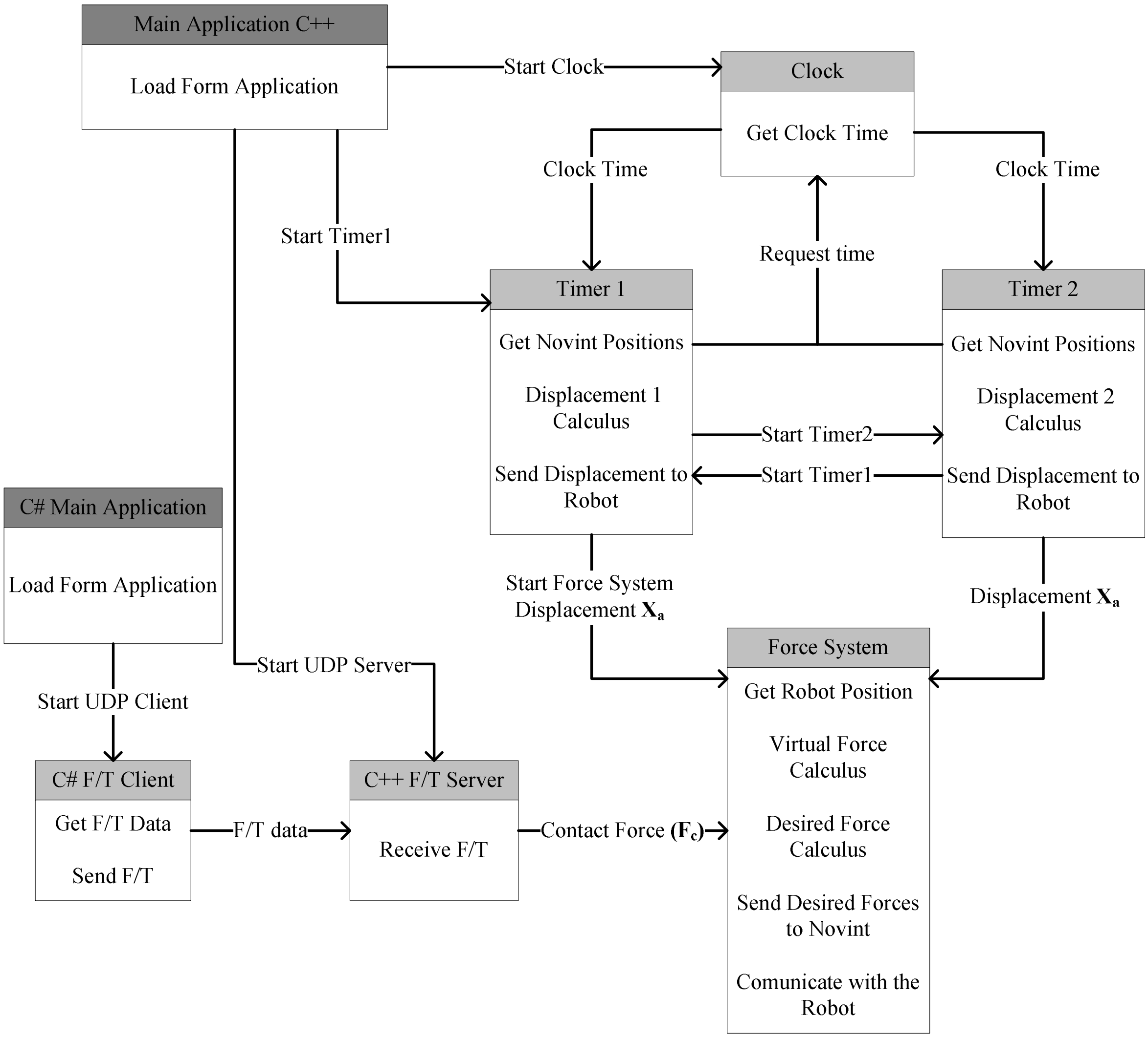


Figure 28 – System interaction (second version)

So, like it was done before in the first version, it will be explained how the second version of the system interacts (Figure 28). In this case it is necessary two main applications, the C++ application and the C# application. The C++ application starts the C++ form application and with its commands starts the Clock, Timer 1, Timer 2 and the UDP communication. The Clock is a functionality to get the CPU clock time, this way it is possible to acquire concrete times from the CPU. The Timer1 is responsible for activate the force system, get the position of the Novint Falcon, calculate the displacement $\mathbf{X_a}$, activate the Timer2 and furthermore send the displacement to the robot. The Timer2 is similar to the Timer1. The C# main application load form C# application and activates the UDP communication. The C# F/T Client communicates with the F/T sensor and starts the communication with the server (sending information to it). The C++ F/T Server accepts the communication with the client and begins receiving forces and sends it to the Force System. The Force System has the heaviest part of the job, communicate with the robot to know in what position it is, doing the calculus for the virtual force and the desired force and send this information to the Falcon. Finally, the robot’s position is adjusted.

5. EXPERIMENTS AND RESULTS

The experience lived in practice is very important for the human being, because he has an intrinsic need to the experiment. All technological developments must be tested in order to be known the need for evolution.

Along the evolution of this study existed the necessity to improve the human-machine interaction, essentially through the introduction of tactile sense on a robotic platform. The haptic interface provides to the user the ability to sense the robot motion, through the resistance that the controller exerts on the hand. Thus, several experiments were made to better understand the way the system works and to verify its advantages and disadvantages (Experiment1).

Taking into account all possible objects that a robot end-effector can contact and knowing that these objects are from materials with the most different stiffness values, it is important to perform tests involving contact with different materials. It is important understand how the robot reacts to contact and how the forces are “transported” to the haptic controller, so that the operator can “feel” the resistance that the robot has encountered on its movement (Experiment 2). A simple manipulation tasks was also tested (Experiment 3).

The challenge is part of life, there is the notion that the existing limitations occur for those who refuse to develop. Given the constant evolution of technology and constant connection that we share "side by side", so “why don’t go beyond”. Since we can feel what the robot “feels”, why not put the robot doing some tasks that we also do in our daily life. It was proposed to demonstrate the ability of the robot to show us a human face by making a simple task like peeling a banana (Experiment 4).

To perform these experiments the user has to hold the grip of the Falcon (Figure 29), press the button1 and move it in the direction that he wants the robot moves, if the user releases the button1 the robot automatically stops its movement (stand in hold position), until the user presses again the button1 and execute hand movements. The button2 allows the user to turn ON/OFF the vacuum system on the robot end-effector, which is required when the user wants to pick up some or release an object. The button3 is used to save the actual position of the robot end-effector and can be applied when it is wanted to repeat operations as, for example an operation identical to the experiment 3 (see section 5.3). The button4 allows the user to hold the robot in an emergency situation.

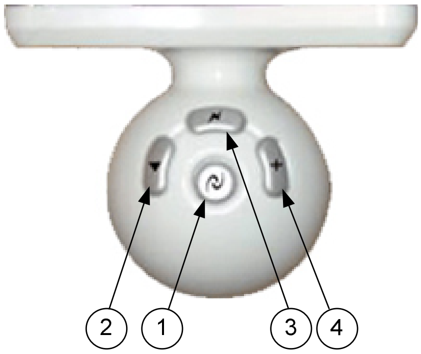


Figure 29 – Novint Falcon Grip. Each balloon represents the number of the button.

In order to execute the mentioned experiments, the user has the possibility of changing some parameters on the robotic platform:

(I) The variation of two consecutive time intervals (t_n-t_{n-1}).

(II) Scale factor of the velocity magnitude of the master device (k_{nov}); this factor will be used when the user wishes to navigate the slave robot to the work area (k_{nov}=1) or when he needs to perform a precision task (k_{nov}=1/4).

(III) Work with or without the virtual spring, in case of the user employing the virtual spring he will feel a force proportional to the difference between the robot end-effector position and the actual position of the master device.

(IV) Constant elastic modulus of the virtual spring (K_{vir}) in case of being selected the virtual spring option.

(V) Virtual force limit ($F_{vir,lim}$); in case of being selected the virtual spring option. If the virtual force passes a certain limit it can be confused with real contact forces.

(VI) Maximum contact force ($F_{máx}$); in case of unselect the virtual spring option the system will not change the correction displacement vector, because on this method it is intended to reduce the communication channels with the robot. The robot is repositioned when it goes beyond a maximum force, in this case the robot takes the control of the operation and moves to a new position to reduce the contact force.

(VII) Scale factor for the contact force of the master device (k_{cmd}).

(VIII) In all tests was used the fourth root degree (b=4), but this value can be changed.

5.1. Experiment 1 – Free Movement

This experience consists in testing the teleoperation system (and telepresence with the virtual force), videlicet, testing the movement of the robot when controlled by the master device and without contact forces. In this test the Novint Falcon haptic capabilities will not be fully exploited and only used as a motion controller (like a joystick). For this specific test it is pretend that the user moves the robot to a close position of the paper box (these types of box was also called as cardboard box).

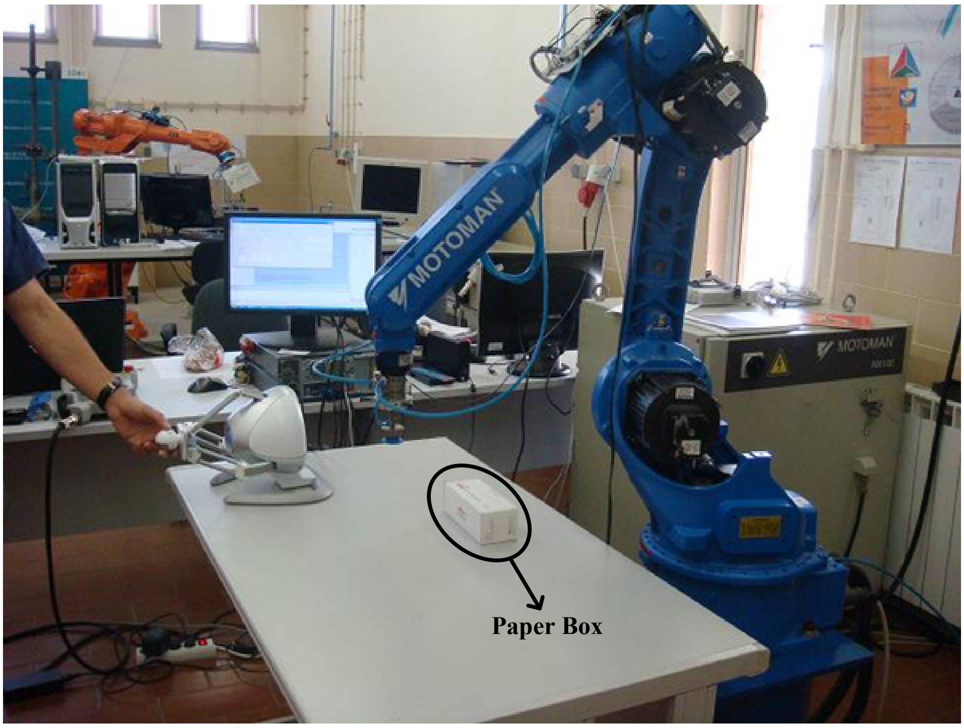


Figure 30 – Experiment 1 apparatus. The Objective of this experiment is to put position the robot end-effector to a close position of the paper box

In this specific experiment the control of robot movement is tested, changing of some system parameters like:

(I) The variation of two consecutive time interval (t_n-t_{n-1}); for this test this value is changed between 100 and 1000 ms

(II) Scale factor of the velocity magnitude of the master device (k_{nov}); for this test this factor take the values 1 and 0,25.

(III) Work with virtual spring or without it, in case of the user employes the virtual spring he will feel a force proportional to the difference between the robot end-effector position and the master device actual position.

(IV) Constant elastic modulus of the virtual spring (K_{vir}); the system will be tested with the values of 30 and 50 N/m

(V) Virtual force limit ($F_{vir,lim}$) will take the value of 2 N.

5.1.1. Results and Discussion

Generally speaking, the system works well, indeed it permit to control the robot in an easy and intuitive way, even for an inexperienced person in robotics (Figure 31). It is worth mentioning that the possibility to vary the velocity factor scale turn in to an advantage to perform accuracy movements. This experiment shows also some disadvantages and some aspect that need to be improved:

(I) There is a need to constantly repositioning the grip of the master device. This is due to the small workspace of the Novint Falcon, which does not allow large displacements (Figure 31).

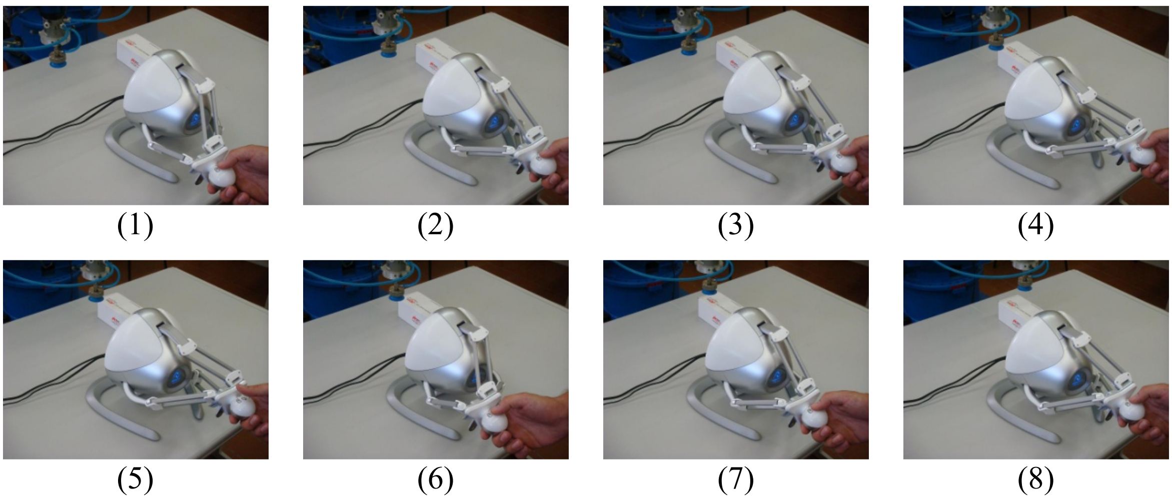
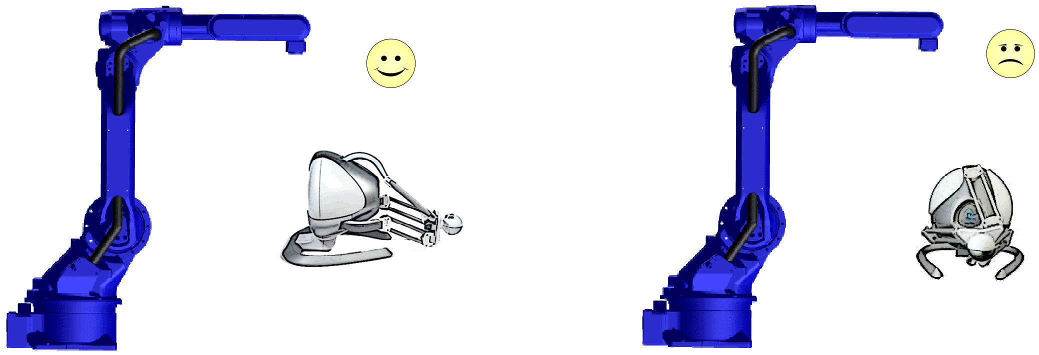


Figure 31 – The images show the repositioning of the Falcon; (1) button1 is pressed and the user begins to move the Falcon; (2) (3) and (4) movement of the Falcon until the user reaches the end of the controller workspace; (5) the user depress the button1; (6) reposition the Falcon and the user press the button1; (7) and (8) user moving the controller grip.

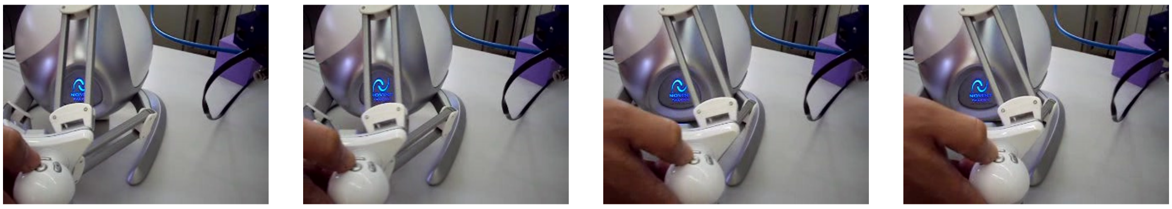
(II) When the coordinate system of both master and slave devices are misaligned the user feels uncomfortable with it and he takes some time to adapt to the new conditions (Figure 32).



(a) (b)

Figure 32 – (a) Falcon coordinate system aligned with the robot coordinate system; (b) Falcon coordinate system misaligned with robot coordinate system.

(III) The time-delay between the beginning of the displacement of the Falcon (pressing button1) and the beginning of the movement of the robot is a little high, but acceptable four our applications. Four trials were performed, for each one was make a film and the films were studied and analysed frame by frame in order to calculate the time-delay (Figure 33). It was achieved that the time-delay is approximately 140ms.



(1) (2) (3) (4)

Figure 33 – Images from one of the videos performed to get the time belay; (1) the user begins to press the button1; (2) the user is pressing button1 and begins to move the Falcon grip; (3) the robot begins to move; (4) robot motion is visible.

(IV) The virtual spring system works well and gives the feeling of "where the robot is", but unfortunately makes the system a little bit slow when compared with the same system but without using the virtual spring concept. This problem is due to the fact that when the virtual spring method is used it is necessary an extra communication channel with the robot. The system with virtual spring normally fails when the time intervals is less than 600 ms and for time intervals inferior to 400 ms the system break in a few seconds.

(V) The variation of time intervals change the robot motion and for the user is complicate to move the robot if he is always changing the rate of time intervals. Also, were observed three different major cases for the variation of time intervals:

(i) For time intervals of 100 and 200 ms, it was detected a strange motion of the robot, the motion is fast and unstable. It is also verified that the robot fails to follow the movements of the controller, that is, the robot lost some motion commands.

(ii) For the time intervals between 300 and 600 ms, the system presents a good control, however for a haptics approach these times are too high.

(iii) The time intervals between 700 and 1000 ms present also a good control but the system reaction is very poor. It is good for the user to adapt to the system, in an experimental phase.

5.2. Experiment 2 – Objects Contact

This experience consists in testing the system with teleoperation and telepresence, videlicet, the robot will be remotely controlled by the master device and at the same time the operator can feel the remote environment. In this situation, the aim is to extend as far as the sensitivity of the force sensor reacts when the robot is touching an object, and how it brings these forces to the master device. This way the operator can feel the resistance that the robot found on their motion. Then, to execute this test were chosen two objects with different stiffness properties, a sponge and a rigid paper box. In both experiments it is tested the contact with the referred objects. The user has the possibility to change some system parameters:

(I) The variation of two consecutive time interval (t_n-t_{n-1}); for these tests this value was approximately 800 ms.

(II) Scale factor of the velocity magnitude of the master device (k_{nov}); for these test this factor take the value 0,25 (k_{nov}=1/4).

(III) Maximum contact force ($F_{máx}$) is 15 N.

(IV) Work with or without virtual spring.

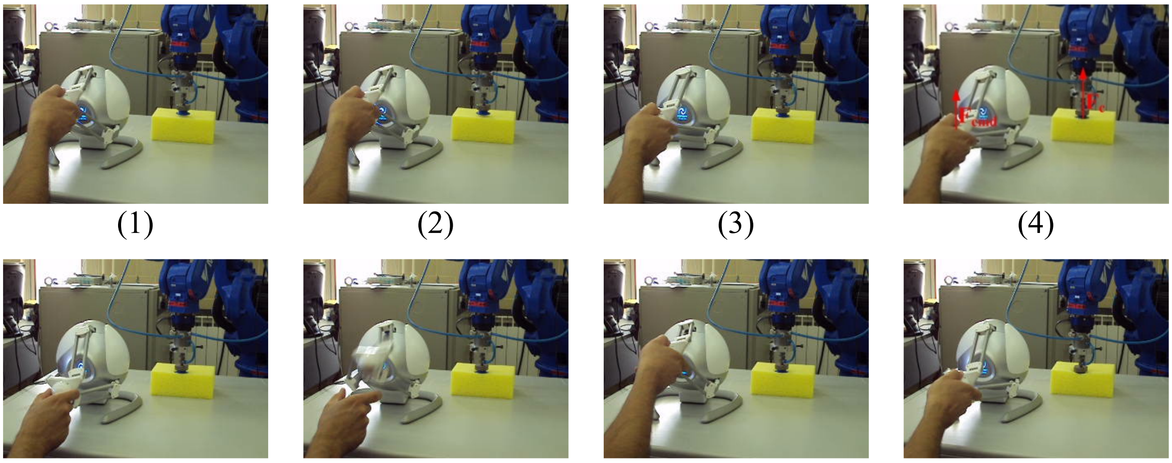
(V) Constant elastic modulus of the virtual spring (K_{vir}); for these test it was used the value 30 N/m.

(VI) Virtual force limit ($F_{vir,lim}$) will take the value of 2 N.

(VII) Scale factor for the contact force of the master device (k_{cmd}). This factor has the value of 1/2.

5.2.1. Test with Sponge

For the first test with contact forces was selected a sponge as a contact object. This choice was due to the physical properties of the sponge, because it allows contact with the robot without damaging the object and allows us to get interesting conclusions about the reaction of the Falcon. This test also allows the user to adapt to the system (Figure 34).



(1) (2) (3) (4)

(5) (6) (7) (8)

Figure 34 – Images from contact with sponge.

The test executed in Figure 34 shows the test with the sponge. It is possible to observe on the sequence of images that:

(1) The user is grabbing the Falcon grip.

(2) The user starts pressing button1.

(3) The user is moving the robot through the Falcon, and the robot end-effector begins contact with the sponge.

(4) The robot goes down 3mm in the sponge and it is possible to feel the forces on the Falcon grip.

(5) The user releases the button1 and grabs the Falcon grip.

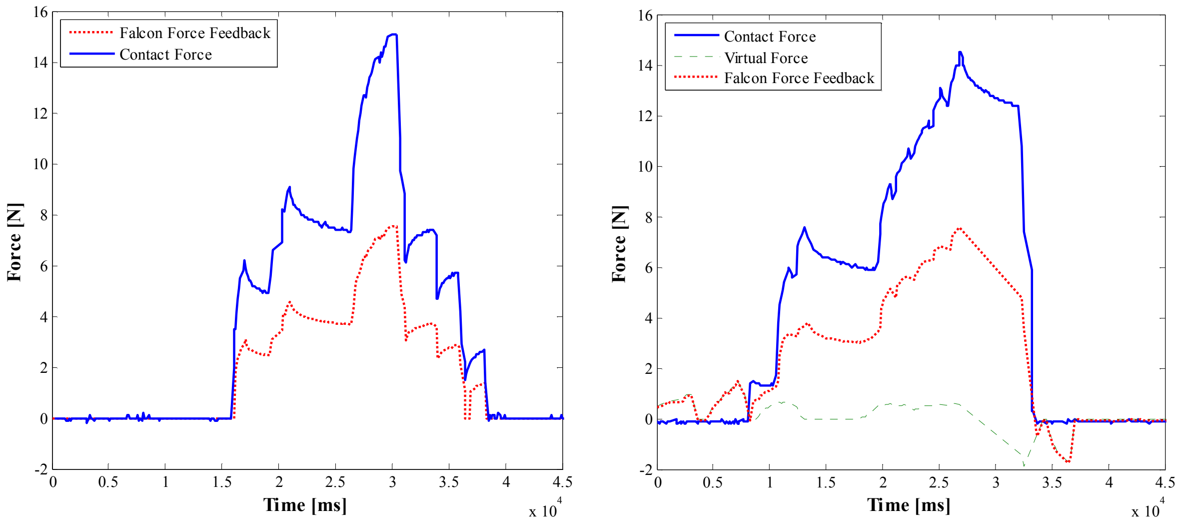
(6) The user releasing the grip and it rises due to the force feedback.

(7) The user begins a new movement.

(8) The user deepened into the sponge until reaches the maximum force and cannot go further.

5.2.1.1. Results and Discussion

The main purpose of this experiment is to study the contact with the surrounding robot environment (just focusing on it). The user receives force feedback from Falcon and perceives the forces acting on the robot end-effector (Figure 35). When the robot is in contact with the sponge it is interesting to note that the haptic device gives the feeling of pushing the hand up. The possibility of being able to “feel” objects is a good advantage because this sensation is increasingly necessary in robotics field.



(a) (b)

Figure 35 – Graphics of forces along Z axis in the sponge tests; (a) without virtual spring; (b) with virtual spring.

Figure 35 shows the graphics of forces along the Z axis for the sponge test. On both graphics it is possible to see the characteristic of the sponge. During the impact the sponge gave a hard reaction, but after the impact the sponge reduces the reaction forces. When the contact forces start decreasing the reaction is on the opposite direction. On Figure 35-b it can be seen that the virtual spring is an advantage because when the robot goes against an object the force feedback increases and when the user wants to relieve the forces that the robot is subject the virtual spring gives a “help”.

The option of work with velocity scale factor (k_{nov}) equal to 0,25 gave us a good mobility when it is necessary to execute tasks accurately, just as in this case.

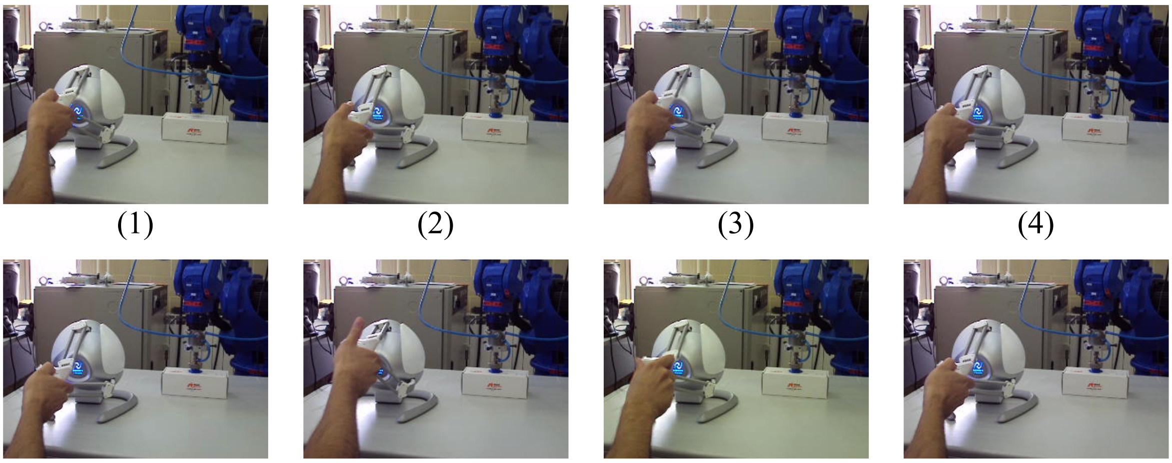
As mentioned before with the virtual spring implemented on the system, it is possible to feel "where the robot is" and at the same time feel the forces being exerted on it. Unfortunately the system is a bit slower compared with the same test without the virtual spring. The real-time concept becomes more difficult when the virtual spring is applied.This test shows also some disadvantages and some aspects that needs to be improved:

(I) The Novint falcon does not allow high rates of force feedback. When the frequency of sending forces is greater than 10Hz sometimes causes the Novint Falcon to reboot its system and it takes time to become operational again

(II) The Flacon has sometimes vibrations due to the frequency of the received force feedback.

5.2.2. Test with a Rigid Paper Box

For the second test with contact forces it was selected a rigid paper box as a contact object. It is pretended to test the system with a rigid object with different properties from the sponge. This way it is possible understand how the Falcon reacts when the robot contacts with a more rigid object (Figure 36).



(1) (2) (3) (4)

(5) (6) (7) (8)

Figure 36 – Images from the contact test with a rigid paper box.

Figure 36 shows the contact test with the paper box. It is possible to observe on the sequence of images:

(1) The user repositions the robot end-effector.

(2) The user depresses button1when the robot almost reach the paper box.

(3) The beginning of a new movement.

(4) The robot end-effector is already touching the box.

(5) The user goes deepened into the box.

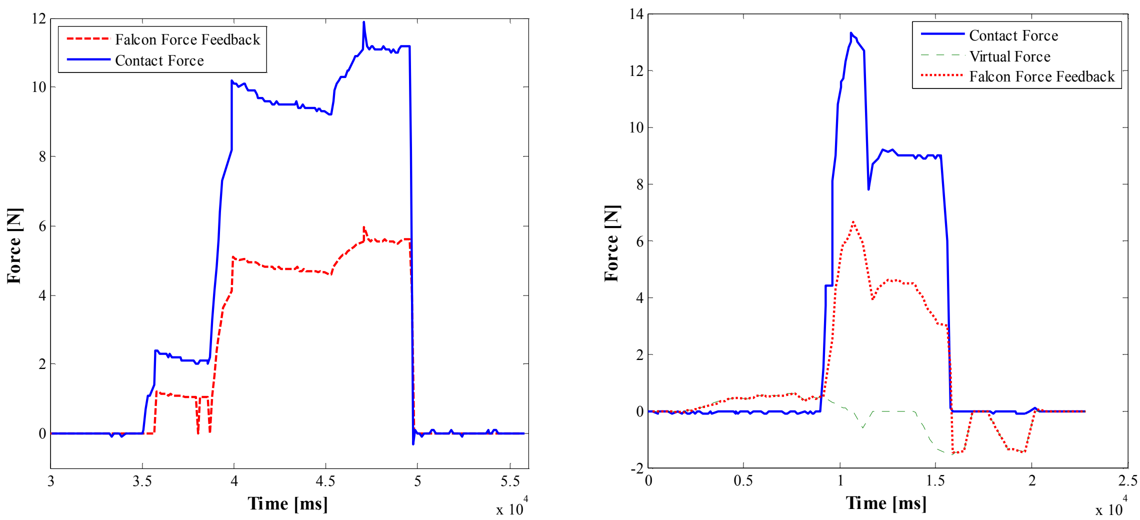
(6) The user releases the grip and it rises due to the force feedback.

(7) The user takes the grip down to the maximum position that the force feedback of the controller allows.

(8) The user executes an upward move to relieve the force.

5.2.2.1. Results and discussion

The user receives force feedback in the Falcon grip and the forces that the robot is subject are perceptible. One difference that was found, from the contact with the sponge, was that when the robot was in contact with the box it was interesting to realize that the haptics device gives a hard push up in the user hand (Figure 37).



(a) (b)

Figure 37 – Graphics of the forces along Z axis in the contact test with a paper box; (a) without virtual spring; (b) with virtual spring.

Figure 37 shows the graphics of forces along Z axis for contact with the paper box. In both graphics it is possible to observe that the characteristic of the paper box are different from the sponge. In this case the forces rise or decrease quickly when the robot end-effector enter or leaves the contact with the box. In Figure 37-b can be seen that the virtual spring is an advantage (like in the sponge test), because when the robot goes against an object the force feedback increases and when the user wants to relieve the forces that the robot is subject the virtual spring gives a “help”.

In this test, besides the disadvantages mentioned for the sponge test, it shows another problem. Owing to the time delay sometimes the system can not react promptly when the maximum force ($F_{máx}$) is exceed.

5.3. Experiment 3 – Object Manipulation

This experience consists in testing the system with teleoperation and telepresence and in this case it will be a mix of Experiment 1 and 2. Therefore, the aim of this experiment is to carry a small paper box to the interior of a big paper box and close the big paper box (Figure 38). In this test were used as system parameters:

(I) The variation of two consecutive time interval (t_n-t_{n-1}), for this test this value is approximately 800 ms
(II) Scale factor of the velocity magnitude of the master device (k_{nov}) in this cases this factor take the values of 1 and 0,25.
(III) Contact force maximum ($F_{máx}$), in this test this value is 20 N.
(IV) Work with or without virtual spring.
(V) Constant elastic modulus of the virtual spring (K_{vir}), for this test was used the value 30N/m.
(VI) Virtual force limit ($F_{vir,lim}$) has the value of 2 N.
(VII) Scale factor for the contact force of the master device (k_{cmd}). This factor has the value of 1/2.

Figure 38 shows the images of the object manipulation test. It is possible to observe on image:

(1) Experiment apparatus.
(2) The user moves the robot to a close position to the small box.
(3) The robot end-effector is in contact with the small paper box and the vacuum is turned on.
(4) Lifting up of the small box.
(5) The small box is on the top of the big box.

(6) The small box is being placed inside the big box

(7) The vacuum is turned off and the robot drops the box.

(8) Reposition of the robot end-effector, to close the big box.

(9), (10) and (11) robot end-effector closing the box.

(9) The big paper box is closed.

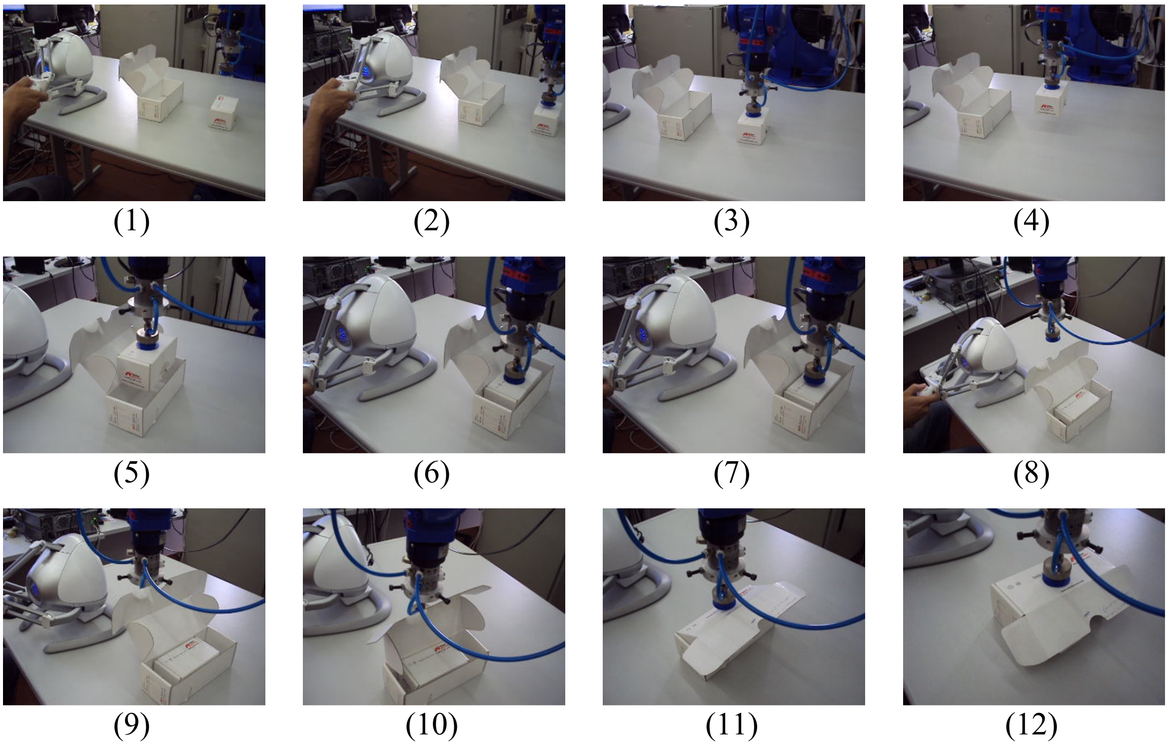
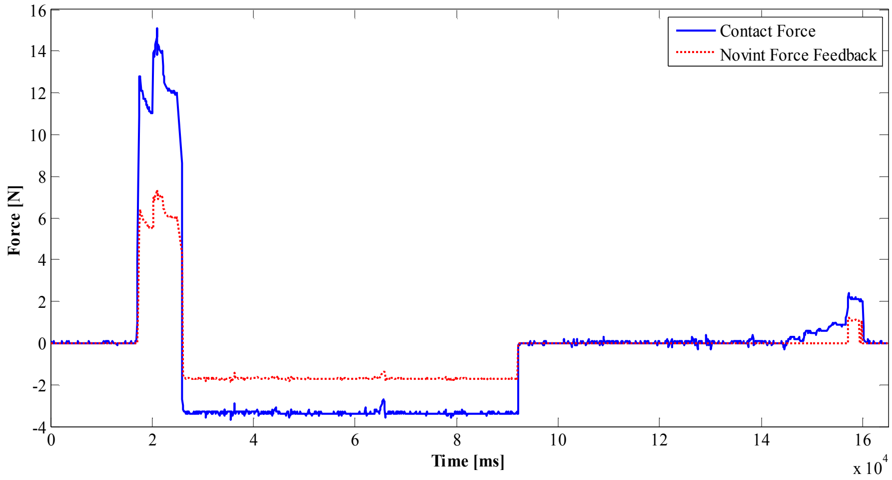


Figure 38 – Images of the object manipulation test.

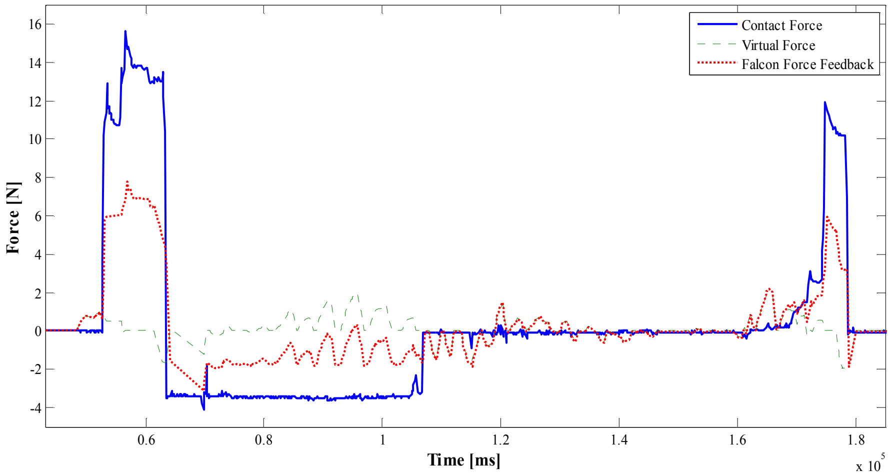
5.3.1. Results and Discussion

The purpose of this test is to study how the system interacts and works with different tasks. The results are similar to the previous tests. It is intended to note that when the robot was grabbing the small box the controller was being pushed down, giving even the notion that the user was carrying something.

Figure 39 shows the graphics of forces along Z axis for the manipulation objects test. In both graphics it is visible when the robot end-effector is touching an object and when is carrying it. For carrying objects, the virtual force option do not gave as good results as without it. It is perceptible in Figure 39-b that when the robot is carrying an object, the forces which go to the controller are always oscillating, confounding the users. For this case the virtual spring do not provide such good results as when the contact simple with objects.



(a)



(b)

Figure 39 - Graphics of the forces along Z axis in the test of object manipulation; (a) without virtual spring; (b) with virtual spring.

5.4. Experiment 4 – Cutting Operation

This experience consists in testing the system with teleoperation and telepresence but with a more challenging task, like it was mentioned before peeling a banana. In this situation the aim is to extend as far as the system can go to execute a more difficult task. In this test were used as system parameters:

(I) The variation of two consecutive time interval (t_n-t_{n-1}), for this test this value is approximately 300[ms].

(II) Scale factor of the velocity magnitude of the master device (k_{nov}), in this cases this factor will be use to perform a precision task (k_{nov}=1/4).

(III) Contact force maximum ($F_{máx}$); in this test this value is 20[N].

(IV) Work without virtual spring option.

(V) Scale factor for the contact force of the master device (k_{cmd}). This factor has the value of 1/2.

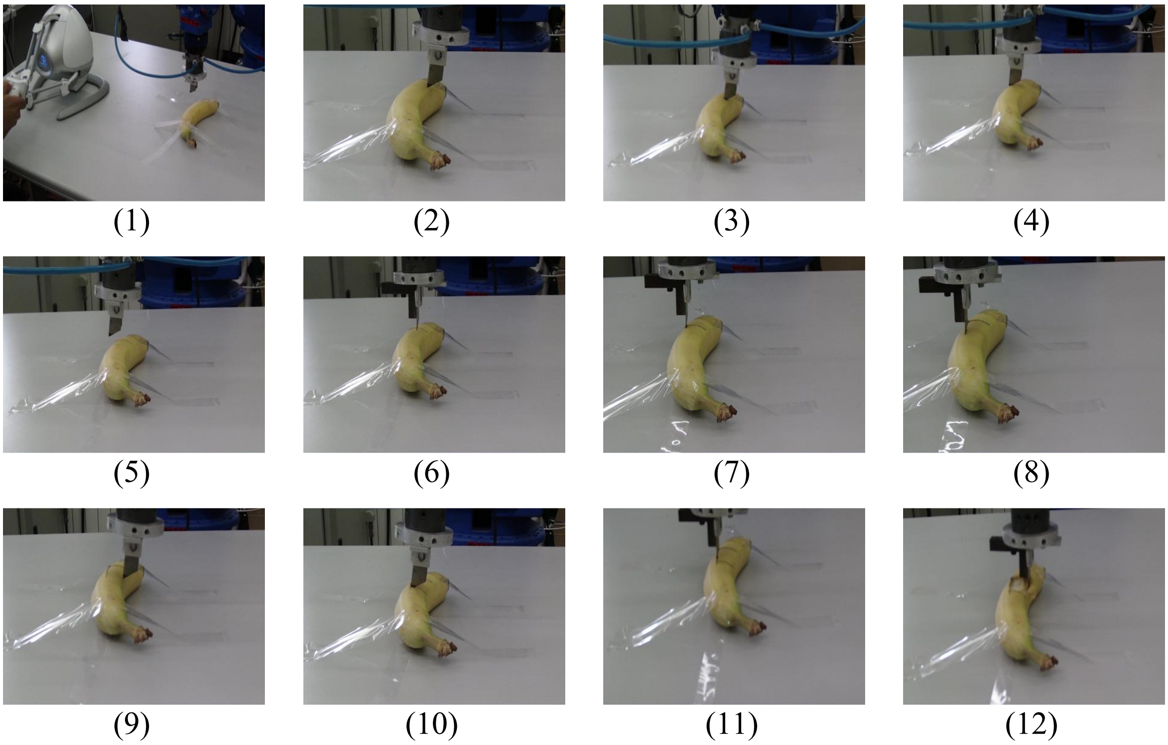


Figure 40 – Images from the cutting operation - peeling a banana;

The cutting operation is showed on Figure 40 and is possible to observe on image:

(1) Experiment apparatus.

(2) Positioning of the robot.

(3) The blade has entered into the banana peel and the cutting operation begins.

(4) End of the first cut.

(5) The robot end-efector is repositioned

(6) The robot end-efector rotate 90 degrees around Z axis

(7) Beginning of the second cut.

(8) End of second cut.

(9) The robot end-effector is positioned to the third cut.

(10) End of the third cut.

(11) The robot end-effector is positioned to remove the banana peel.

(12) The banana is partially peeled.

5.4.1. Results and Discussion

The main purpose of this experiment is to study how the system behaves with an "unusual" task (for a robot). The user receives force feedback from Falcon and perceives the forces acting on the robot end-effector when cutting the banana peel. In this test there were not found significant differences in relation to what was mentioned in the previously tests.

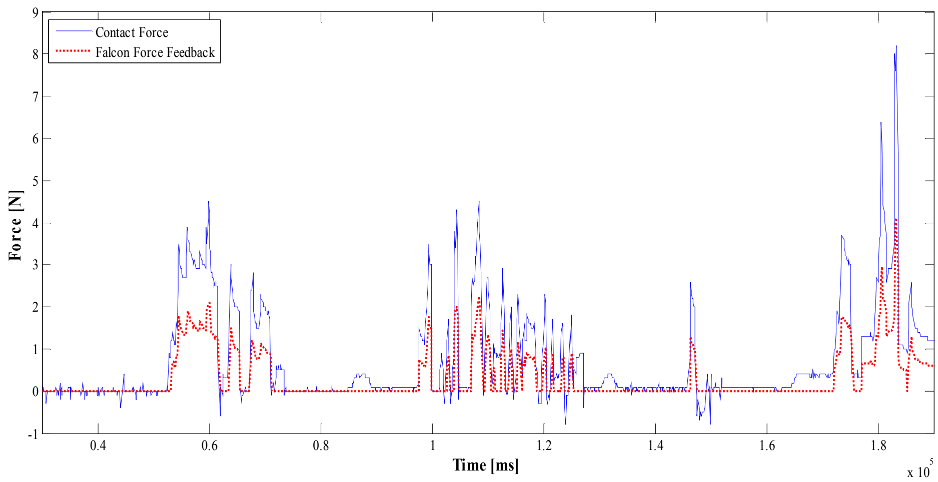


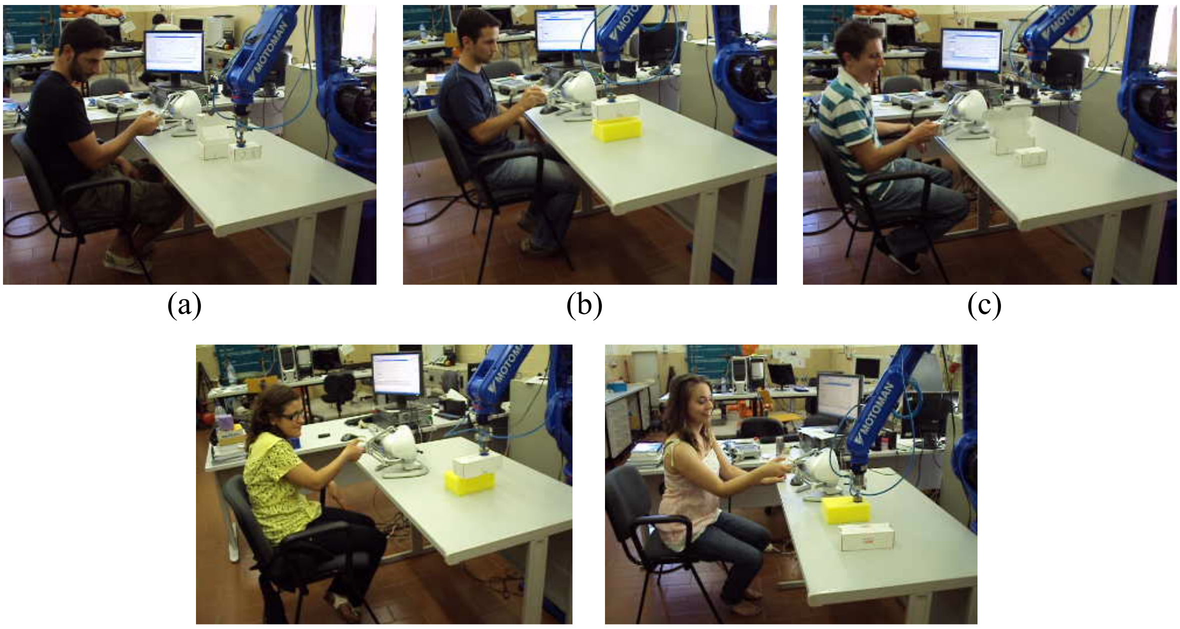
Figure 41 – Graphics of the forces along Z axis in the Cutting test.

Figure 41 shows the graphics of forces acting along Z axis for the cutting operation test. The graphic shows four different groups of peaks where is noticeable the cutting opperation. The three first groups of peaks are effective cuts and the fourth group is where the banana peel is removed. In each cut is visible that is not a continuous cut because this task is difficult and sometimes instead of doing a continuous cut the tool gets off the banana peel and the cut is almost executed by cuts up. The third cut is shorter because it is executed a quick cut.

5.5. Overall Results and Discussion

At my standpoint, all the tests have gone well. The biggest problem is to work with the virtual spring option which made the system a little slow. For the cases when the time intervals are smaller than 600 ms the system became slower (and the program sometimes crash). For real-time operations this values are inadmissible because for this type of operations the system should work with time response of at least less than 10ms. As it was mentioned before this slowlyness was due to the communications with the robot taking much time (at least 140 ms for each communication).

In order to be able to verify if the system was intuitive and easy to use, it was requested the help of five persons outside the robotics area (Figure 42): three male students in mechanical engineering and two female students, one studying in mechanical engineering and the other studying anthropology. These five persons have realized the experiments 1, 2 and 3.



(a) (b) (c)

(d) (e)

Figure 42 – Users testing the system.

At the end of the tests, were asked to the users "what they think of the system?". In general, all reported that the system is easy to work and very intuitive. They also reported that the interaction with the device receiving force feedback is much more interactive and intuitive. Just to mention that the female student of anthropology have presented more difficulties than the others to adapt to the system, which in my standpoint, this can be explained through the lack of contact with technology. Overall they all were able to perform these experiments in less than 10 minutes, and only needed a quick explanation (about 2-3 minutes) of how the system worked.

Finally, it is possible to say that this system is functional and could be applied in industry, but it still presents some issues that have to be solved in future works.

6. CONCLUSION AND FUTURE WORK

6.1. Conclusion

The work presented had as starting point the interaction with real environments using a haptic device. This system allows the user to control the robot without any technical knowledge while feeling the forces that the robot wrist is subjected. The use of a haptic device allows the control of an industrial robot in a simple and intuitive way, without requiring from the user advanced knowledge in programming of robots. Thus, the number of potential users of robotics can be expanded.

The first step of this work was to study the best control architecture for this system. Then, it was established and how all the components interact with each other. Some issues concerning to time-delays in the communication process were reported and analysed.

The second step of this study was to create a teleoperation system that would allow an operator to control a robot and feel the contact forces being exerted on the robot end-effector. Thus, was built a system that integrate three components; a ballistic control to provide the user a dynamic control of the robot, a virtual spring to connect the haptic device and the robot, and a recognition system of contact forces. After these three components were implemented, some tests were carried to see how all the system works. These tests indicate that:

(I) Two different ballistic control systems were tested. The first method did not give as good results as expected, but this system can still be used. This method requires a detailed analysis because it showed potential like, for example, to differentiate movements. In other words, through algorithms it was possible to know if the operator had done a straight move or if he had done a curve. However at this stage of the study we were unable to advance further in this method and explore its full potential. The second method showed good results, allowing the user to easily control the robot, but this process still needs to be improved in future.

(II) The concept of virtual spring as it has been mentioned before allows the operator to feel the motion of the robot. As result it gives the operator a direct sense of the whole operation that is taking place, which will lead to a more sensitive interaction with the objects. Thanks to this approach, any robot task does not become an abstract operation and without sensitivity. Unfortunately, the concept of virtual spring does not give as good results as without it, partly due to excessive time-delay in the robot communications, making all the system slow.

(III) The contact force system works as expected, allowing the user to know the forces that the robot wrist was subjected. The only drawback of this system was the communication with the robot, which delayed the whole system.

In general the system works well for a first approach to the haptics teleoperation, as is possible to observe in the performed tests, with different operation and objects.

Finally, the Novint Falcon for a low cost device is a very good haptic device. It allows working with teleoperation and telepresence, permitting the user to remotely control a slave robot while feeling the remote environment. This device has a lot of potential in applications like educating and training people, and in a near future it is expected that systems like the one tested in this work, can be applied in industry.

6.2. Future Work

The purpose of this thesis was to gather as much information as possible. However, there are still many tasks to be performed before this project reaches an industrial level. Some suggestions about future work:

(I) Have a computer running an operating system that allows real-time control, for example Linux-RT.

(II) Try to put out the UDP Server/Client system to acquire the F/Tdata and use only the main application to do that.

(III) Improve the ballistic control, for example by the variation of k_{nov} with the velocity of the master device.

(IV) Implement a recognizing movement trajectory forms system, to the robot execute the same movement as the user, and not only straight moves.

(V) Using a robot which allows low-level control.

(VI) Implement a control system that allows the motion of the robot in 6 DoF.

(VII) Apply the Kalman Active Observers filter (AOB), to estimate the system state and disturbances.

7. REFERENCES

Anderson, T., Golden, N., Jeff, Smith (2007), "Novint 3D haptics technology software", Sandia Corporation pp. 8-36. Available at http://www.sandia.gov/mission/ste/r&d100/2007winners/novint.pdf, April 13, 2010
Borst, C. (2008), "DLR Automatica 2008". In a DLR brochure for the Internacional trade fair for automation and mechatronics, Automatica 2008, Munich, Germany, june 10-13, 2008. Available at http://www.dlr.de/rm/en/desktopdefault.aspx/tabid-4789/7945_read-12712/, June 13, 2010.
Carneiro, M. (2003), "Interfaces Assistidas para Deficientes Visuais utilizando Dispositivos Reativos e Transformadas de Distância", Doctoral Thesis, pp.14-41.
Chavel, R. (1989). "Device for the movement and positioning of an element in space." United States Patent, 1990. Patent nº: 4976582
Choi, S. and Tan, H. (2004). "Effect of update rate on perceived instability of virtual haptic texture", Intelligent robots and system, Vol 4, September 28 – October 2, Sendai, Japan. Available at http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1389970, April 25, 2010.
Chotiprayanakul, P. and Liu, D.K. (2009), "Workspace Mapping and Force Control for Small Haptic Devide based Robot Teleoperation", International Conference on Information and Automation, June 22 -25, 2009, Zhuhai/Macau, China, pp.1613-1618. Available at http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05205175, April 13, 2010
Chotiprayanakul, P., Wang, D., Kwok, N. and Liu, D. (2008), "A haptic base human robot interaction approach for robotic grip blasting", International Symposium on Automation and Robotics in Construction, June 26 -29, 2008, Vilnius, Lithuania, pp.148-154.
Colgate, J., Stanley, M. and Brown, J. (1995). "Issues in the Haptic Display of Tool Use", International Conference on Intelligent Robotics and Systems IROS, August 1995, Pittsburgh, USA.
Conti, F. and Khatib, O. (2005), "Spanning large workspaces using small haptic devices", World Haptics Conference: First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virutual Environment and Teleoperator Systems, March 18-20, 2005, Pisa, Italy, pp. 183-188. Available at http://ai.stanford.edu/~conti/papers/WH05-CONTI.pdf, April 13, 2010
Cortesão, R., Park, J., Khatib, O. (2003). "Real-Time Adaptative Control for Haptic Manipulation with Active Observers". International Conference on Intelligent Robotics and Systems (IROS), October 2003, Las Vegas, Nevada, USA, pp. 2938-2943.
Cortesão, R., Park, J., Khatib, O. (2005). "Real-Time Adaptative Control for Haptic

Telemanipulation with Kalman Active Observers".IEEE Transactions on Robotics, October 2006, pp. 987-999.

Eid,M., Orozco,M. and Abdulmotaleb El Saddik (2007), "A guided tour in haptic audio visual environments and applications", International Journal of Advanced Media and Communication 2007 - Vol. 1, No.3 pp. 265 - 297. Available at http://www.mcrlab.uottawa.ca/journals/eoe_haptics07.pdf, April 12, 2010

El-Far, N., Nourian,S., Zhou, J., Hamam, A., Shen, X. and Nicolas, D. (2005). "A Cataract Tele-Surgery Training Application in a Hapto-Visual Collaborative Environment Running over CANARIE Photonic Network" IEEE Internacional Workshop on Haptic Audio Visual Environments and their Applications, October 1 and 2, 2005,Ottawa, Ontario, Canada, pp.29-32.

Fraser, S. (2006). "Multithreaded Programming" and "Network Programming" in *Pro Visual C++/CLI and the .Net 2.0 platform*, Appress, 2006, pp. 661-694, 695-728. Available at www.4shared.com, may 20, 2010

Gillespie, R. B. (2005), "Haptic interface to virtual environments", In *Robotics and Automation Handbook*. Tom Kurfess (ed.), CRC Press, 2005, chapter 23.

Hägele,M ,Nilsson, K., Pires, J. (2008). "Field and Service Robotics – Industrial robotics" in *Robotic Handbook Springer*, Bruno Siciliano and Oussama Khatib (eds), Springer, 2008, part F-42 pp-963-986.

Heege, M. (2007). "Expert C++/CLI", Appress, 2007, Chapter7, Chapter9, Chapter10 and Chapter12 pp. 143-172, 203-232, 233-251, 279-302. Available at www.4shared.com, may 20, 2010

Hunt, V. (1983), "Industrial Robotics Handbook", Industrial Press Inc.

IFR (2009). "World Robotics – Industrial Robots", IFR Statistical Department 2009. Available at http://www.ifr.org/, May 20, 2010.

Inition (2010), Available at http://www.inition.co.uk/, march 30, 2010.

JR3 Datasheets, JR3 Incorporation. Available at http://www.jr3.com/DataSheets.html, May 20, 2010.

JR3 Manual's, JR3 Inc.. Available at http://www.jr3.com/manual.html, April 20, 2010.

Martin, S., Hillier, N. (2009). "Characterization of the Novint Falcon Haptic Device for Application as a Robot Manipulator". In Australasian Conference on Robotics and Automation (ACRA), December 2-4, 2009, Sidney, Australia.)". Available at http://docs.nonpolynomial.com/libnifalcon/pdf/CharacterizationOfANovintFalconForApplicationAsARobotManipulator.pdf , March 12, 2010.

Mitchell, B. (2010). "User Datagram Protocol" Available at http://compnetworking.about.com/ may 25,2010

Neto, P., Pires, N., Moreira, A., (2010) "High-level programming and control for industrial robotics: using a hand-held accelerometer-based input device for gesture and posture recognition", Emerald Group Publishing Limited, 2010, vol. 37, pp. 137-147.

Nof, S. (1999). "Handbook of Industrial Robotics". 2nd ed. John Wiley & Sons.

Novint Technologies Incorporated (2008), "Haptic Device Abstraction Layer – programmer's guide". Available at http://home.novint.com/products/sdk.php, February 20, 2010.

Olsson, A. (2009). "Modelling and control of a delta-3 robot". Master thesis, February, 2009.

Palsbo, S. , Marr, D., Norbald, Hopkins, W., Bay, B., Streng, T., Valenti, B.(2008). "Desktop haptics for fine-motor hand rehabilitation (handwriting)". Available at http://www.obslap.com, April 15, 2010.

Park, J., Khatib, O., (2006). "A Haptic Teleoperation Approach Based on Contact Force Control", The International Journal of Robotics Research, Vol.25, May-June 2006, pp.575-591.

Salisbury, k., Barbagli, F., Conti, F. (2004), "Haptic Rendering: Introductory Concepts", IEEE Computer Graphics and Applications, Vol. 24, Issue 2, March, 2004. Available at http://jks-folks.stanford.edu/papers/Haptic-Rendering.pdf, April 20, 2010

Schill,F., Mahony, R., Corke, P., Cole, L.(2008). "Virtual force feedback teleoperation of the insectbot using optic flow". In Australasian Conference on Robotics and Automation, Canberra, Australia, December 3-5 2008. Australian Robotics and Automation Association. Available at in: http://www.araa.asn.au/acra/acra2008/papers/pap159s1.pdf, May 12, 2010

Soulié, J. (2007) "C++ Languague Tutorial". Available at http://www.cplusplus.com/doc/tutorial/, March 20, 2010.

Srinivasan, M. and Basdogan, C (1997), "Haptics in Virtual environments: taxonomy, research status, and challenges", computer and graphics, Vol.21, No 4, pp. 393-404.

Streng, T. (2008) "Mechanical linkage design for haptic rehabilitation and development of fine motor skills". Master's thesis, Oregon State University.