UNIVERSITY OF PADUA

FACULTY OF ENGINEERING
MASTER OF SCIENCE IN MECHATRONICS ENGINEERING

MASTER THESIS

RENDERING OF FORCE FEEDBACK ON A ROBOTIZED TELECHOGRAPHY SYSTEM: A GRAPHIC USER INTERFACE AND HAPTIC PROBE MODELING

Supervisor: Roberto Oboe

Correlator: Pierre Vieyres

Graduating: Marco Chiccoli
626208-IMC

ACADEMIC YEAR: 2011-12

SUMMARY

In this document the candidate works on some points to develop of two projects of robotic tele-echography inside the Prisme Laboratory. This last one is a laboratory of research situated in France, in the city of Bourges, one of the four campuses of University of Orleans. These two projects, that will be described in the next chapters, are centred on remote tele-echography: with use of one haptic probe the medical expert can control the end effector of robot manipulator to execute the echography from/to any place. In this project many people involved composed a large team. In this thesis the candidate has done a study centred on the haptic probe performing some simulations and developping parts of the system. This document is divided in the following points:

- After a introduction of the system, the first chapter will show the drawing of the haptic probe with the software "Blender", a software open source to the modeling of object in 3D. These model will be used, as specified in the next point, to develop the graphic user interface.
- In the second chapter, we will talk about the design of the graphic user interface with the software Qt, a Nokia software open source used to design the graphic interface. This interface allows the expert engineer to set up and to execute the tests on the system; moreover allowing the medical expert that executes the echography to have better information for his diagnosis.
- In the third part there will be a electrical and mechanical model of the haptic probe built with the tool of the software Matlab "Simulink". With this model was used to performed several simulations to verify the functionality of the model and the best configuration was searched to size the system in order to obtain minimum response times.
- The fourth chapter will describe two events in which the candidate has participated: the first one is "Wortex 2012", the event created with the collaboration of different universities to test and advertise the robotized teleoperated system; and "Bibe 2012", a IEEE conference that will be hold in Cyprus in November 2012; the laboratory Prisme will participate to the conference with the article on the work done by the candidate and two other people. The candidate is one of the authors of the article.
- The last chapter give the conclusions of the work and future proposals to be developed for the project.

THANKS

For this work I would like to thank the professor Oboe, that gave me the possibility and the contact to unwind my thesis in France; then I would like to thank all components of the Prisme laboratory for their hospitality and for all helps that gave me to finish and to complete my work. In particular I would like to thank Pierre Vieyres, Cyril Novales, Laurence Josserand, Tao Li and Nicolas Morettes. I would like to thank also my family, for the support, moral and economic, that gave me in all these years of university: my father Vanni, my mother Rosanna and my sister Martina. I would like to thank my girlfriend Chiara, that gave me a good support in all situations more criticals, staying near me and encouraging me for this work and this experience. I would like to thank also all my friends that, since our childhood, have always been near me in the need moments.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

This work is centred on the robot PROSIT and on the robot PROTECH which are utilized to execute tele echographies. These platform are divided in three main stations: the control point where the medical expert is, the remote robotic station where the patient is and the communication system to change data between the two stations. The first work in this document is centred on the graphic user interface of the robot: to execute the echography the medical expert needs a graphic interface to start and initialize the system, select the image of ecography and, when he has finished his telediagnosis, shut down the system . All these things have to be the more easy possible to drive the medical expert to do a echography without to find obstacles. The work is born to give to expert a new important function of the graphic interface: when the expert pilot his probe he can see on the graphic interface a model 3D of the probe that follows exactly the real movement of the probe and, moreover, he can see also a model 3D of the end-effector of the robot and he can compare the movement of his probe and the movement of the end-effector directly on his graphic interface to have an idea of how the ecography is going . The after work performed by the candidate is a study of the electrical and mechanical model of the haptic probe: after a mechanical study of the forces acting on the system, have been written the dynamics equations to have a direct and inverse model. In this model has been designed also a current loop in the motor in order to deliver the right power that will go to make feel the interaction between the robot manipulator and the environment at the medical. Moreover, after validation tests of the model, was also searched the optimal dimensioning of the system so that, under suitable assumptions, decrease response times. Another goal of this work is to describe synthetically the various tests that have been done with other countries to test the robot PROTECH. Particularly, the countries friends for these tests are Cyprus, Peru and Usa. We will describe how it function the program to draw in 3D the haptic probe and the end-effector of the robot, the program utilized to design the graphic user interface with all the problems that it meets during the project of these things and a little chapter that talks about the tests.

1.1 THE ECHOGRAPHY

The echography is the medical imaging noninvasive technique that utilizes ultrasounds wave to view internal parts of the body. Ultrasound is used in many medical applications such as cardiac, abdominal investigations and obstetrics. It is also used for the application of minimally invasive surgery to guide the surgeon when inserting the needle and also can be used to evaluate any injury or bleeding of some internal organ after an accident. The ultrasound principle is based on the propagation and reflection of ultrasound waves in the biological tissues. The ultrasonic probe, placed in contact with the skin of the patient and near the anatomic region analized by the expert emits ultrasonic waves that propagate on the whole body of the patient. When the ultrasound wave enter in contact with the body tissues, they diffuse or reflect generation echoes that are collected by the ultrasound probe. The ultrasound image is represented in greys scale levels which report a difference in the acoustical impedance(and echogenich) of the various tissue hit by ultrasounds wave. Fluids like blood or mucus make weaks echos, they appear more or less homogeneous in grey scale levels. The solid structure, e.g. the bones, are very echogenic structure and appear in bright white levels in the image (with the exception of the cranium that it is many thin and less echogenic). The softs tissues that are more or less echogenic, they appear with different greys levels depending on their nature. The air and the gas have hyperechoic properties, like the bones, and they seem very whites. During the echography exam, the medical expert applies a gel on the area of body that want observe. This gel acts as an interface avoiding the presence of the air between the probe and the skin, reducing the impedance mismatch. This technique has many advantages:

- The use of ultrasound is practically harmless, it is the only technique to have an image of the fetus with a good safety. There is no allergy or contraindication for this exam.
- It is painless for the patient, without anesthesia or hospitalization. It can be repeated without problems.
- The echography is an imaging technique inexpensive. Requires only a probe and the price of consumables can be negligible. The exam is executed by a specialist.
- The ultrasound system is mobile, capable of achieving the same examination of a patient bed, for example, in an intensive care unit.
- If the echography is executed by a medic, the result is immediate and it may require a second opinion for unusual pathological cases.

However, the examination, and therefore its results, is expert dependent. Measurements and image quality are highly dependent on the position of the probe and anatomical knowledge possessed by the expert. The diagnosis quality depends largely on the skill and competence of the doctor performing the ultrasound. The expert makes a mental reconstruction of the organ that is observed from ultrasound images, the inclination of the probe and its own anatomical knowledge. Given the little difference of gray levels between organs, it is impossible to do this unless there is a reconstruction of the information described above. This notion of expert-dependency is the main reason for the development of tele-echography robot with the goal to extend the gesture of the expert from a distance.

1.2 THE TELECHOGRAPHY

A tele-echography is an echography exam where the patient and the medical expert are in two different sites. Since 1990, the tele echography systems have evolved gradually from non-robotic systems to robotic systems, built by industrial robots to dedicated robotic systems. An assistant is next to the patient and he participates in operations to position the robot (in the case of tele-echography). The medical expert consults on its screen the patient's echography images from the site and makes his diagnosis. The first types of tele-echographies were not-robotized: these systems permitted the transmission of the echography images between two remote sites. The echography images were acquired by a nurse or by a paramedical driven by the medical expert through a video conference system; then image are sent to remote site for a diagnosis by experts. However, the expert-dependency concept make it difficult for the expert to work during the post-elaboration of the image because he doesn't know initially the probe position and this factor influences the quality of diagnosis. Another type of this technique consists in instrumented probe, allowing the acquisition and recovery of data in 3D in volumetric organs. A medical expert executes a scan of the surface to be examined and with instrumented probe without the help of the remote expert. The 3D data of the patient are sent to the medical expert that, with use of the virtual probe, he consults and he analyzes the 3D data offline through a dedicated software interface. The time delays are not a determinant factor in this type of system, they can be of several minutes for the data to be sent. The next step has been the tele-echography that is called robotized tele-echography utilizing a probe holder robot that allows of reproduce the movements of expert maintaining the probe in contact with the patient's skin. There are many systems of tele-echography robots, dedicated or less to a domain of application. Some robots are mobile, ie the mobility of system is defined like its capacity to be moved to an operating site while other ones are too heavy or cumbersome to be moved.

1.3 THE SYSTEM OF TELE-ECOGRAPHY PROSIT AND PROTECH

The PROSIT project[1](Robotics Platform for the interactive system in Tele-echography) was financed in 2008 by the National Agency of Research(ANR CONTINT 2008) and considered by the PRISME laboratory. The goals are to develop differents platforms of tele-echography robotic and to provide a force feedback to the expert level. Moreover, this system integrates different algorithms of visual tracking based on the ultrasound images with differents control modes(Remotely Operated or autonomous). The robot Prosit1, shown in fig.1, is a serial structure with 4 degrees of freedom; it is a spherical wrist with intersecting axes with an inclination feature as the first axis for extracting a singular configuration out of the workspace, and an axis allowing the translation of the probe along its axis so as to maintain the ultrasound probe into contact with the patient's body. This axis is equipped with a force sensor for controlling the contact force between the probe and the patient's body and to provide force feedback at the medical expert. The robot Prosit2, shown in fig.2, is a parallel structure. This robot has 5 degrees of freedom. In series, there is a horizontal translation, the orientation of the probe axis with the two arms, the rotation of the probe and the translation according to probe axis. Therefore, this robot has 3 degrees of freedom for the orientation and twos for translation. The robot Protech is a robot with 5 dof. This kinemat-

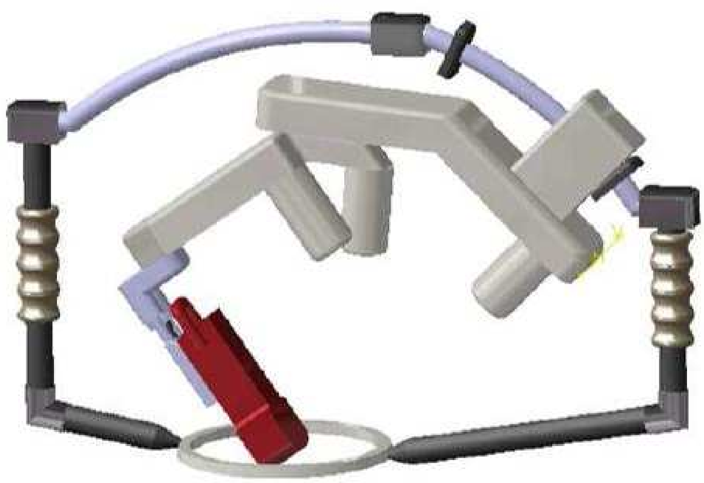


Figure 1: The robot Prosit 1

ics structure is a non-redundant robot with 5 dof: the first module allows to compensate the position error of the robot; the second module allows to orientate the end effector of the robot, and therefore the ultrasound probe; his role is to control the action of the expert. The

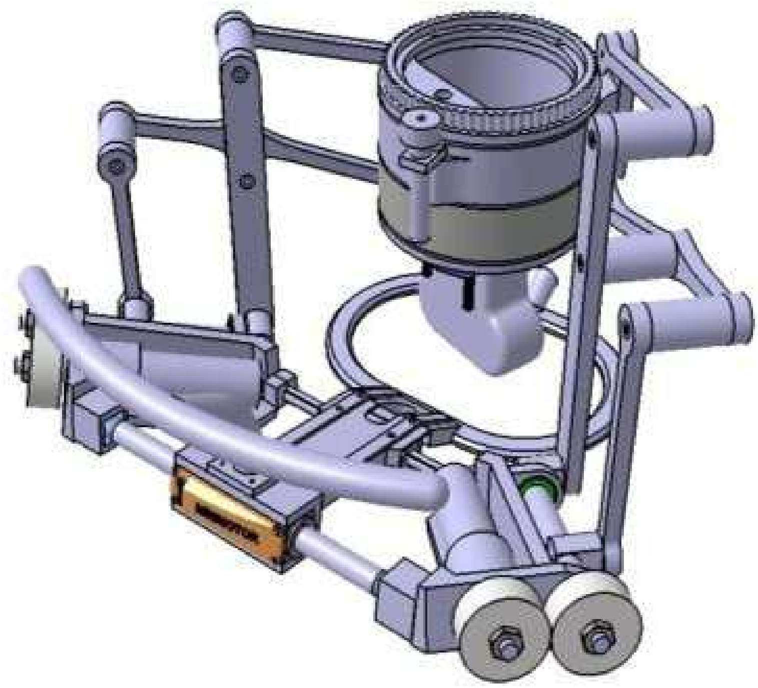


Figure 2: The robot Prosit 2

third module manages the force applied on the patient's skin. This robot is shown in the fig.3.

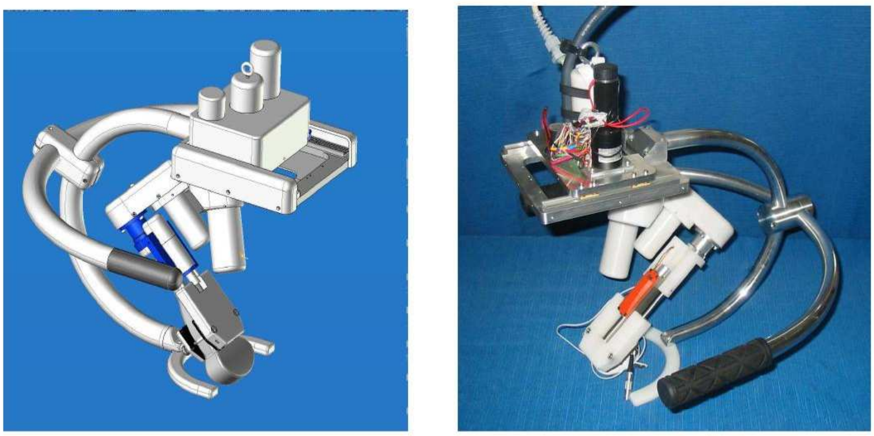


Figure 3: The robot Protech

1.3.1 *The overall architecture*

The robot architecture is composed from three different subsets:

1. The expert station, where the expert commands the remote robot.
2. The patient station, where the robot is positioned by an assistant but it is maintained on the patient's body.

3. The line of communication allows the transfer of data from the expert station to patient station and viceversa.

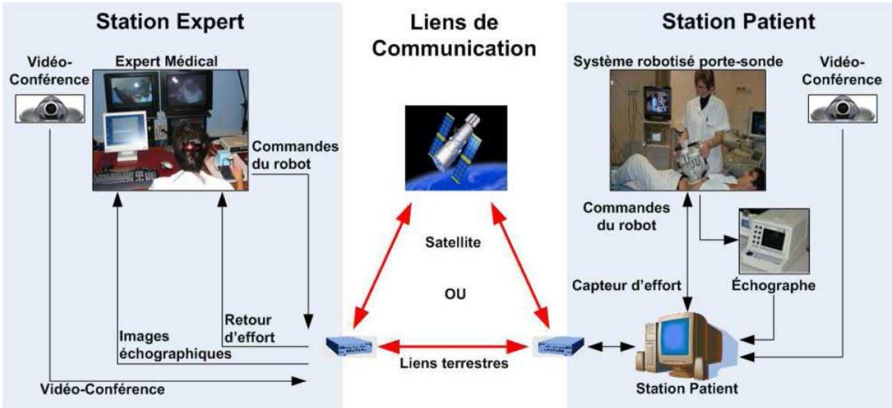


Figure 4: The general scheme of telechography global robotized

1.3.1.1 *The expert site*

The expert station is the site where the medical expert is. This site is composed of:

1. A video conference system allows the medical specialist to drive the assistant in the positioning of robot, to converse with the patient and to receive the ultrasound image. The expert drives the assistant to position the robot in the interesting part of the patient's body.

2. An input instrument called the fictive probe allows to control the robot in the patient's site. This probe looks like real ultrasound probe, it records the medical expert's movements in order to they are reproduced from end effector of robot on the patient's site. This probe can be: passive, it can record only the expert's movements; pseudo-haptic, it has a spring to simulate the rigidity of the patient's body; fully-haptic, the probe is actioned from a motor to render the efforts equal to the efforts generated by the ultrasound probe on the patient's skin. The type of probe is shown in the fig.5.

3. A graphic user interface allows the connection the fictive probe with the remote robot and it offers different functions; for example when the expert finds a interesting point, he can chose to block the robot in the current position and poses his probe to study more precisely the ultrasound image; furthermore, some interface allow to reposition the end-effector of robot in the initial position when he decides to study a different organ.

During the echographic exam, the medical expert based on the echographic images that visualizes with the help of the videoconference

(a) *The passive probe.* (b) *The pseudo-haptic probe.* (c) *The haptic probe.*

Figure 5: The various type of probes

system, the haptic feedback provided by the sensor (when permitted) and knowledge of the target organ for moving the probe fictitious to change the positioning of the probe effect on the patient's body.

1.3.1.2 *The patient site*

The patient site is the place where the robot holds the real bring probe, with the its interfaces of control, the ultrasound device, the videoconferencing system, the medical assistant and the patient. The robot is the central element of the patient site, it produces all the movements of the expert and orients the probe on the patient's body, performing the echographic exam. After the prototype, the robot can be endowed of a force sensor to measure the force exerted by the ultrasound probe on the patient's skin, this information is sent to expert site for the elaboration of the diagnosis. The echographic images are sent to expert site through the videoconferency system. A medical assistant stands next to the patient during the tele-echography exam. He takes care to position the patient, to apply the echographic gel on the area of interest, to position the robot and he follows the instructions of medical expert, and he has to ensure a good contact between the robot and the patient's skin.

1.3.1.3 *The communication line*

The expert site and the patient site can interact through a communication link. The tele-echography system are not limited to a only type of communication connection. If need, it can use the networks line LAN (Local Area Network) or WLAN (Wireless LAN) in the case of local tele-echography, Internet or ISDN(Integrates Services Digital Network) like the remote part of tele-echography, or the 3G networks or satellite systems for the mobile tele-echography systems. However, whatever is the kind of network, the exchange data between the two sites are numerous. These data exchange generates fluxes like:

1. A bidirectional audio-video flux utilized for the video conference between the expert and the patient.
2. A flux from the expert site to patient site for the transmission of commands to robot in order to reproduce the expert's action.
3. A flux from the patient site to expert site to return the proprioceptive sensor data of the robot (real position of the probe), and the return of the force sensor that allows the force feedback to the medical expert.

1.4 THE HAPTIC DEVICE

Position/Velocity and force information are exchanged between the user and the device. A haptic system is mainly composed by three different elements: the human operator, the haptic device, the controller of the device. In detail:

1. The human operator interacts with the patient's site handing the haptic device and receiving force feedback and, eventually, visual feedback from it; he is the final user of the whole system and everything is built in strictly accordance to the tasks he has to carry out.
2. The haptic device; main property of haptic device is transparency, that is his capability of good rendering of the reality being reproduced.
3. The controller is in charge of linking, in some way, the haptic device and the patient's site since it applies the force and the torques computed via simulation of device, it sends the exchanged interaction with the user to the simulator and, basically, it is responsible of the correct perception of the user. The control of the force applied to device is a crucial part of the scheme, since it has to let such a device follow the software simulation of the virtual reality to be implemented and so let the user perceive correctly the virtual environment.

In the next chapter it will be take in consideration the two haptic probes designed and used for the project Prosit.

1.5 THE CONSTRAINTS OF THE SYSTEM

There are many type of constraints: the mains are the mechanical constraints and the system constraints.

1.5.1 *Mechanical constraints of the probe in the robot's site*

Numerous meetings with medical experts on the characterization of the gesture of the medical expert for an ultrasound examination, helped define the main characteristics of the movement to be performed by the robot probe holder to reproduce the actions of the medical expert to distance. These characteristics have led to the development of the following specifications:

- The robot probe holder must be compact and lightweight to be placed on the patient's body without that the latter feels uncomfort, and be easily transportable in the site of intervention and easily manipulated during the examination ultrasound.

- The robot must be able to perform different types of ultrasound probes, but the shape of the ultrasound probe depends on the desired examination. To satisfy this constraint, the end-effector of the robot (mechanical part which contains the probe) must be sufficiently flexible to be able to accept these types of ultrasound probe.

- To follow the actions of the medical expert the end-effector of the robot must perform a spherical motion around a point of contact of the probe / skin and must maintain this contact during the examination. The effector must, in his work area, access to all the guidelines imposed by the expert, and must be able to perform a rotation around its axis of symmetry of at least one turn. The minimum angle that the probe must reach compared to normal skin is 35°. However, the maximum inclination that the robot should reach is 60°inclination; she is considered more than adequate.

- Moving the point of contact probe / skin is not required for organ research. This change can be offset by a repositioning of the robot, made by the assistant on the body of the patient. However, if the robot makes it possible to move the point of contact probe / skin in a circular area 25 mm radius, a maximum inclination the axis of the probe 35° is then sufficient to drive the ultrasound act.

- The sensor must be able to move along its axis of symmetry, to maintain a good contact probe / skin to achieve good ultrasound images. The amplitude of this movement is limited to 40 mm for safety reasons. In addition, the maximum permissible force bearing on the skin of the patient is about 20N.

- The maximum permissible speed for a displacement of the probe are:
    1. The maximum linear velocity of the point contact probe / skin is $30\frac{mm}{s}$;
    2. The maximum angular speed of displacement of the probe is $1\frac{rad}{s}$;
- This allows to set specific constraints to meet the mechanical architecture of the robotic probe holder. However, this specification describes only the ones mechanical, the use of this robot in a teleoperation system involves the constraints of new communication links. These additional constraints are defined in the next paragraph.

1.5.2 *Constraints of teleoperations system*

The requirements imposed previously allow to define the mechanicals architecture of the robot. For the remote command there are new constraints. The use of communication link between the expert site and the patient site can cause delays in the transmission and the possible loses of data that can alter the control of robot. These delay times, depend the kind of network utilized, vary from some tens of milliseconds for the fast networks(LAN o WLAN) to more of one second for the slow networks(3G or satellite). The expert can adjust the control of robot when the delay times are fix, however, he becomes unable to take into account variations of the delay. These variations can generate instability in the behaviour of the robotic system and therefore the deterioration of the trajectory generated by the medical expert. The medical use of the robot brings another constraint: the mechatronic system have to preserve the patient safety. Unlike the control of the velocity normally apply to the robot for the teleoperation , it's preferable to control the position of the robot. In case of an interruption of the communication connection or in case of lost data, the effector of the robot remain in the last received position information avoiding unwanted behaviour. Ie, it need that the probe transported on the robot reproduces perfectly the trajectory imposed from the medical expert manipulating the fictitious probe. Moreover, a goal of the tele-echography system is to consent the return of force to the expert, so he can feel the exerted forces by the ultrasound probe on the patient's skin, giving the impression of manipulate the probe directly on the patient. Hence, the force felt by the expert should be as close as possible to the effective force applied on the patient's body, giving optimal conditions for the medical expert's exam.

1.6 THE TELEOPERATION WITH THE FORCE FEEDBACK

Beyond the tele-echography, the robotic systems for teleoperations are developed in different sectors and applications. The teleoperation robots have been developed to monitor the actions of drones for the terrestrial exploration, for the immersion or in air. Moreover, the teleoperated robots allow to execute actions in place no adapted to human survival or to improve the humans action and his comfort. Finally, the teleoperation robots can be used to manipulate objects at a different scale than that of the operator. In the medical robot context , the sensation of the feedback tactile and force are the informations that the expert can request to better manage his gesture. Appreciate the contact with different type of organs improves the work environment and promotes the medical remote intervention. *A teleoperation system is called bilateral if the operator feel the physical interactions between the robot slave and the remote environment, usually, through a haptic instrument, allowing a better control of the remote robot.*

1.6.1 *General presentation*

1.6.1.1 *Definitions*

The principle of the bilateral control is control the mechanical contact between the robot and the remote environment that it's often not characterized. It allows to operator to be included in the control loop; he masters the control force that wants to apply on the robot with respect to the force he receives. The fig.6 shows a block diagram of a bilateral teleoperation system. The main components of this system are: the operator, master and slave, the remote environment and the communications link. All these elements are defined in the next paragraphs.

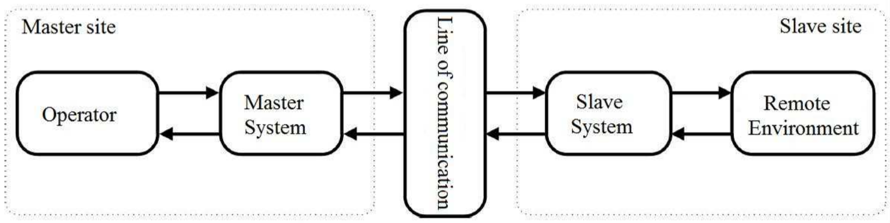


Figure 6: The scheme of tele echography system

The *operator* is the user that controls the slave robot, this one interacts with the remote environment. The actions of the operator propagates through the entire command "chain" to distance. He feels in the form of a force feedback the interactions between the slave robot with the its environment. The *master system* is composed of the main interface and its controller, the master system allows the operator to feel the interaction between the slave robot and its environment. The main interface must necessarily be of tactile type to provide a force feedback. Using these physical sensations, the operator manipulates the interface like if he would work directly on the remote environment. The master interface can be able to reproduce a tactile feedback (generally that corresponds at a return of force) in three spatial directions that they the operator to feel all the remote interactions. This interface uses the master controller the provide to operator the best perception of interaction of the remote environment. This control is a software with the control law and the type depends on the type of bilateral control that it is applied; this choice is determinant because it is in grand part responsible of the performances(transparency / stability) of the teleoperation system. As already mentioned the nature of *transmission line* leads directly to transmission delay times, via communication like the WLAN or LAN induces a low delays(<100 ms), while the communication connections like the 3G or via satellite induces a high delays(>1 s). In some case they could be fixed or variable, notes or estimates, they can influence the performances (transparency / stability) the teleoperation system. The *slave system* is composed of the slave manipulator and its controller. The role of slave manipulator is to replicate the operator movements and to interact with the remote environment of the operator; it has sensors in order to provide all the informations necessary from manipulator to operator. The slave controller is also associated at a type of software with a control law. It calculate the velocity and the torque of both robot actuators in order to reproduce as faithfully as possible the movements imposed by the operator. This controller has to be possibly sensible respect the delays time about the the communication connections and the remote environmental characteristics. The remote environment is all the external system of the slave system and all things that interact with it. There are different type of environment:

- They can be "structured", ie that is composed of objects whose the position and the nature are known.
- "Partially known", some of his characteristics are known, but it doesn't consent a complete comprehension of context.
- "Totally unknown".

Like part of a bilateral control system, in addition there is also the goal to allow the force feedback to the operator level; the bilateral control has to maintain the stability of the actioned remote system, ensuring the best possible transparency. The warranty of these properties depend of the nature of the master and slave controllers used and their consider also the delays of transmission. The master and slave controllers are realized in order to guarantee the stability of the system of operation. The transparency expresses the capacity of the system to render to the operator the mechanicals / physical interactions between the slave manipulator and the remote system. Concretely, a teleoperation system is said to be transparent if the resistance that the operator feels, when generates a X displacement, is equal to the resistance of the remote system in contact with the environment [10]. So, the operator has the impression to realize his work directly on the remote system making the teleoperation system totally transparent. For the mathematical model the first schemes are through a quadrupole: in the fig.7 the xh e Fh are the position and applied force from the operator with their input impedance Zh; while the xe and Fe are the position and the force applied on the manipulator system with their output impedance Ze[9][16][6]. This approach allow to realize the system model of teleoperation with the her matrix of impedance realized with the force and the position[16]; after it's described the hybrid matrix. The impedance matrix is rep-

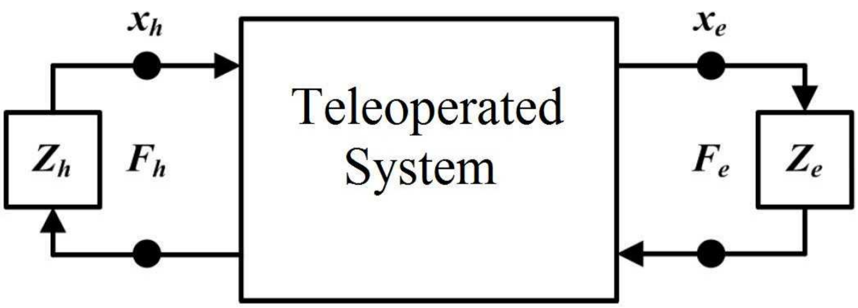


Figure 7: The mathematical model

resented considering that the position x_h and xe are the inputs of quadrupole, a represented can connect directly the position and the force from the next matrix:

$$\begin{bmatrix} F_h(s) \\ F_e(s) \end{bmatrix} = \begin{bmatrix} z_{11}(s) & z_{12}(s) \\ z_{21}(s) & z_{22}(s) \end{bmatrix} \begin{bmatrix} x_h(s) \\ x_e(s) \end{bmatrix} \tag{1}$$

The parameters z are the components of the impedance matrix of the teleoperation system and they depend on the impedances which characterize the master and slave system and their controller. The hybrid matrix[9] is used if the slave system is provided of force sensor because it's possible model that system in this mode:

$$\begin{bmatrix} F_h(s) \\ x_e(s) \end{bmatrix} = \begin{bmatrix} h_{11}(s) & h_{12}(s) \\ h_{21}(s) & h_{22}(s) \end{bmatrix} \begin{bmatrix} x_h(s) \\ F_e(s) \end{bmatrix} \tag{2}$$

The parameters of this matrix have a physical interpretation. The parameters h_{11} and h_{22} represent the impedance of the master system and the admittance of the slave system; the parameter h_{12} represents respectively the capacity of the teleoperation system to transmit the applied force from the slave system to remote system, while the parameter h_{21} represents the capacity of the slave manipulator to follow the movement imposed from the operator.

1.7 GLOBAL ARCHITECTURE OF ROBOTS PROSIT

The realization of robots PROSIT requires to define first of all the global platform of system, like that the develop of logical on this architecture in function of their roles[15]. In effect, the architecture is used like the "main element" of all system; therefore its implementation requires a clear vision of the system in its whole. For this reason the architectures in the next fig.8 has been defined after many discussions between the partners of the subject. The architectures of robots

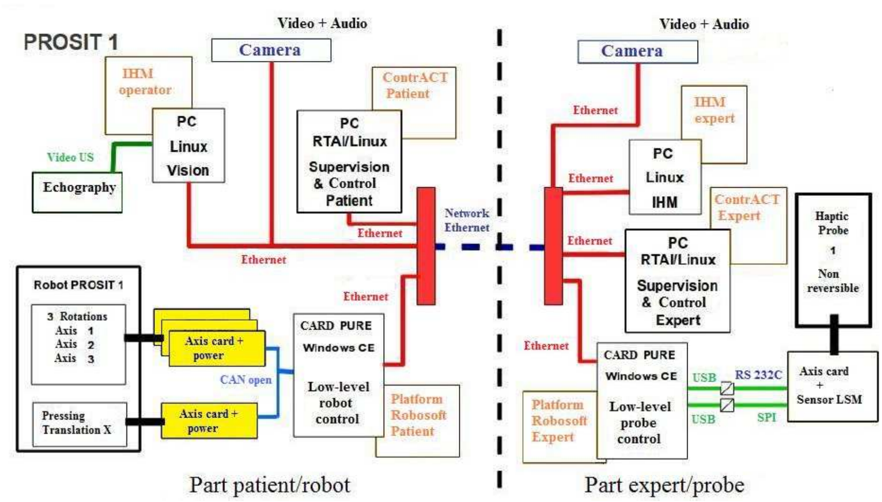


Figure 8: The architecture of robot Prosit 1

in the project Prosit 1 and Prosit 2 are the same. The patient's site like the expert's site has three calculation units very useful: hosting an architecture of control and supervision Contract (PcLinux/RTA1), hosting a GUI and algorithms related to vision (Linux PC) and a controller that supports low-level (pure control board Windows) of the material (rating probe expert and patient side robot). In each site(patient and expert) these three calculation units are connected between them through a network ethernet switchable via a administrable switch. This switch is also connected to the internet network through the different links (wired, satellite, GSM, etc.), this one allows to conserve the homogeneity of the mechanism of local communication(i.e. of the different calculation units) regardless of Internet connection choice. The advantage of such a connection is also to allow each partner to clearly identify the units of calculation on which it operates and thus facilitate the development of software patches and integration. The partner ROBOSOFT is responsible to implement for the low-level controllers of the haptic probe (expert site) and the robot itself (patient site). This controllers are cards "PURE" in which there are the WINDOWS CE systems that have logical platforms of ROBOSOFT. The latter is responsible to implement the interactions with the control devices, the implementation low-level of this devices, and the UDP server allows this external units to emit exploited commands in the enslavement and to recover the current measures. In the patient site the platform robosoft communicates with the axis board that command the different degree of freedom of robot via bus CAN Open. The expert site communicates with the robot site with the axis board and the sensor of the haptic probe via different USB line (for Prosit 1 and 2: it commands and measures the axis Z via adaptor USB/series and recovers the gyroscopic and accelerometer measures via a line USB/SPI. For prosit 2: there is a read of optic sensor via a USB line that it is type system of user counting). The partner INRIA Rennes and Prisme laboratory are responsible to implement the PC vision(patient site) and GUI(expert site), the work on the GUI in the patient and expert site are the responsibility of PRISME and the work on the algorithms of vision, the reconstruction 3D of the image and the different treatments on the image(expert site) are the responsibility of the INRIA Rennes. The GUI have to include also the flux video of environmental(expert site and patient site), the GUI expert site must have in more the flux video for the echography images that comes from patient site. We consider that the camera has its own occupational background video server and are directly connected to the switch internet, while the ultrasound system will be connected via an analog way to a capture card on the PC vision, the latter one has also the task of broadcast video streams via its own ethernet connection. In according to the technological possibility of echography, it will broadcast directly via an ethernet connection its video stream, the PC Vision did not achieve this diffusion. The partners LIRMM and PRISME are responsible for implementing the pc loads to host architectures Contract for supervision and control of real-time system. LIRMM does the development of Middleware "contract" and the management of wider communication (communication "low level" with other computing units of local networks in order to interact with the hardware, the GUI and the algorithms vision, communication and management of internet link of teleoperation and communication dedicated to monitoring) and Prisme handles the implementation of most control schemes envisaged. The implementation of supervisors including both aspects has control and supervision, it will be managed jointly by the two partners. The entire communication will be via the use of network sockets: UDP for local synchronous (ie periodicals, including the interaction between the vision algorithms and the PC hosting architecture Contract, and the interaction between the latter and the PURE card controller hosting the low Level) and the link of teleoperation; TCP for local asynchronous involved in sending/receiving commands/events (Vision or between PC and PC GUI Contract) and the communication link dedicated to monitoring(another chapteur not interesting). It has been described a general structure for the architecture Prosit 1 and 2; this document will talk about particularly on the graphic user interface (GUI) for the patient site. Therefore it's important to describe how the GUI and the PC contract interact:

- The manage of GUI is supported for two types of modules that communicate with the PC GUI in the same mode of the PC Vision. The synchronous modules guarantees the periodic exchanges(UDP) with the PC GUI for the transmission of detected measures on the robot and the transmissions via internet to PC Contract Expert. So the PC ContrACT Expert allows to GUI of update the his data on the state of the robot or on the state of the haptic probe. A asynchronous modules guarantees all the exchanges (TCP) related to the reception of the order from the IHM and the emissions of the events to IHM. This asynchronous modules routes the messages to the global supervisory of PC ContrACT Expert then will activate the correct operating mode.

1.8 THE ARCHITECTURE OF BILATERAL CONTROL IN THE ROBOT PROTECH

Noting that only the passive bilateral architectures guarantee the control of stability system piloted to distance, independently of delays, the bilateral control architecture chosen for the chain of tele-echography has been developed to insure passivity. There are two types of architecture position/force for the Protech robot: an architecture based on the waves variable[7] and an architecture based on control of rigidity[18][17].

1.8.1 *The main goal of the system*

The main goals of the tele-echography systems are two: the transparency and the stability. The transparency is the most important and the most difficult to achieve. In particularly to respect the transparency of the system it has to verify these two equations:

$$X_{patient} = X_{expert} \tag{3}$$

$$F_{expert} = F_{patient} \tag{4}$$

In the eq.(3) the position that the expert applies (input of the system) has to be the same position that it's applied on the patient (output of the system); while in the eq.(4) the force that it's applied on the patient (input of the system) has to be the same position that the expert feels on the haptic probe. For this reason the control of the system is in position/force. The tele-echography system has to guarantee, at the same time, that for all delay times the system is transparent and stable. The two architectures have advantages and disadvantages.

1.8.2 *The architecture of wave variables*

The architecture position/force passive, that has been developed by Gwenaël CHARRON, is shown in the fig.9. The pattern is very well

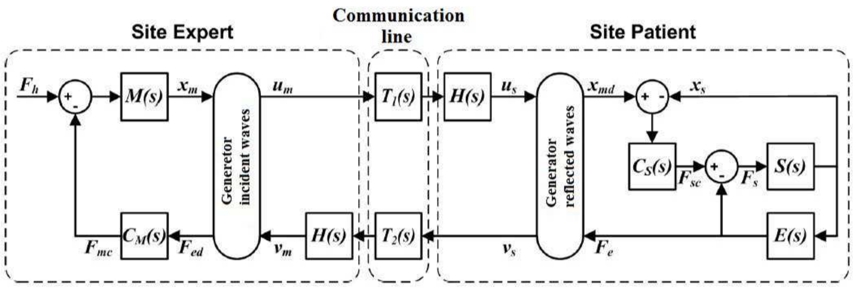


Figure 9: The architecture of robot Protech

engineered: the expert applies the reference of force to the probe that, after the comparison with the force feedback, pass on the probe model $M(s)$ and become the reference of position to apply at the slave robot. The lines of communication is represented from two blocks of variables delays $T_1(s)$ and $T_2(s)$. The reference of position x_{md} is compared to the position of the robot x_s, the error is processed from the control $C(s)$ to have an output the reference of force that, after that is compared with the force of the environment, it provide the exactly force to the slave robot. The slave robot is represented by the model $S(s)$ and the remote environment is represented by the model $E(s)$. Starting to the measure of the force F_e, and passing from the model of the waves variable and via the line of communication, it has the true feedback of force F_{mc}. This architecture is divided in three subsets:

- The expert site: it composed from the doctor; the haptic interface $M(s)$ to control the remote robot; the master controller $C_M(s)$, that provides a force feedback F_{mc} starting from the force F_{ed}; the generator of accident waves and the generator of force F_{ed}.

- The site patient: it composed from the slave robot $S(s)$ that interacts with the remote environment $E(s)$, the slave controller $C_S(s)$ and the generator of reflected wave from which the accident wave and from the environment force F_e generate the instructions of position x_{md} and the reflected wave v_s.

- Communication links: they allow the transmission of incident waves and the reflected waves from the expert site to patient site and vice-versa.

1.8.2.1 *The waves variable*

The next figure represent the generator of incident and reflective waves. Based on the formalism of the variable waves, this generator represents the bijective operator that allows to define the wave v and u from a linear combination of position x and force F. The incident

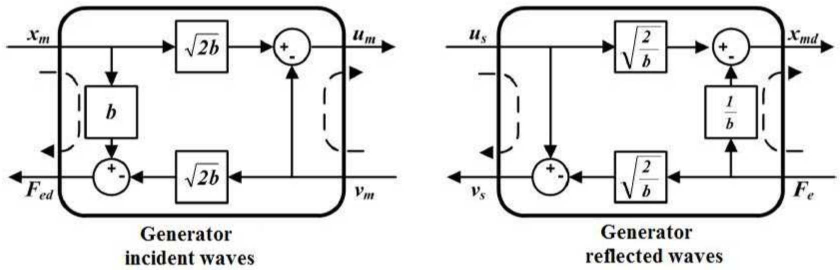


Figure 10: Architecture inside the wave generator

wave u and the reflected wave v are determined from the following equations:

$$u(t) = \frac{bx(t) + F(t)}{\sqrt{2b}} \tag{5}$$

$$v(t) = \frac{bx(t) - F(t)}{\sqrt{2b}} \tag{6}$$

where $u(t)$ and $v(t)$ are respectively the incident waves that transit from the expert site to the patient site and the reflect waves that transit from the patient site to the expert site; b is the characteristic impedance of the variable waves. The equations for the transmission wave are the following:

$$u_s(t) = u_m(t - T_1(t)) \tag{7}$$

$$u_m(t) = u_s(t - T_2(t)) \tag{8}$$

where $T_1(t)$ and $T_2(t)$ are respectively the delay times for incident reflected waves. The generator of incident waves allows the reception of the force at the expert site and as the incident wave:

$$F_{ed}(t) = bx_m(t) - \sqrt{2b}v_m(t) \tag{9}$$

$$u_m(t) = \sqrt{2b}x_m(t) - v_m(t) \tag{10}$$

The reflected wave generator delivers the incident wave:

$$x_{md}(t) = \sqrt{\frac{2}{b}}u_s(t) - \frac{1}{b}F_e(t) \tag{11}$$

$$v_s(t) = u_s(t) - \frac{2}{b}F_e(t) \tag{12}$$

From the eq.(8), (7) and (12) it's possible to define the force feedback F_{ed} felt at the expert on the base of environmental force F_e as follows:

$$F_{ed}(t) = bx_m(t) - \sqrt{2b}u_s(t - T_2(t)) + 2F_e(t - T_2(t)) \qquad (13)$$

This expression shows that the return of force Fed depends on other three components of force:

- The component $[bx_m(t)]$ is immediate, it is due to the reflections due to the transformation of the physical parameters in wave variables. For all movements of the robot, this conversion generates a immediate return of force in form of force bx_m like in the eq.(9). These reflections allow to the operator to feel immediately an apparent rigidity.
- The component $[\sqrt{2b}u_s(t - T_2(t))]$ is a reflection of the product during the conversion of physical quantities in waves. Similarly, when the incident wave arrives at the slave, of the information it contains is stored in the wave reflected v_s. Similarly, when the reflected wave arrives at the master, the wave v_m is reflected in wavelength u_m, according to the equation. The incident waves and reflected waves do not contain data only contain information useful but old artifacts that are caused by these different reflections. This phenomenon generates disorders that cause loss of performance because can cause oscillations on the system due to the non-energy dissipation in the system master or slave.
- The component $[2F_e(t - T_2(t))]$ contains information on the contact force between slave robot and its environment.

Among these three components, only the last contains the desired data (F_e). The first component allows the operator to feel a immediate force feedback through the characteristic impedance of the wave. However, the second component is a source of significant problems if it is not attenuated.[13] proposes two methods to eliminate the second component: either by performing impedance matching of the master and slave positions [11], or by filtering the incident and reflected waves to reduce the effect of reflections [12]. In tele-echography robot, the behavior of the operator (value Fh) and the value of the impedance of the environment can vary greatly. In this case, [14] advocate the use of wave filters. These wave filters can smooth the behaviour system independently of the task at hand. The insertion of these filters between master and slave sites does not affect the passivity, as wave variables are constructed so as not to be affected by the transmission delays or phase shifts. The only limitation to the use of these filters is the value of their gain, which must be less than or equal to 1. The use of these wave filters modifies the eq.(7) and the eq.(8) as follows:

$$\frac{1}{w_0}\frac{d}{dt}u_s(t) + u_s(t) = k_f u_m(t - T_1(t)) \tag{14}$$

$$\frac{1}{w_0}\frac{d}{dt}v_m(t) + v_m(t) = k_f u_s(t - T_2(t)) \tag{15}$$

with k_f the gain of the wave filter and $w_0 = 2\pi f_{cut}$, where f_{cut} is the cutoff frequency of the low-pass filter of first order $H(s)$.

1.8.3 *The control architecture in rigidity*

The other type of architecture has been developed by Juan Sebastian Sandoval. This architecture is based of Internal Model. The scheme of this architecture is shown in fig.11. The blocks of the scheme are

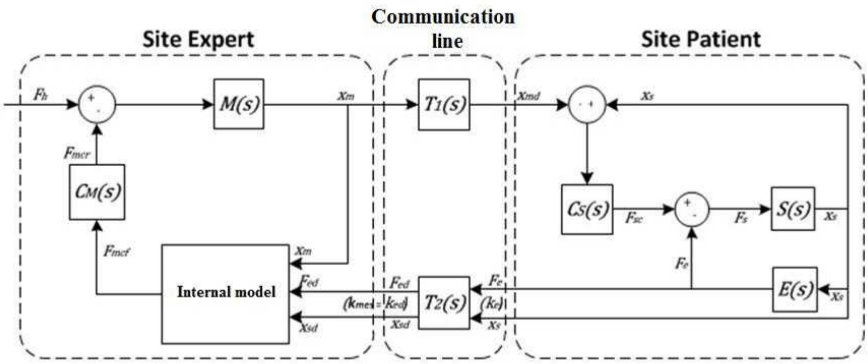


Figure 11: The control architecture in rigidity

equal to the blocks of the other scheme; it changes only the principle. The position x_s and the force F_e, after the passing from the lines of communication, become x_{sd} and F_{sd}. These last ones and the initial position x_m are the input to the model intern that it provide like the output the force F_{mcf}, that, after the filter $C_M(s)$, is the main return of force in the system. In this architecture there are two important concepts that will be deepen: the descriptions of the environment $E(s)$ and the Internal Model Principle of the system.

1.8.3.1 *The environment* $E(s)$ *and the Internal Model*

Inside the Internal Model, a method to characterize the remote environment is described considering the body characteristics and the breathing pattern of the patient. The complete system is shown in the fig. 12. To model the breathing a sinusoidal model is chosen: the feedback data F_{ed} and X_{sd} enter in the block "Caract" for the characterization the parameters; the outputs of the system are the average rigidity Kmoy, the frequency and amplitude of breath[20][19]. These

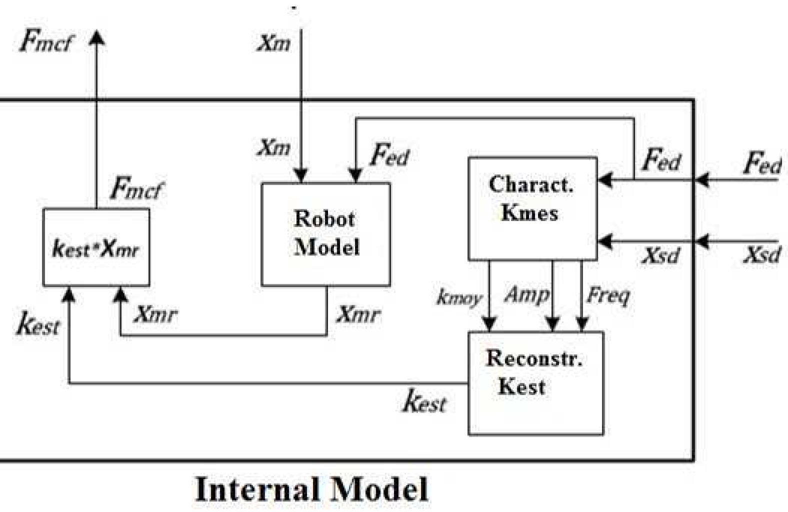


Figure 12: The internal model

parameters enter in the block "Reconstr" and, through the following equation, it returns as output the rigidity K_{est}.

$$K_{mes} = K_{moy} + Amp * \sin(2\pi\frac{t}{T} + \phi) \quad (16)$$

Morover there is a block that contains a mathematical model of the robot, this ones serves to the synchronization of the expert times. The inputs of this system are the position x_m and the force F_{ed}, and the output is the postion x_{mr}. The two outputs F_{ed} and x_{mr} serve to find the exact feedback force F_{mcf}:

$$F_{mcf} = K_{mes} * x_{mr} \quad (17)$$

1.9 THE WORK TEAMS

The works inside of the Prisme Laboratory are being developed in teams. In this moment there are two projects that have to be developed : the first one is called Protech, a project for the enterprise AdEchoTech, and the second one is called Prosit, for ANR national project. The teams is composed from many people. The main coordinator of the two projects is Pierre Vieyres, professor of IUT; he works since more ten years on the technology of the robot for the tele-echography systems; there are other two professor, Cyril Novales and Laurence Josserand ; the first ones works on the various topics of the project and the second ones work on the part of control of the robots. On these teleoperation system works many peolpe: Juan Sebastian Sandoval, a columbian student, Tao Li, a PhD chinese student and Chiccoli Marco. On these two projects works other people from other laboratories and universities. The work of the candidate intersects the two projects ; for the ProTech the projects the works consists mostly on the execution a remote tests for the robot, particularly it has been tested the communication of the robot for the transmission from the France to other country like Chypro, Perù and United States; this work will be described in the next chapters. For the Prosit project the works is centred mostly on the mechanical and electrical model of the probe and the GUI of the robot like already described previously; also this work will be described in the next chapters.

1.10 THE STRUCTURE OF THIS DOCUMENT

This document is divided in the following parts: in the first chapter talks about on the Blender software and the realization of the two probes for Prosit project with this program that will be used after in the GUI; the second chapter talks about on the GUI of the Prosit project: after a little introduction to the Qt software, it will be described the program with which the GUI has been developed and it will be described the mainly steps of this program; the third chapter talks about the mechanical and electrical model of the probe and the tests effectuated on this model to find the better response times; in the fourth chapter it will be described the many tests of communication that have been done with the robot Protech with others countries in the event WORTEX 2012; moreover, always in this chapter, it will be described the IEEE article that has been accepted for the conference in Cyprus called "BIBE 2012"; in the last chapter it will be described the conclusions of this works and future tasks that may be developed on this project.

2 THE DESIGN OF THE 3D MODEL OF TWO HAPTIC PROBES

2.1 INTRODUCTION

In this chapter will be described in the first moment the software "Blender"[3], with which ones has been designed the two haptic probes. After it will be described the functionality of two haptic probes for the project Prosit, particularly the haptic probe 1 and the haptic probe 2, and finally it will be described all steps to realize the two probes. The haptic probe is one of the main parts of the system; it is located in the expert station and it allows at the medical expert the control of the robot slave to distance. In recent years have been developed many types of probes. As already said in the par.1.3.1.1, the first type of probes were passive probe, where the expert could impose only the reference position; with the pass of the years the next step has been to add a spring in the probe, in order that the medical expert could feel a "fictitious" contrast of force while he is executing the tele-ecography. And finally, the last project is the haptic probe, that it allow at the expert medical, other at the done to impose the reference of position, to feel a true return of force, in order to help him to execute the tele-echography in the best mode possible. Of course, at the last type of probe serves more electronic and control devices. This document is not centred on this arguments, but the more informations can find in the ch.4.

2.2 THE SOFTWARE BLENDER

Blender is a free and open-source 3D computer graphics software product used for creating animated films, visual effects, interactive 3D applications or video games. Blender's features include 3D modeling, UV unwrapping, texturing, rigging and skinning, fluid and smoke simulation, particle simulation, animating,match moving, camera tracking, rendering, video editing and compositing. It also features a built-in game engine.

2.2.1 *History of Blender*

Blender was developed as an in-house application by the Dutch animation studio NeoGeo and Not a Number Technologies (NaN). It was primarily authored by Ton Roosendaal, who had previously written a ray tracer called Traces for Amiga in 1989. The name "Blender"

Figure 13: The software Blender

was inspired by a song by Yello, from the album Baby. Roosendaal founded NaN in June 1998 to further develop and distribute the program. The program was initially distributed as shareware until NaN went bankrupt in 2002. The creditors agreed to release Blender under the terms of the GNU General Public License, for a one-time payment of 100,000€. On July 18, 2002, a Blender funding campaign was started by Roosendaal in order to collect donations and on September 7, 2002 it was announced that enough funds had been collected and that the Blender source code would be released. Today, Blender is free, open-source software and is, apart from the two half-time employees and the two full-time employees of the Blender Institute, developed by the community. The Blender Foundation initially reserved the right to use dual licensing, so that, in addition to GNU GPL, Blender would have been available also under the "Blender License", which did not require disclosing source code but required payments to the Blender Foundation. However, this option was never exercised and was suspended indefinitely in 2005. Currently, Blender is solely available under GNU GPL.

2.2.2 *Features*

Blender has a relatively small installation size, of about 70 megabytes for builds and 115 megabytes for official releases. Official versions of the software are released for Linux, Mac OS X, Microsoft Windows, and FreeBSD.[4] Though it is often distributed without extensive example scenes found in some other programs, the software contains features that are characteristic of high-end 3D software. Among its capabilities are:

- Support for a variety of geometric primitives, including polygon meshes, fast subdivision surface modeling, Bezier curves, NURBS surfaces, metaballs, digital sculpting, outline font, and a new n-gon modeling system called B-mesh.
- Internal render engine with scanline ray tracing, indirect lighting, and ambient occlusion that can export in a wide variety of formats.
- Experimental new Cycles pathtracer render engine.
- Integration with a number of external render engines through plugins.
- Keyframed animation tools including inverse kinematics, armature (skeletal), hook, curve and lattice-based deformations, shape keys (morphing), non-linear animation, constraints, and vertex weighting.
- Simulation tools for Soft body dynamics including mesh collision detection, LBM fluid dynamics, smoke simulation, and Bullet rigid body dynamics.
- A particle system which includes support for particle-based hair.
- Modifiers to apply non-destructive effects.
- Python scripting for tool creation and prototyping, game logic, importing and/or exporting from other formats, task automation and custom tools.
- Basic non-linear video/audio editing.
- Game Blender, a sub-project, offers interactivity features such as collision detection, dynamics engine, and programmable logic. It also allows the creation of stand-alone, real-time applications ranging from architectural visualization to video game construction.
- A fully integrated node-based compositor within the rendering pipeline.
- Procedural and node-based textures, as well as texture painting, projective painting, vertex painting, and weight painting.
- Real time control during physics simulation and rendering.
- Camera and object tracking.

2.2.3 *User interface*

Blender has had a reputation of being difficult to learn for users accustomed to other 3D graphics software. Nearly every function has a direct keyboard shortcut and there can be several different shortcuts per key. Since Blender became free software, there has been effort to add comprehensive contextual menus as well as make the tool usage more logical and streamlined. There have also been efforts to visually enhance the user interface, with the introduction of color themes, transparent floating widgets, a new and improved object tree overview, and other small improvements (such as a color picker widget). Blender's user interface incorporates the following concepts:

2.2.3.1 *Editing modes*

The two primary modes of work are Object Mode and Edit Mode, which are toggled with the Tab key. Object mode is used to manipulate individual objects as a unit, while Edit mode is used to manipulate the actual object data. For example, Object Mode can be used to move, scale, and rotate entire polygon meshes, and Edit Mode can be used to manipulate the individual vertices of a single mesh. There are also several other modes, such as Vertex Paint, Weight Paint, and Sculpt Mode. The 2.45 release also had the UV Mapping Mode, but it was merged with the Edit Mode in 2.46 Release Candidate 1.

2.2.3.2 *Hotkey utilization*

Most of the commands are accessible via hotkeys. Until the 2.x and especially the 2.3x versions, this was in fact the only way to give commands, and this was largely responsible for creating Blender's reputation as a difficult-to-learn program. The new versions have more comprehensive GUI menus.

2.2.3.3 *Numeric input*

Numeric buttons can be "dragged" to change their value directly without the need to aim at a particular widget, thus saving screen real estate and time. Both sliders and number buttons can be constrained to various step sizes with modifiers like the Ctrl and Shift keys. Python expressions can also be typed directly into number entry fields, allowing mathematical expressions to be used to specify values.

2.2.3.4 *Workspace management*

The Blender GUI is made up of one or more screens, each of which can be divided into sections and subsections that can be of any type of Blender's views or window-types. Each window-type's own GUI elements can be controlled with the same tools that manipulate 3D view. For example, one can zoom in and out of GUI-buttons in the same way one zooms in and out in the 3D viewport. The GUI viewport and screen layout is fully user-customizable. It is possible to set up the interface for specific tasks such as video editing or UV mapping or texturing by hiding features not utilized for the task. The user interface supports multiple monitors.

2.2.3.5 *Hardware requirements*

Blender has very low hardware requirements compared to other 3D suites. However, for advanced effects and high-poly editing, a more powerful system is needed.

2.2.4 *File format*

Blender features an internal file system that allows one to pack multiple scenes into a single file (called a ".blend" file).

- All of Blender's ".blend" files are forward, backward, and cross-platform compatible with other versions of Blender, with the exception of loading animations stored in post-2.5 files in Blender pre-2.5 (this is due to the reworked animation subsystem introduced in Blender 2.5 being inherently incompatible with older versions).
- Snapshot ".blend" files can be auto-saved periodically by the program, making it easier to survive a program crash.
- All scenes, objects, materials, textures, sounds, images, post-production effects for an entire animation can be stored in a single ".blend" file. Data loaded from external sources, such as images and sounds, can also be stored externally and referenced through either an absolute or relative pathname. Likewise, ".blend" files themselves can also be used as libraries of Blender assets.
- Interface configurations are retained in the ".blend" files, such that what you save is what you get upon load. This file can be stored as "user defaults" so this screen configuration, as well as all the objects stored in it, is used every time you load Blender.

The actual ".blend" file is similar to the EA Interchange File Format, starting with its own header (for example BLENDER$_v$248) that specifies the version, endianness and pointer size, followed by the file's DNA (a full specification of the data format used) and, finally, a collection of binary blocks storing actual data. Presence of the DNA block in .blend files means the format is self-descriptive and any software able to decode the DNA can read any .blend file, even if some fields or data block types may need to be ignored. Although it is relatively difficult to read and convert a ".blend" file to another format using external tools, there are several software packages able to do this, for example readblend. A wide variety of import/export scripts that extend Blender capabilities (accessing the object data via an internal API) make it possible to inter-operate with other 3D tools. Jeroen Bakker documented the Blender file format to allow inter-operation with other tooling. The document can be found at the The mystery of the blend website. A DNA structure browser is also available on this site. Blender organizes data as various kinds of "data blocks", such as Objects, Meshes, Lamps, Scenes, Materials, Images and so on. An object in Blender consists of multiple data blocks-for example, what the user would describe as a polygon mesh consists of at least an Object and a Mesh data block, and usually also a Material and many more, linked together. This allows various data blocks to refer to each other. There may be, for example, multiple Objects that refer to the same Mesh, allowing Blender to keep a single copy of the mesh data in memory, and making subsequent editing of the shared mesh result in shape changes in all Objects using this Mesh. This data-sharing approach is fundamental to Blender's philosophy and its user interface and can be applied to multiple data types. Objects, meshes, materials, textures etc. can also be linked to from other .blend files, allowing the use of .blend files as reusable resource libraries.

2.2.5 *Development*

Since the opening of the source, Blender has experienced significant refactoring of the initial codebase and major additions to its feature set. Recent improvements include an animation system refresh; a stack-based modifier system; an updated particle system (which can also be used to simulate hair and fur); fluid dynamics; soft-body dynamics; GLSL shaders support in the game engine; advanced UV unwrapping; a fully recoded render pipeline, allowing separate render passes and "render to texture"; node-based material editing and compositing; Projection painting. Part of these developments were fostered by Google's Summer of Code program, in which the Blender Foundation has participated since 2005. The current stable release version is 2.63a, the previous version was 2.62 and was released on February 16, 2012.[1] New features included:

- New user interface
- New animation system, which allows almost any value to be animated
- Re-written, Python 3.x scripting API[29]
- Smoke simulation
- Updated toolset, with improved implementation

- Approximate indirect lighting
- Volume rendering
- Ray tracing optimizations, rendering some scenes "up to 10x faster"
- Solidify modifier
- Sculpt brush and stroke upgrade
- Add-on system
- Custom keyboard shortcuts
- Spline IK
- Color management
- Fluid particles (smoothed-particle hydrodynamics)
- Ocean simulation
- Network rendering
- Cycles render engine
- Deep shadow maps[30]
- 3D audio and video
- Game engine navigation meshes
- Motion capture tools
- Collada integration
- Updated motion tracking
- Camera tracking
- New interactive Global Illumination GPU accelerated render engine (Cycles)

The main difference between 2.63 and 2.62 is the introduction of BMesh which allows for n sided polygons (ngons), as opposed to the previous limit of 4 vertices.

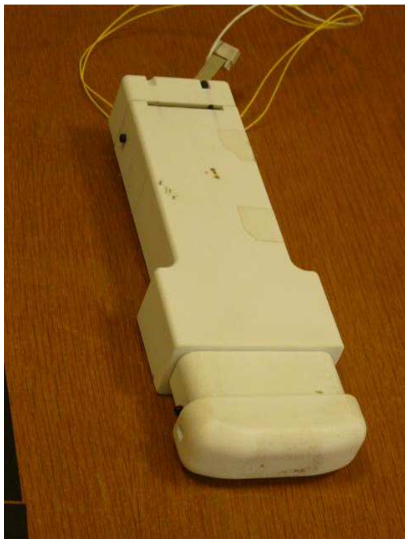
2.3 THE UTILITY OF TWO PROBES

The blender software will be used for the design of the two probes Prosit to have, as a final result, two objects 3D to insert in the GUI. But why are so important these two models of the probe? In this and in the next chapter it will answer at this question. While the tele-echography is in progress, as already said, the expert medical controls the probe and, at the same time, looks on the screen the ultrasound images. Moreover, on the screen where he looks the images, it will be added a new function. Knowing that the connection between the master and the slave can be of various types as ethernet, via satellite, 3G and other ones, it may have in the line of transmission the delay times variable, and, accordingly, the movement that the medical expert imposes with the probe is not the same movement that will be produced in the patient site with the robot slave. To help the doctor to execute the tele-echography therefore it will be created in the user interface a image 3D that represents the model of the probe and another one that represents the end-effector of the robot. Through the samples that arrive from the system, when the doctor begins to move the probe, also in the user interface the model of the probe moves in the same mode, and, when the robot begins to follow the reference of position, also the model of the end-effector of the robot moves. In this way the doctor can see the exact difference between the position of the probe and the position, delayed, of the end-effector of the robot.

2.4 THE TWO PROBES OF PORJECT PROSIT

In the project prosit have been developed two probes. The real first probe is shown in the fig.14 while the second probe is under development and therefore there isn't a real image yet.

(a) *The first view.*



(b) *The second view.*

Figure 14: The probe 1

2.5 THE BLENDER DESIGN OF TWO HAPTIC PROBES

The blender design of two probes is a work very complex if the designer doesn't know very well the environment of Blender. In fact, before to begin these projects, the candidate has spent a good time to study very well the software and all the the special features. In this paragraph it will be described before the probe 1 and after the probe 2.

2.5.1 *The probe Prosit 1*

The haptic probe Prosit 1 has a geometric structure more easy of the probe Prosit 2. The design is based on the model CAD created from Terence Essomba [8]. The model is shown in the fig.15. As you can see from the figure the probe is composed from three principal parts: the first part is the organ with which the medical expert keeps the haptic probe; the second part is composed from the end effector of the

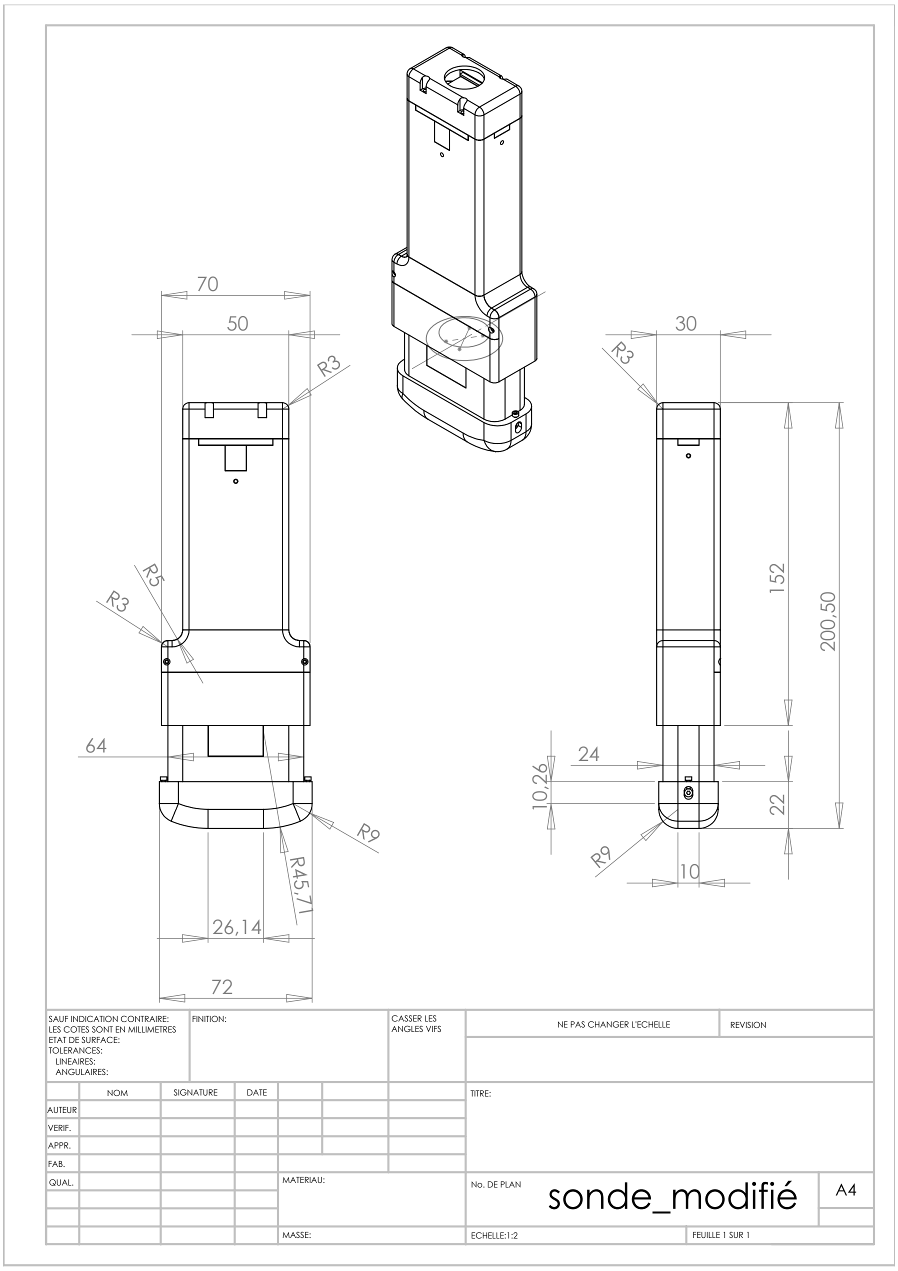


Figure 15: The model CAD of probe Prosit 1

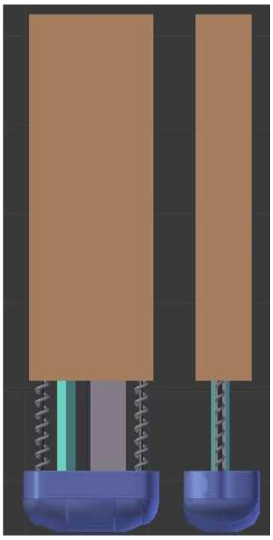
probe, the organ with which it can be possible capture the ultrasound image and the last part is the organ that attacks the first two parts. For the design of this probe has been used the meshes blender. The first part has been designed with the use of classic meshes like the cube resized; the difficult part has been to create the junction radius. For this purpose have been used the spheres and have been cut in quarters of spheres or hemispheres. One thing to be careful about is, as other software for the design 3D, that during the design of the object the construction have to be done in all the views, because it can be that in the frontal view the object seems in the first time perfect while in the lateral view all the parts of object are detached. With the same principal has been created the part of connection between the first part and the end-effector of the probe. The end effector of the probe has been created looking to reproduce for the most possible the same shape: for this reason have been put more spheres that, after the cut and the resize, have been connected to the final object. After to have created all the part of the final object, blender allows, through a little sequence, to compact all the parts of the object in order to obtain a unique part. The command is called "Join" and it finds under the the voice Object, at the bottom left of the screen. To utilize this process it need select the interesting parts together and to apply the command "Join". The final object is shown in fig.16; it is shown also the frontal view, the lateral view and the 3D view.

(a) *The frontal and lateral view.* (b) *The 3D view.*

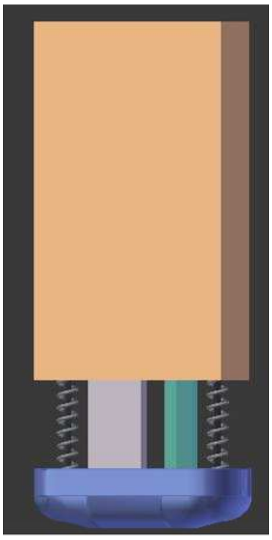
Figure 16: The views of haptic probe design

2.5.2 *The probe Prosit 2*

The haptic probe 2 is a new project and it has been created to better and to complete the probe 1. In this case there are not a CAD models but the project is based on the models provided from Laurence Nouaille, a professor in the IUT that she works on the Prosit project. For the blender model have been used always the meshes. The end-effector is the same one used in the first probe while the other parts of the object change. The first part is more easier than the first part of probe Prosit 1; in fact it has been realized only with a mesh cube resized because it has not curvatures. The connections between the other two parts is the critical section of this design because, respect to previously case, this probe has two springs. There are many modes to represented the spring in Blender; in particularly it is necessary to decide if it wants a static spring or dynamic spring. The dynamic spring is a spring that it can move during a animation for example; but in the our case it is not necessary this type of characterization. Therefore the type of spring that it has been implemented is a static spring, and it has been designed with the mesh "Circle". From this mesh, to arrive to a static spring, it is necessary to apply a "modify" that it is called "screw"; this type of modification serves to create the geometry of the spring, selecting also the options axis, screw offset, angle of revolution and the numbers of step in the revolutions. After to have completed the spring, the various parts of the probe 2 can be linked in a unique object with the same options utilized previously. The final result is shown in the fig17.



(a) *The frontal and lateral view.*



(b) *The 3D view.*

Figure 17: The model of probe Prosit 2

2.5.3 *The comparison with the end effector of the robot*

The final objective of this design is to allow to the medical expert to compare directly on his interface the movements of his probe and the movements of end-effector of the robot. For this reasons, always with the Blender software, has been designed also a end-effector of the robot. This design has been modified from a previous design and after it has been resized in order to have the same scale of the haptic probe. It very interesting to see the two probes together in the fig.18 and in the fig.19.

In this chapter, where have been shown the models of the two probes, is an introduction at the next chapter where it will be shown

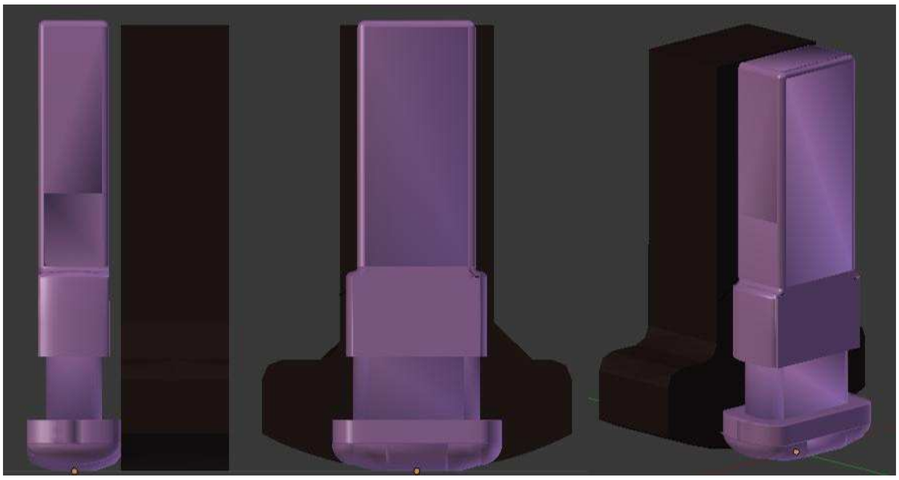


Figure 18: The comparison between the haptic probe Posit 1 and the end-effector

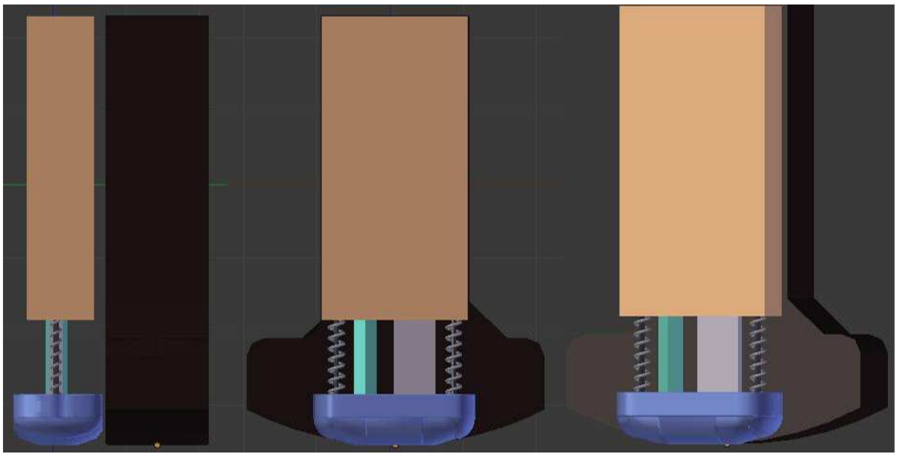


Figure 19: The comparison between the haptic probe Posit 2 and the end-effector

the complete description of the graphic user interface, particularly centred on the logical part of this one and on the part of the model of the probe.

3 THE GRAPHIC USER INTERFACE OF PROSIT SYSTEM

3.1 introduction

This chapter will talk about the graphical user interface in the Prosit project. As already said previously the human-robot interface is very important, in the same time, for the expert to execute the tele-ecography in the better mode possible,in fact he can see, in the same screen, the echography image to propose his diagnosis, and, at the same time, the position of his haptic probe respect to the position of end-effector of the robot; and for the expert engineer to set in the first time the all system and if, in second time, there will be unexpected operations and failures on the system, to reset the system in simple way. Why is important to see the exactly position of the haptic probe respect to the end-effector of the robot? As already said, to execute a tele-echography, the data have to be sent in the line of transmission clashing with, in some case, delay times that they can be fixed or variable, note or estimated, and, they can influence the performances (transparency / stability) of the teleoperation system. For this reason, when the medical expert moves his haptic probe, due to these delay times, the echography image that he is seeing may not correspond to the same point on the patient's body that instead he thinks to see. To help the medical expert to overcome this difficulty it has been created this interface of the probe, that it allows to compare the two positions of the haptic probe and the end-effector to the robot, in order to understand the exact point that he is seeing in that precise instant. On the project of the graphic user interface have worked more people, everyone has developed a part of the system: the main structure has been developed from Tao Li, a PhD student at the University of Rennes that works for the INRIA project and works also for the Prisme Laboratory in Bourges; the part of communication between the control boxes, the laptop and the robot has been developed from Robine Passama, a engineer from the Univeristy of Montpellier; and, finally, the candidate has developed the interface graphic of the haptic probe. In this chapter it will be described initially the software Qt[5], then it will be illustrated the structure of the graphic user interface and, finally, it will be explained how to interface the Blender models in the main screen of GUI and the mathematic operations to move the probes.

3.1.1 *The software Qt*

3.1.1.1 *What is Qt?*

Qt is a cross-platform application framework that is widely used for developing application software with a graphical user interface (GUI) (in which cases Qt is classified as a widget toolkit), and also used for developing non-GUI programs such as command-line tools and consoles for servers. Qt is most notably used in Autodesk Maya, The Foundry's Nuke, Adobe Photoshop Elements, OPIE, Skype, VLC media player, VirtualBox, and Mathematica, and by the European Space Agency, DreamWorks, Google, HP, KDE, Lucasfilm, Panasonic, Philips, Samsung, Siemens, Volvo, Walt Disney Animation Studios and Research In Motion. Sometimes Qt is asserted for being the toolkit of Opera web browser, but actually Opera uses internally a platform-independent custom toolkit, using Qt only as an interface to the Linux platform. Qt is developed by an open source project, the Qt Project, involving developers as individuals and from firms working to advance Qt, such as Nokia, Digia, and others. Before the launch of the Qt Project, it was produced by Nokia's Qt Development Frameworks division, which came into being after Nokia's acquisition of the Norwegian company Trolltech, the original producer of Qt. In February 2011 Nokia announced its decision to drop Symbian technologies and base their future smartphones on Microsoft platform instead. One month later Nokia announced the sale of Qt's commercial licensing and professional services to Digia, although Nokia will remain the main development force behind the framework. On May 9, it was announced on the Qt Labs website that the groundwork was being laid for the next major version of Qt, with the expectation that Qt 5 would be released in August 2012. Qt uses standard C++ but makes extensive use of a special code generator (called the Meta Object Compiler, or moc) together with several macros to enrich the language. Qt can also be used in several other programming languages via language bindings. It runs on the major desktop platforms and some of the mobile platforms. It has extensive internationalization support. Non-GUI features include SQL database access, XML parsing, thread management, network support, and a unified cross-platform application programming interface (API) for file handling. Distributed under the terms of the GNU Lesser General Public License (among others), Qt is free and open source software. All editions support many compilers, including the GCC C++ compiler and the Visual Studio suite.

3.1.1.2 *Platforms*

Qt works on the following platforms:

- **Windows** – Qt for Microsoft Windows.

- **Windows CE, Mobile** – Qt for Windows CE and Windows Mobile.
- **Symbian** – Qt for the Symbian platform. Qt is to replace Nokia's Avkon as the supported UI SDK for developing Symbian applications. The Qt for Symbian development group has many quality-controlled articles available.
- **OS X** – Qt for Apple OS X; supports applications on Cocoa.
- **X11** – Qt for X Window System (GNU/Linux, FreeBSD, HP-UX, Solaris, AIX, etc.)
- **Embedded Linux** – Qt for embedded platforms: personal digital assistant, smartphone, etc.
- **Maemo, MeeGo** – Qt for Maemo, merged with Moblin to MeeGo; many applications are already written for Maemo based on the prior Internet tablets; the Nokia N900 also supports Qt; the Forum Nokia Wiki has quality–controlled articles supporting Qt development; mhe Maemo has a development group on Forum Nokia Wiki at Forum Nokia Wiki Maemo.
- **Wayland** – Qt for Wayland display server. Qt applications can switch between graphical backends like X and Wayland at run time with the -platform command line option.

3.1.1.3 *External ports*

Since Nokia opened the Qt source code to the community on Gitorious various ports have been appearing. Here are some of them:

- **Qt for OpenSolaris** – Qt for OpenSolaris
- **Qt for Haiku** – Qt for Haiku
- **Qt for OS/2** – Qt for OS/2 eCS platform
- **Qt-iPhone** – experimental development of Qt for iPhone
- **Qt for webOS** – experimental development of Qt for webOS on Palm Pre
- **Qt for Amazon Kindle DX** – experimental development of Qt for Amazon Kindle DX Necessitas – Qt for Android
- **Qt for BlackBerry** – Qt for BlackBerry

3.1.1.4 *Editions*

There are three editions of Qt available on each of these platforms, namely:

- **GUI Framework** – commercial entry level GUI edition, stripped of network and database support (formerly named "Desktop Light")
- **Full Framework** – complete commercial edition
- **Open Source** – complete Open Source edition

Qt is available under the following copyright licenses:

- GNU LGPL 2.1 version with Qt special exception
- GNU GPL 3.0 version
- Commercial Developer License

3.1.1.5 *Current*

Trolltech released Qt 4.0 on June 28, 2005 and introduced five new technologies in the framework:

- **Tulip** A set of template container classes.
- **Interview** A model–view–controller architecture for item views.
- **Arthur** A 2D painting framework.
- **Scribe** A Unicode text renderer with a public API for performing low–level text layout.
- **MainWindow** A modern action–based main window, toolbar, menu, and docking architecture.

3.1.1.6 *Next*

The next major version of Qt will be Qt 5. It was originally expected to be released in June 2012 but was the release was delayed several times with November 2012 being now targeted. This new version will mark a major change of paradigm in the platform, with hardware-accelerated graphics, QML and JavaScript playing a major role. The traditional C++-only QWidgets will continue to be supported, but will not benefit from the performance improvements available through the new architecture. Qt5 is expected to bring significant improvements to the speed and ease of developing user interfaces. At the same time the framework has moved to open governance with ongoing development of Qt 5 taking place at qt-project.org. It is now possible for developers outside Nokia to submit patches and have them reviewed.

3.1.1.7 *Design*

The innovation of Qt when it was first released relied on a few key concepts.

3.1.1.8 *Modules*

1. Modules for general software development.
    - **QtCore** – contains core non-GUI classes, including the event loop and Qt's signal and slot mechanism, platform independent abstractions for Unicode, threads, mapped files, shared memory, regular expressions, and user and application settings.
    - **QtGui** – contains most GUI classes; including many table, tree and list classes based on model–view–controller design pattern; also provides sophisticated 2D canvas widget able to store thousands of items including ordinary widgets.
    - **QtMultimedia** – implements low-level multimedia functionality.
    - **QtNetwork** – contains classes for writing UDP and TCP clients and servers; implementing FTP and HTTP clients, supporting DNS lookups; network events are integrated with the event loop making it very easy to develop networked applications.
    - **QtOpenGL** – contains classes that enable the use of OpenGL in rendering 3D graphics.
    - **QtOpenVG** – a plugin that provides support for OpenVG painting.
    - **QtScript** – an ECMAScript-based scripting engine.
    - **QtScriptTools** – provides added components for applications using QtScript.
    - **QtSql** – contains classes that integrate with open-source and proprietary SQL databases. It includes editable data models for database tables that can be used with GUI classes. It also includes an implementation of SQLite.
    - **QtSvg** – contains classes for displaying the contents of SVG files. It supports the static features of SVG 1.2 Tiny.
    - **QtWebKit** – provides a WebKit-based layout engine as well as classes to render and interact with web content.
    - **QtXml** – implements SAX and DOM interfaces to Qt's XML parser.
    - **QtXmlPatterns** – provides support for XPath, XQuery, XSLT and XML Schema validation.

- **Phonon** – multimedia API, provides simple multimedia control.
- **Qt3Support** – provides classes that ease porting from Qt 3 to Qt 4.
- **Qt Declarative** module is a declarative framework for building fluid user interfaces in QML.

2. Modules for working with Qt's tools
   - **QtDesigner**
   - **QtUiTools**
   - **QtHelp**
   - **QtTest**
3. Modules for Unix developers
   - **QtDBus** – a library to perform inter-process communication via D-Bus protocol.
4. Modules for Windows developers
   - **QAxContainer** – an extension for accessing ActiveX controls and COM objects.
   - **QAxServer** – a static library to turn a standard Qt binary into a COM server.

3.1.1.9 *Use of native UI-rendering APIs*

Qt used to emulate the native look of its intended platforms, which occasionally led to slight discrepancies where that emulation was imperfect. Recent versions of Qt use the native style APIs of the different platforms to query the platform for the desired appearance of the Qt controls, and so do not suffer from such issues as much. One should note that on some platforms (such as MeeGo and KDE) Qt is the native API.

3.1.1.10 *Metaobject compiler*

The metaobject compiler, termed moc, is a tool that is run on the sources of a Qt program. It interprets certain macros from the C++ code as annotations, and uses them to generate added C++ code with Meta Information about the classes used in the program. This meta information is used by Qt to provide programming features not available natively in C++: the signal/slot system, introspection and asynchronous function calls.

3.1.1.11 *QtScript ECMAScript interpreter*

QtScript is a cross-platform toolkit that allows developers to make their Qt/C++ applications scriptable using an interpreted scripting language: Qt Script (based on ECMAScript/JavaScript). From Qt 4.3.0 onward, the scripting API, which is based on QSA, is integrated as a core part of Qt and is no longer a separate library.

3.1.1.12 *Tools*

- Qt Creator, a cross-platform IDE for C++ and QML qmake, a tool that automates the generation of Makefiles for development project across different platforms
- Qt Designer
- Qt Assistant
- Qt Linguist
- lupdate
- lrelease
- lconvert
- QVFb
- makeqpf
- uic (User Interface Compiler)
- rcc (Resource Compiler)
- qtconfig
- qconfig
- qtdemo
- qt3to4
- qdbusxml2cpp
- D-Bus Viewer
- Qt Visual Studio Add-in
- Qt Eclipse Integration
- Qt Simulator
- Nokia Smart Installer for Symbian
- qmlviewer
- Qt Quick, a QML based user interface development kit

3.1.2 *The structure of the GUI*

3.1.2.1 *The structure of the interface*

In this paragraph it will be shown the general structure of the graphic user interface, describing the vary parts of it and the its mode of functioning. The software Qt allows to develop the GUI in the two parts: one part graphic, where it is possible to chose the arrangement and the type of the images and buttons; and one part of programming, where it is possible to have the management thereof, for example active/disactive the button, to manage the active/disactive the image with the events. The GUI developed is composed from two subparts, as already said, one for the medical expert and one for the engineer expert; by the button in the top of interface it is possible to choose the page of interest like it's shown in the fig.20. The engineer part

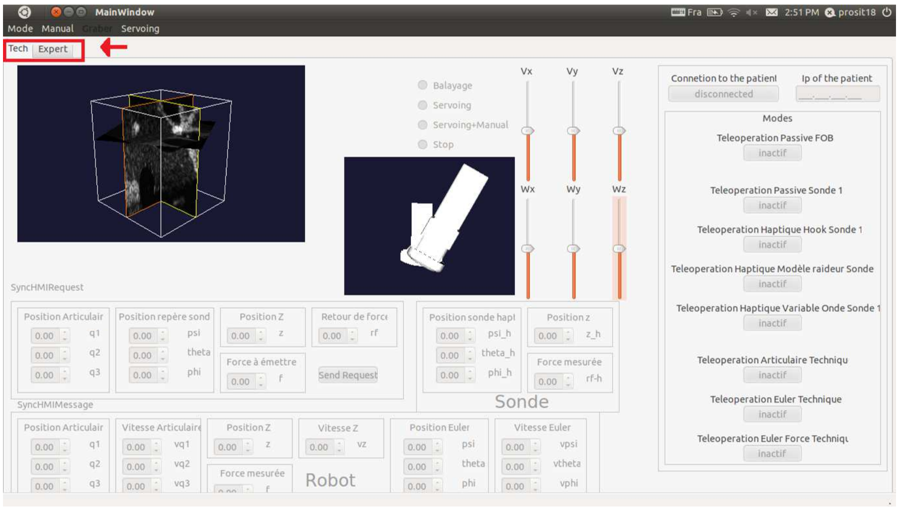


Figure 20: The graphic user interface: the choice between the expert window(medical) or technical window

is composed from more things because the engineer has to have all the useful informations to set the system in the correct mode. In this parts there are three images like it's shown in the fig.21: the ultrasound image, both in 3-D both in 2-D, and the interface of the probe. There are sliders to simulate the movement of the haptic probe (it

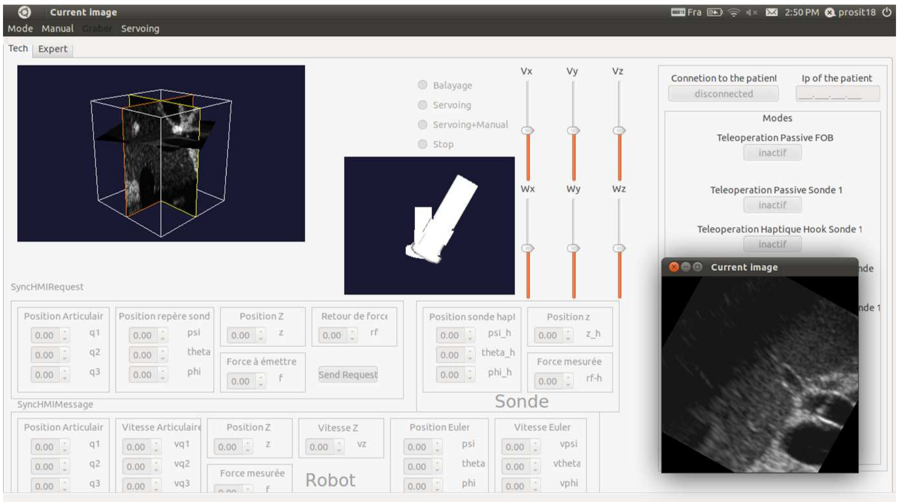


Figure 21: The technical window

will be described also in the next subparagraph) and in the bottom of the page there are all sampled values that the system Prosit provides, both for the probe and both for the robot. In the part of the probe are provided the position articular, the position of reference the probe, the translation z , the force emitted that it is the force measured in the patient site and return of force that is the force feedback in the expert site. Are also provided the orientation of probe, in Euler angles, the vertical position of the probe Z and the feedback force that it is transmitted on the probe. In the part of robot there are similar measures: there is again the articular position of the robot, there is the articular velocity of the robot, the vertical position Z of the end-effector, the measured force of the end-effector and the velocity of the end-effector. There are moreover the orientation and the speed of the end-effector in the Euler angles. In the right of screen the expert engineer can begin the connection by entering the IP address of the patient site and pressing on connect to the patient. As in the par.1.3.1.1 and in the par.1.8, it remembers that there are three types of probes and that there are more modes to control the system; here it is possible choose the mode of functioning according to the probe used and the system utilized to control the robot. The mode of functioning that it can choose are:

- Teleoperation passive FOB
- Teleoperation passive Probe 1
- Teleoperation Haptic hook Probe 1
- Teleoperation Haptic control stiffness
- Teleoperation Haptic variable wave Probe 1
- Teleoperation Articular Tecnical Control Mode
- Teleoperation Euler Tecnical Control Mode
- Teleoperation Euler Force Tecnical Control Mode

The medical part is composed from less things, in order to facilitate the medic's work: there are the ultrasound image and the interface of the probes. There is again the same part of the engineer part where he can connect the system easily and choose the mode of functioning.

3.1.2.2 *The interface logic*

The logical part of the GUI is very important. In the top of the interface it can choose between some options:

- **Mode**
- **Manual**
- **Graber**
- **Serving**

In the **Mode** it can choose vary options:

- **Start server**, to begin the teleoperation.
- **Stop server**, to stop the teleoperation.
- **Initialization**, to choose the format of ultrasound image and to initialize the position of the robot.
- **Simulator**, it is a mode of functioning very useful to test the system with the images of tests: in this mode it can move the model of probe with the sliders to try and to see if the model and the displacement of the probes is good.
- **Robot**, it is the real mode of functioning for the system.
- **About**, it is the mail address of GUI designer.
- **Exit**, to exit from the GUI.

The button **Manual** is activated only if in the mode simulator and it can choose vary options:

- **Manual**, to begin the test in mode simulator.
- **Stop**, to finish the test in mode simulator.

All the buttons have been designed in order to mutual exclusion their when the operator select a type of functioning: for example when the operator choose the simulator mode the button with the robot mode is obscured, the same thing between the start server—stop server and between also the manual—stop mode. In the simulator mode, to manually move the probe, it have been created a sliders and with their, it is possible to move and to rotate respect the three axes of x,y,z.

3.2 THE INTERFACE OF THE PROBES

To interface the probes it must insert the two models in an image of GUI. For this work, it is necessary to save the blender model like object file and after insert this object in the GUI by a function. The input that arrive from the system are the orientation of the haptic probe and of the end effector expressed as angles of Euler ZXZ; the idea is, as there is already a function that starting from the ZYZ angles returns the final orientation of the probe θ_u, to transform the matrix from ZXZ angles to ZYZ angles. Once it found the ZYZ rotations, it apply these angles to the function "TU vector" to find the orientation final. In the following there will be shown the procedure step by step. All the transformation matrices (special linear and orthogonal) can be expressed as the composition of three rotations ZYZ[21]. Put $H = Z * X * Z$, this matrix can be seen(where $c = \cos$ and $s = \sin$):

$$R_z(\phi)R_y(\theta)R_z(\gamma) = \begin{bmatrix} c\phi c\theta c\gamma - s\theta s\gamma & -c\phi c\theta s\gamma - s\phi c\gamma & c\phi s\theta \\ s\phi c\theta c\gamma + c\phi s\gamma & -s\phi c\theta c\gamma + c\phi c\gamma & s\phi s\theta \\ -s\theta c\gamma & -s\theta s\gamma & c\theta \end{bmatrix} \quad (18)$$

Now, it need find the angles of Euler ZYZ; there are two cases: the first ones where the solutions are unique and the second ones where the solutions are infinite. If the values of the matrix h_{31} and h_{32} are different to zero the solutions are uniques:

$$\phi = atan2(h_{23}; h_{13}) \quad (19)$$

$$\theta = \arccos(h_{33}) \quad (20)$$

$$\gamma = atan2(h_{32}; h_{31}) \quad (21)$$

In the other case, if the values of the matrix h_{31} and h_{32} are equal to zero, the solutions are infinite because the value of θ can be 0 or π. Depending on the choice of this free coordinate the matrix can change in two ways:

$$\theta = 0 \rightarrow H = R_z(\phi + \gamma) \quad (22)$$

$$\theta = \pi \rightarrow H = R_z(\phi - \gamma) \quad (23)$$

In the eq.(22) the matrix H is a easy rotation around the axis Z while in the eq.(23) the matrix H is a combination of one rotation around the axis Z and another ones around the axis X. In this work the solution chosen in the case with the infinite solutions is:

$$\theta = 0; \qquad \phi = \gamma = \arccos h_{11}/2; \quad (24)$$

The unique problem is that, in the case where the solutions are infinite, the function $acos$, unlike the function atan2, it doesn't take into account to the quadrant of the angle. Therefore, it has been added a little part to search the quadrant and to return the exact value starting to cosine and sine of the angle. The complete code is in appendixB. But which point of the haptic or of the end effector probe is moved? The reference is always in the contact part of the probe, but there is a difference between reference of the blender model and the reference of the Prosit system. In fact, the blender model has the reference with the z-axis downwards while in the system prosit one has the y-axis downwards. Therefore, to move the model of the probe, it is used a change of reference, composed by a rotation of -90 degree around x-axis and followed by a rotation of 90 degree around y-axis:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 90 & \sin 90 \\ 0 & \sin 90 & \cos 90 \end{bmatrix} \begin{bmatrix} \cos 90 & 0 & -\sin 90 \\ 0 & 1 & 0 \\ \sin 90 & 0 & \cos 90 \end{bmatrix} \quad (25)$$

Then, it remembers that to change the coordinate of the point in the new system of reference, it serves to have the inverse matrix that, in the case of the transformation matrix, is equal to transposed matrix:

$$\begin{bmatrix} x' & y' & z' \end{bmatrix} = T^t \begin{bmatrix} x & y & z \end{bmatrix} \tag{26}$$

3.3 THE TEST OF THE PROJECT

In the month of septembre 2012 the team PROSIT met at the laboratory Prisme to execute the test on the robot. The test have been effectuated with the robot PROSIT 2, after some problems met with the robot PROSIT 1. The architecture of the robot, as already explained in the par.1.7, is composed from many computer, each with different functions. All the code to allow the control and the communication in the tele-echography chain has been developed on the platform "Eclipse" and has been thought as a modular structure. Each module has its own function and communication between the modules is controlled by several supervisor modules. The communication between the expert site, the patient site and the computer vision has been done with the use of the socket in a asynchronous communication client / server. After to have restored these communication, where the candidate has participated actively, the tests are began. In fact, with the modular structure it's possible test the electronic card of the robot and the electronic card of the haptic probe, to see and to validate own functioning. This one has been the first step that the team has tested. Subsequently, the candidate has tested his work; the final result is very good. The movement of the two 3d models of the haptic probe and of the end-effector of the robot has not been tested because there were some problems downstream of this work: the work has been thought for the robot PROSIT 1 and therefore there were many things to change; moreover, the direct geometric model didn't response in correct mode, giving the Angle of Euler incorrectly; also, the connection between the expert site and the patient site had some problems of communication, particularly in the protocol TCP / UDP, but after, this problem has been solved. However, one good local test has been executed between the GUI and the robot PROSIT 2: it has been tested the correct connection with the GUI and the robot, and the correct orientation of the 3d model of the haptic probe driving the robot directly by the GUI with a specific tools created ad hoc. The test has been very good because the orientation of the 3d models and the orientation of the robot were the same. Moreover the candidate has done some confidential video of this test.

4 STUDY OF THE HAPTIC PROBE 2: DYNAMIC MODEL AND RESPONSE TIME

4.1 INTRODUCTION

In this chapter it will be shown a study of the second haptic probe for the project Prosit. As already said, one of the objectives of the tele-echography project is to ensure the return of force from the patient site; for this reason the haptic probe in the expert site must have a mechanical and electrical part able to reproduce this force. After a introduction on the mechanical and electrical structure of the probe 2, it will be studied two main aspects: the dynamic model and the response time of the probe. The dynamic model is designed with the software Matlab, particularly through the use of simulink, after a study of the dynamic equations that bind the force of the expert and the return the force from the patient site. In this model it will be done numerous observations to make the system as realistic as possible and to optimize the response time of the haptic probe.

4.1.1 *The mechanical model of the probe*

The probe has a specific mechanical system to transmit and to feel to the medical expert the return of the force that it represents the interactions between the echography probe on the robot slave and patient's body. The force in the patient site is captured through a force sensor and is sent to the medical expert via satellite or via ethernet. The slave force can be seen like the force that is proportional at the translation of the probe and the stiffness of the patient's body; this concept is shown in the following equation:

$$F_{slave} = -K_a z \tag{27}$$

Where:

- F_{salve} is the force that returns to expert site.
- K_a is a medium stiffness measured on the patient's body.
- z is the translation of the system.

The return of force is transmitted through a particular mechanical system. This system is composed by a torque motor, to generate the necessary torque to reproduce the exact force; a pair of pulleys, to convert the rotatory motion of the motor in the linear movement of

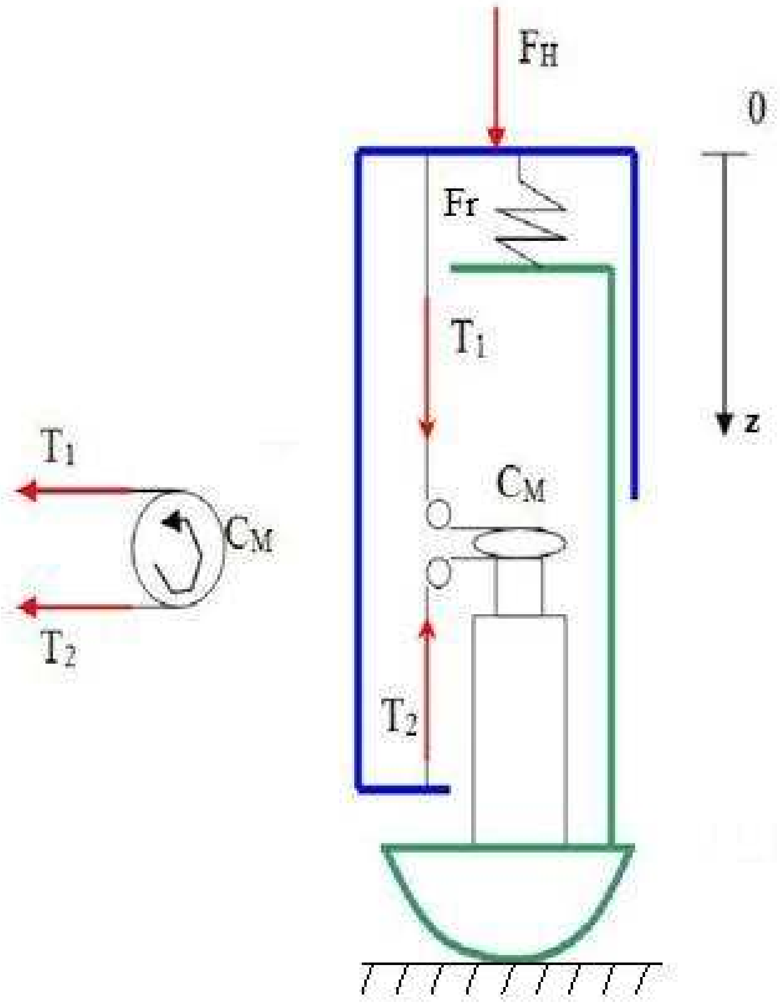


Figure 22: Model of the Haptic Probe

the haptic probe; a pair of wires, to pair the pulleys and the part mobile of the probe and a spring, to pair the fixed part and the mobile part of the haptic probe. In the fig.22 is shown the complete model. The force F_H is the force that the medical expert applies on the haptic probe to change the vertical position of the end-effector of the robot in order to view the better ultrasound image. This force is not predictable and it is not model; but in the study it has seen that the force is variable in a range to 0-9 N.

4.1.2 *The mechanical equation*

To find the equation of equilibrium it applies the fundamental principle of dynamics, considering the gravity forces negligible in front of the others forces applied and, in a first time, considering a perfect mechanism of transformation the movement(without losses). With:

- F_H: Force exerted by the expert on the probe.
- F_M: Resultant force exerted by the motor-pulley system on the mobile part of the probe.
- F_r: Force exerted by the spring on the mobile part of the probe.
- M: Mobile part mass with integrated inertia pulleys.

The equation of the dynamic equilibrium is:

$$F_H + F_M - F_r = M\ddot{z} \tag{28}$$

And:

$$F_M = T_1 - T_2 \rightarrow (T_1 - T_2)r = C_{pul} \tag{29}$$

Where:

- T_1, T_2: Tension forces in the Kevlar wire.
- C_{pul}: Pulley torque.
- K: Stiffness of the spring.
- r: Rolling up ray of the pulley.

If we consider the mechanical equation of the rotor, we can write:

$$C_{pul} = C_m\rho - J_{pul}\dot{\Omega} \tag{30}$$

$$J_{pul} = J\rho^2 \tag{31}$$

$$\Omega = \frac{\dot{z}}{r} \tag{32}$$

Where:

- C_M: Motor torque(including mechanical losses in the gear speed reducer).
- J_M: Inertia rotor.
- J_{pul}: Inertia rotor expressed on the pulley axis.
- ρ: Reduction of speed reducer.

The combination of the precedent equations gives us the relation between F_H, z and the motor torque C_M:

$$F_H + \frac{C_M}{r} - \frac{J\dot{\Omega}}{r} - Kz = M\ddot{z} \tag{33}$$

And, the speed can be seen as:

$$\Omega = \frac{\dot{z}}{r} \tag{34}$$

Therefore, it arrives the final equation:

$$F_H + \frac{C_M}{r} - \frac{J\ddot{z}}{r^2} - Kz = M\ddot{z} \tag{35}$$

To obtain the equilibrium dynamic between the slave and master force it obtain the equation of the slave force:

$$F_{slave} = \frac{C_M}{r} - \frac{J\ddot{z}}{r^2} - Kz \tag{36}$$

Another one objective is to calculate the necessary torque that the motor has to provide in order that the medical expert feels the same force feedback from the patient site:

$$C_M = rF_{slave} + \frac{J\ddot{z}}{r} + rKz \tag{37}$$

This equations will be used to build a dynamic model to simulate the system.

4.1.3 *The dynamic model of the system*

To build the dynamic model of the system it will be used the equations obtained in the last chapter. The idea is to obtain the acceleration of the system through the dynamic equilibrium between the slave and master force, assuming the master force an input of the system, and the slave force, as already said, proportional to the translation z of the system:

$$\ddot{z} = \frac{F_H + F_{slave}}{M} \tag{38}$$

and, having the acceleration of the system, it is possible to have, through two integrations, the speed and the position of the system. Having modeled the force slave like the eq.(27), the equation of the haptic probe can be seen like the second order system, with the adding of damping dynamic:

$$F_H = M\ddot{z} + K_v\dot{z} + K_a z \tag{39}$$

One time that it find the position, speed and acceleration of the system, it possible to obtain the necessary torque with the eq.(37). Assuming that the motor torque is directly proportional to the current of the motor through a torque constant:

$$C_M = K_t I \tag{40}$$

Then, the reference of the current of the motor will be:

$$I = \frac{C_M}{K_t} \tag{41}$$

After a regulation of the current loop of the motor it will be the torque wanted. To verify that the slave force that the medical expert feels from the motor action is the same calculated in the eq.(27) it is utilized previously in the eq.(36).

4.1.4 *The Direct Model of the Probe*

The model of the haptic probe is regulated by the eq.(38). Inserting the damping dynamic, it is possible see the second order system:

$$F_H = M\ddot{z} + K_v\dot{z} + K_a z \tag{42}$$

And the parameter of this system are:

$$w_n = \sqrt{\frac{K_a}{M}} \tag{43}$$

$$\zeta = \frac{K_v}{2\sqrt{K_a M}} \tag{44}$$

The scheme of this model is shown in the fig.23:

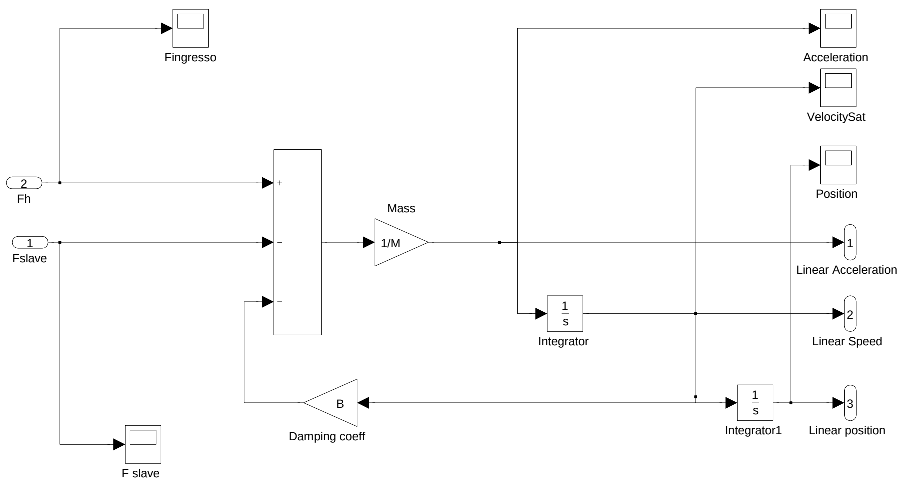


Figure 23: Direct model of the Haptic Probe

4.1.5 *The model of the motor*

In the first time, to test the functionality of the scheme, is used a scheme of the DC motor(fig.38). The input of the motor, to control the torque, is the current calculated in the eq.(41). This current, is compared with the other current calculated, and, passing for the controller, generates the exact voltage for the motor. After the pass of the voltage for the part electrical it will get the motor current:

$$I = \frac{V}{R + sL} \tag{45}$$

The current is then multiplied for the torque constant to obtain the exactly necessary torque that, passing for the mechanical load of the motor, gives in output the velocity of the motor.

$$s\theta_m = \frac{C_m}{B + sJ} \tag{46}$$

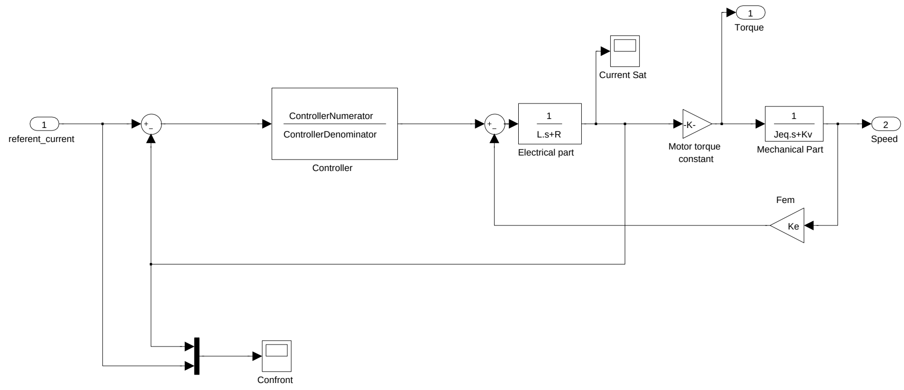


Figure 24: Model of the motor

4.1.5.1 *The controller of the current*

In this context, the output of the system is the torque calculated. To have a good tracking of the reference it is necessary to calculate a good controller of the system; the loop of the current is described by the following transfer function:

$$GE = \frac{\frac{1}{K_t}}{\frac{(R+sL)(B+sJ)}{K_t^2+1}}(B + s\frac{J}{K_t}) \tag{47}$$

With the parameter of the motor, the transfer function of the current has this Root locus and Bode plot like in the fig.25. To control this

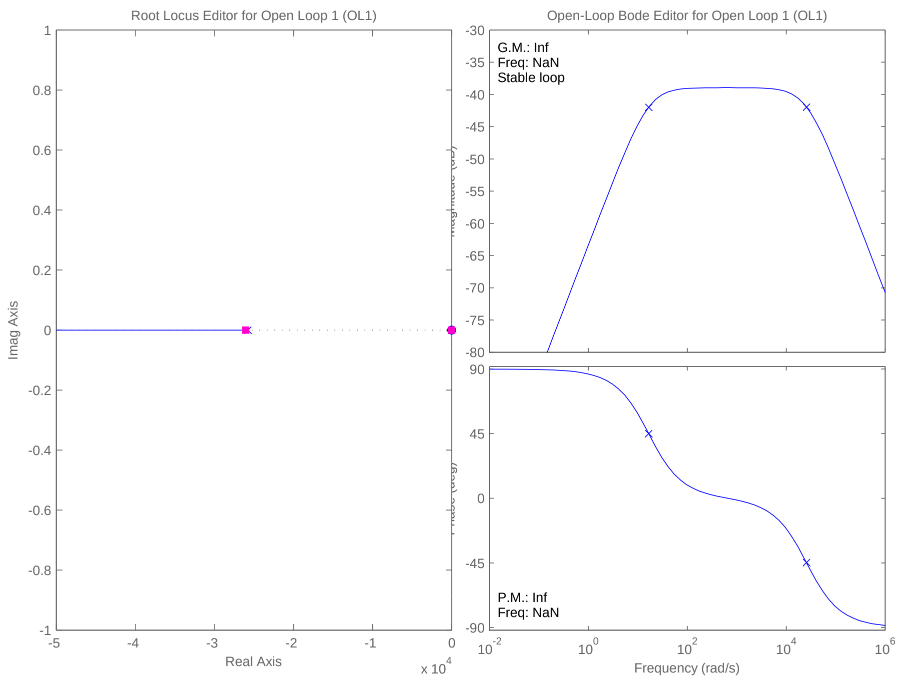


Figure 25: Root locus and Bode plot of the current loop

system it have been looked the two parameters:

- Phase Margin
- Bandwidth

The controller calculated is the followed:

$$C(S) = \frac{2.6e6}{s} \tag{48}$$

The results with this controller are very good and are showed the Root locus and Bode plot after to have inserted the controller in the chain(fig.26). With this controller the performance is very good: it is

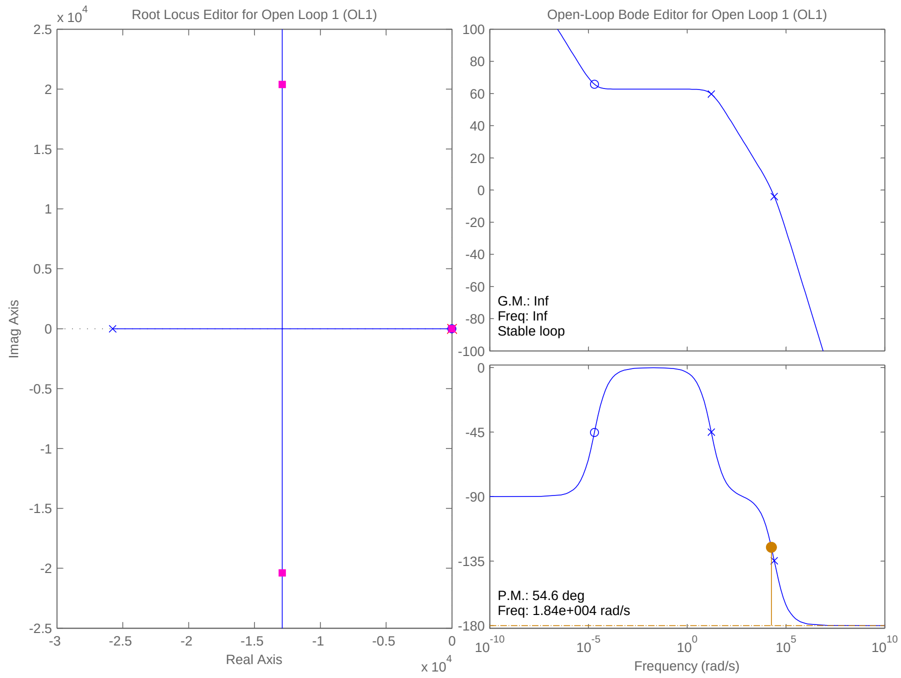


Figure 26: Root locus and Bode plot of the current loop more the controller

obtained a good phase margin, about 55 degree, and a good bandwidth, about $1,84e4 \frac{rad}{s}$. It is possible to test the loop of the motor applying a ideal reference of current and seeing the output of the system; the result is shown in the fig.27. The system follows the reference very well.

4.2 SIMULATION OF THE SYSTEM

4.2.1 *Size of the system*

For the simulation of the system it is necessary sizing the second order system in order to obtain a damped response and the fastest possible. Like in the equation (42) it possible see that, to obtain a good results, the system must be composed from two real poles and, moreover, these poles have to be in the high frequency. To satisfy the first condition it is necessary that the discriminant of the two

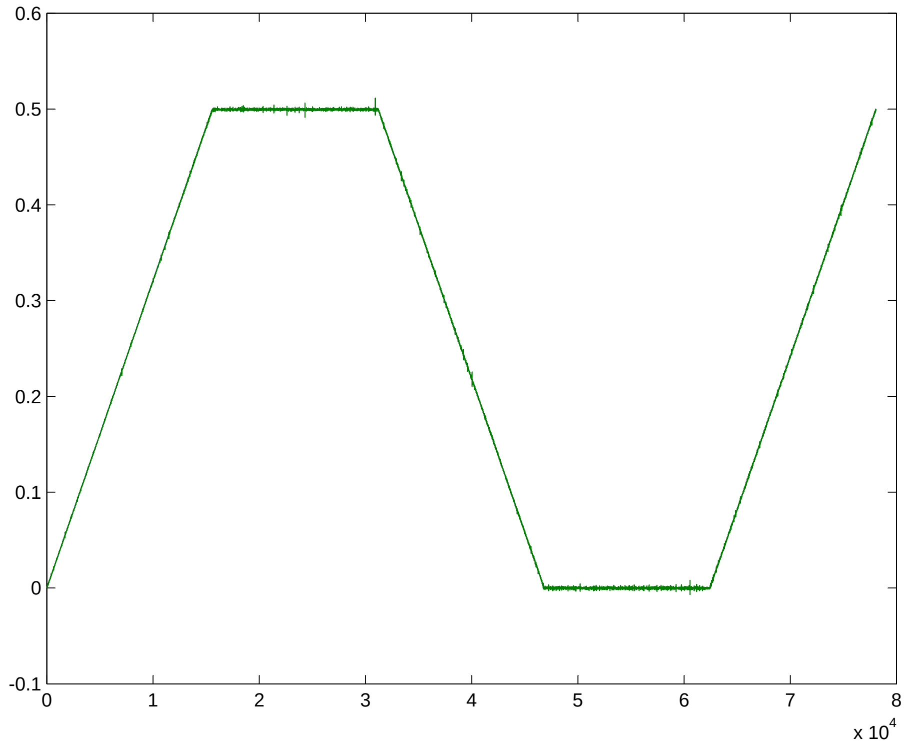


Figure 27: Test of the current loop

solutions of the equation is zero; seeing that the two solutions are equal:

$$p_{1,2} = \frac{-\frac{K_v}{M} \pm \sqrt{\frac{K_v^2}{M^2} - 4\frac{K_a}{M}}}{2} \tag{49}$$

and therefore the condition sought is the following:

$$K_v = 2\sqrt{K_a M} \tag{50}$$

With this condition the poles will be real and coincident:

$$p_{1,2} = -\frac{K_v}{2M} = -\frac{2\sqrt{K_a M}}{2M} = -\sqrt{\frac{K_a}{M}} \tag{51}$$

To satisfy the second condition, knowing that the parameter Ka is fixed, it is necessary that the mass of mobile part is as small as possible. In the fig.28 is shown the root locus of the system dimensioned as stated above. Below it will be done a study and evaluation of the rise time of the system varying only the mass of the mobile part. It will be evaluated the range of mass between 20g and 80g in the fig.29, fig.30, fig.31 and fig.33. The result is shown in the table:

Mass[g]	20	40	60	80
Rise time[s]	0.026	0.037	0.045	0.052

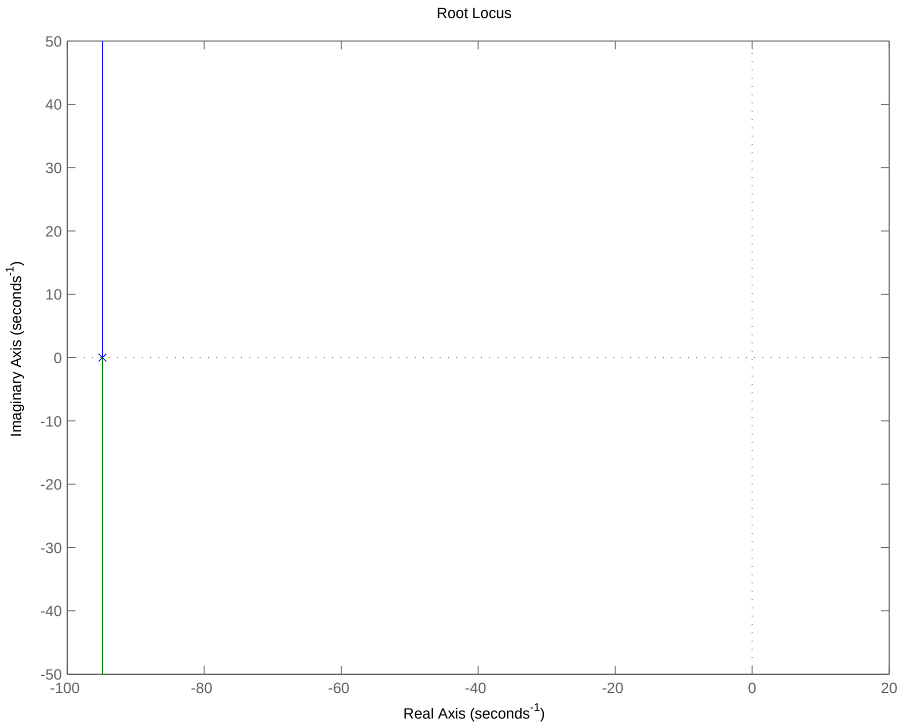


Figure 28: Root locus of system of second-order reduced

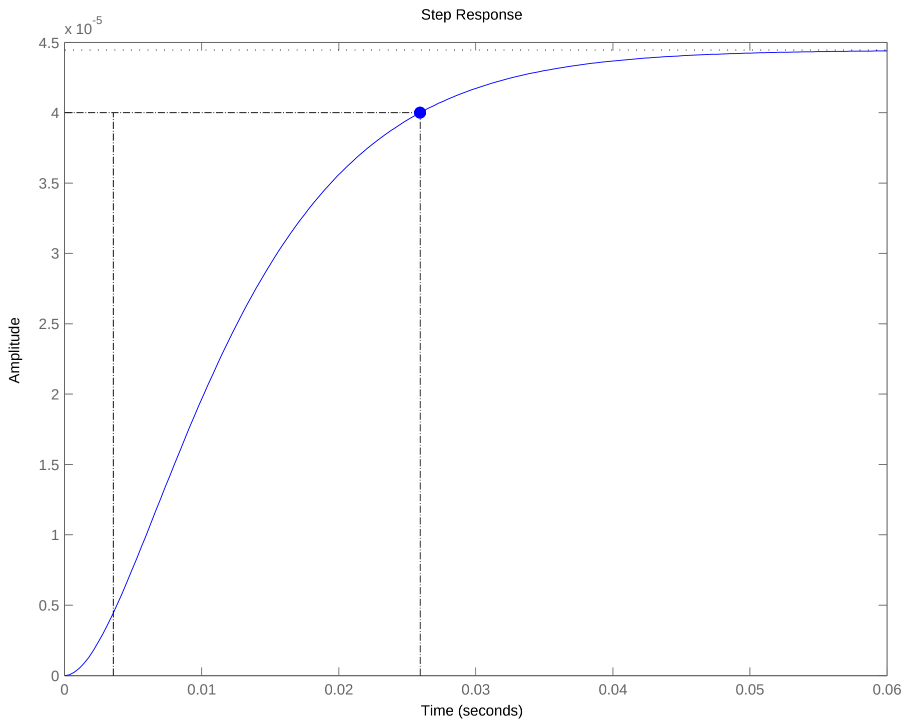


Figure 29: Rise time of system of second-order with a mass of 20g

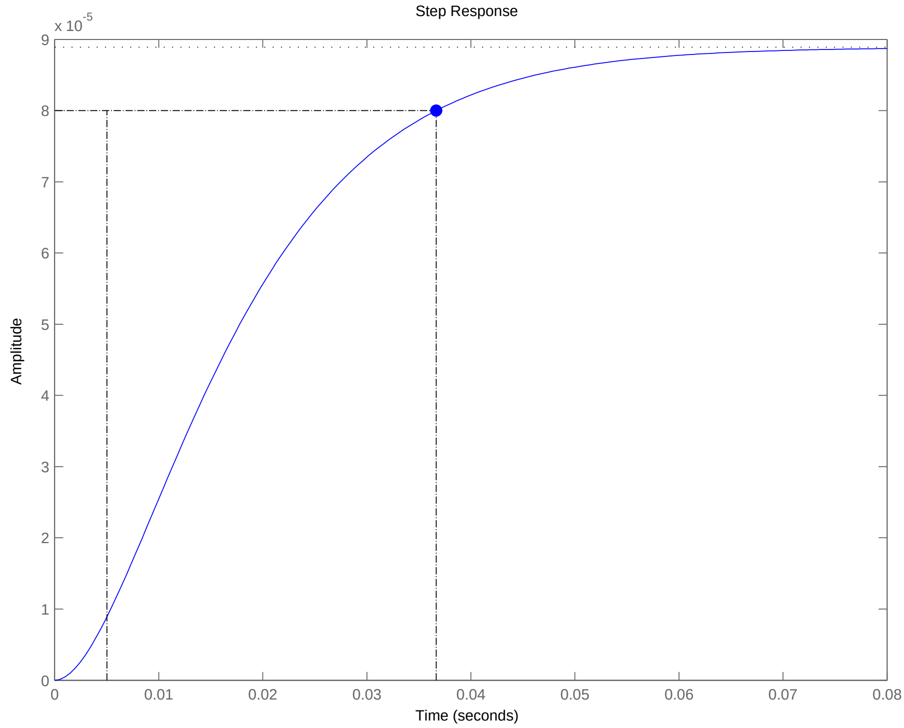


Figure 30: Rise time of system of second-order with a mass of 40g

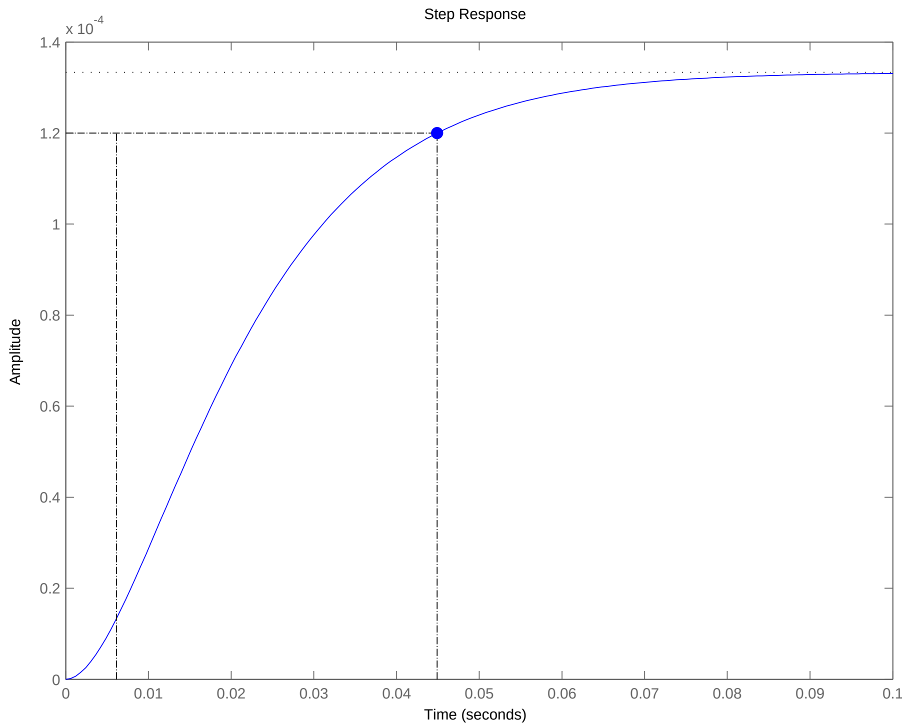


Figure 31: Rise time of system of second-order with a mass of 60g

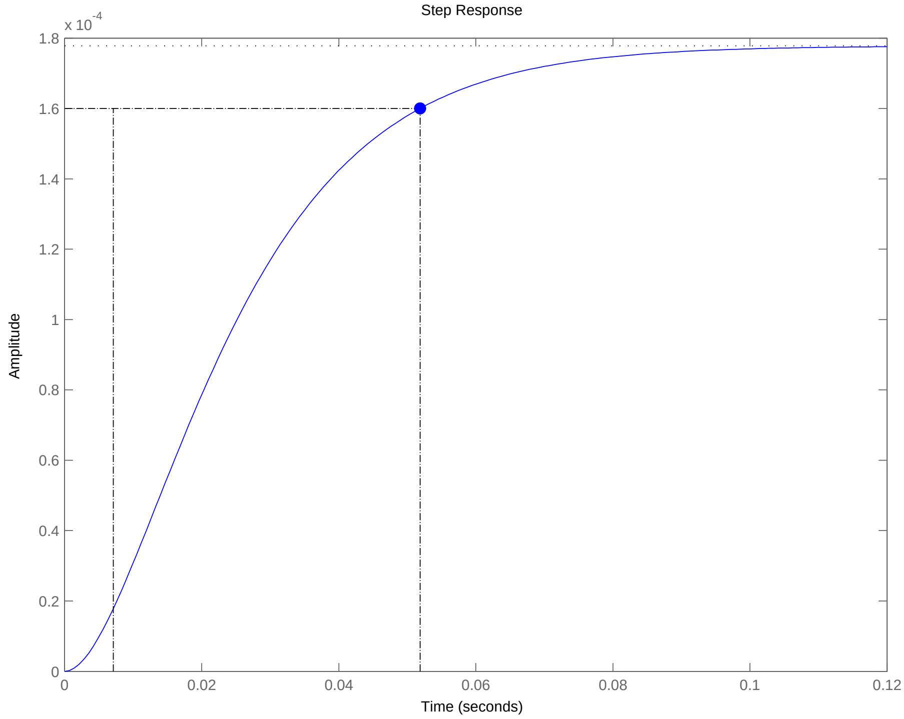


Figure 32: Rise time of system of second-order with a mass of 80g

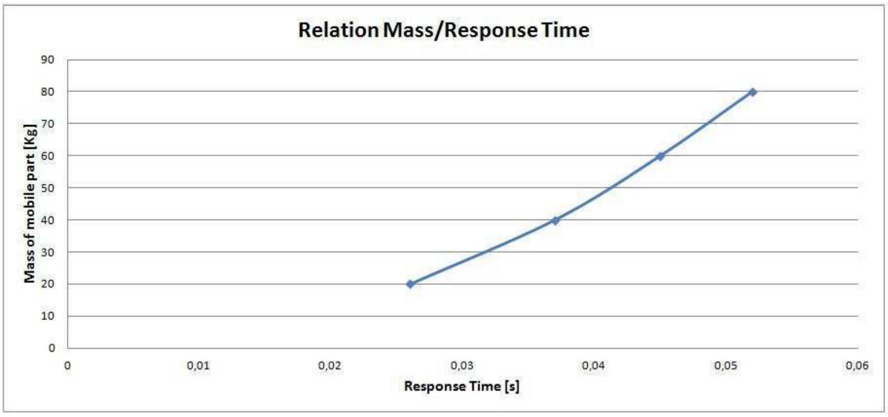


Figure 33: Rise time of system of second-order with a mass of 80g

4.3 SIMULATION AND CONCLUSION

The final result is to see if the slave force produced by motor torque is the same that the slave forced product through the eq.(27). To obtain this result the torque in output from the model of motor is put in the eq.(36). The final result is shown in the fig.34 and in the fig.35 with an average value of the mass of 50 g. Generally, the motor doesn't

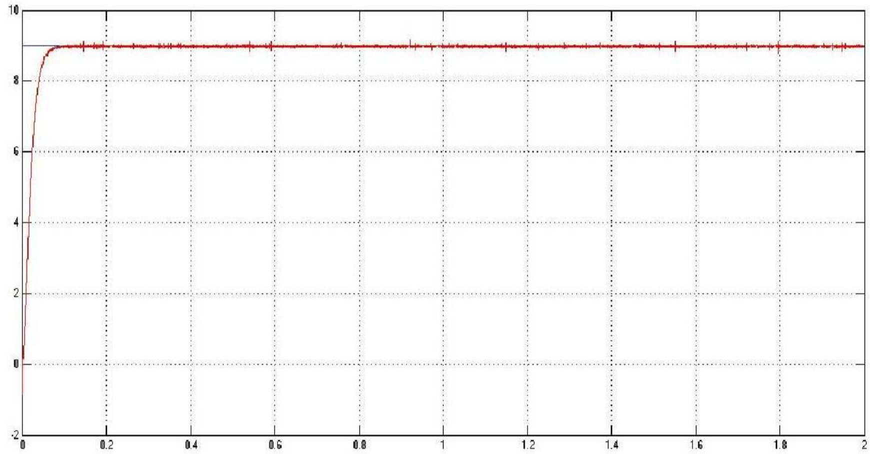


Figure 34: Compare for the final result

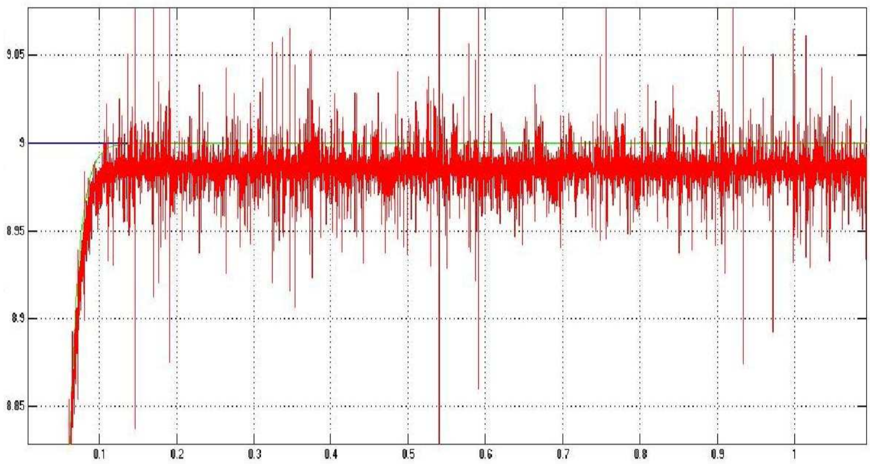


Figure 35: Compare for the final result more detail

provide all the necessary torque to produce the slave force but only a part, that it can be seen like the calculated torque multiplied for a coefficient:

$$C_{produced} = K_c C_{calculated} \tag{52}$$

Usually, it can assume the coefficient $K_c \approx 0.6$. After this assumption the end result is modified as in fig.36. In conclusion it can say that the

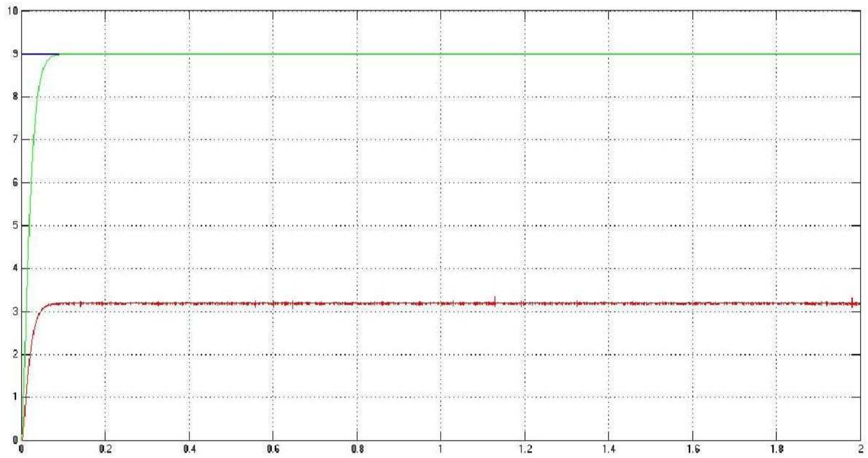


Figure 36: Compare for the final result with the torque coefficient

system responds very well. The response time of the system depend from the following factors:

- A good sizing of the system: in fact it is seen that the system with the mass of mobile part very little responds much faster then the system with a big mass.

- The control loop of the current in the motor: in fact, a good control of the current allows a good and fast enslavement of reference. It can improve the control avoiding a leakage oscillations about the current reference.

In this study several assumptions were made:

- The F_{slave} has been modeled like the force proportional to a translation through a coefficient fixed K_a; a future work is to model this coefficient taking into account also the breathing of the patient(like in the eq.(16).

- To obtain a good response in the direct model of the probe has been introduced a haptic coefficient damping. We must verify the real possibility of achieving a damper inside the probe.

- The motor has been modeled like a DC motor. It is necessary to change the scheme of this motor with a AC motor brushless.

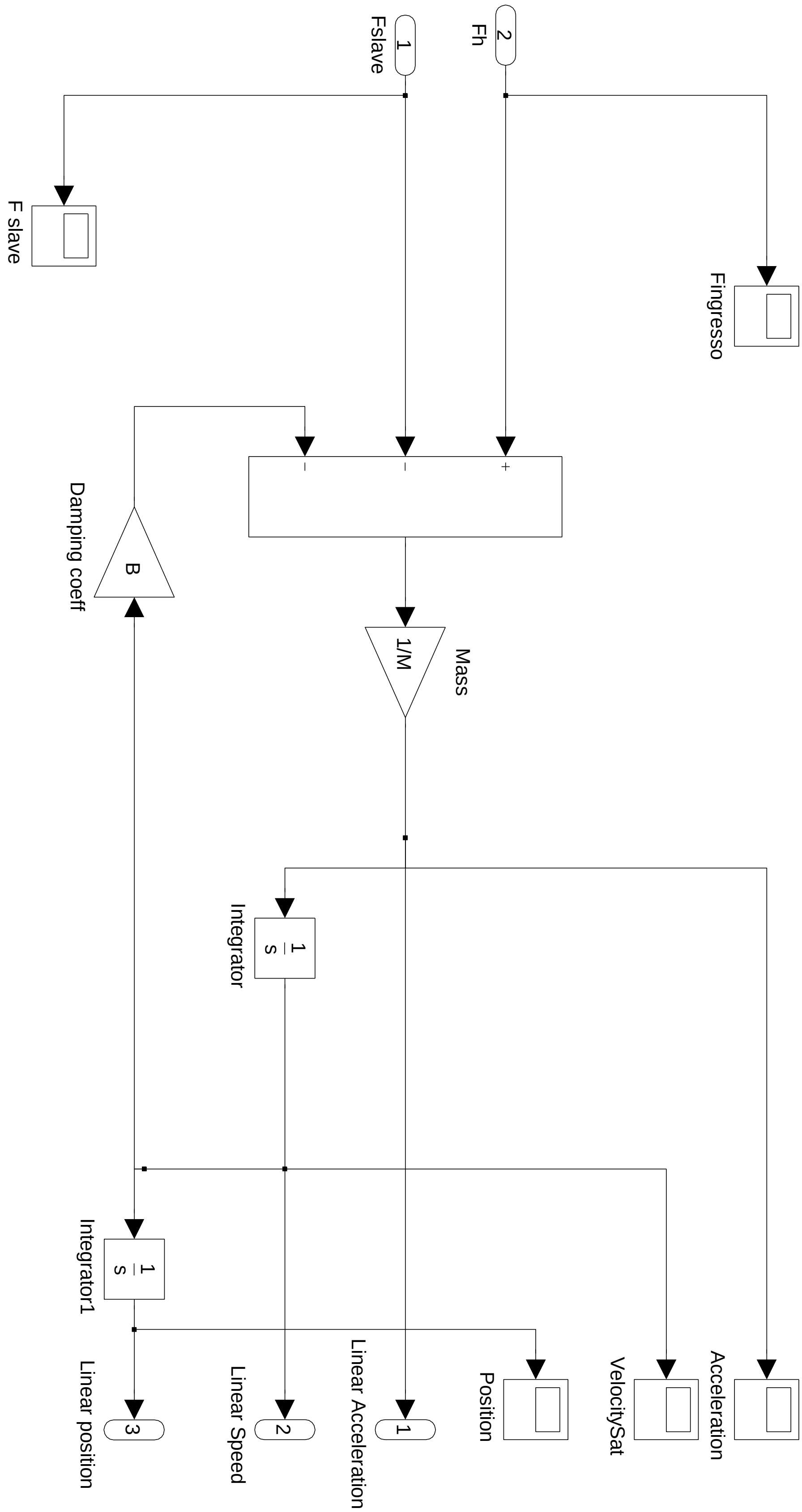


Figure 37: Direct model of the Haptic Probe(more in detail)

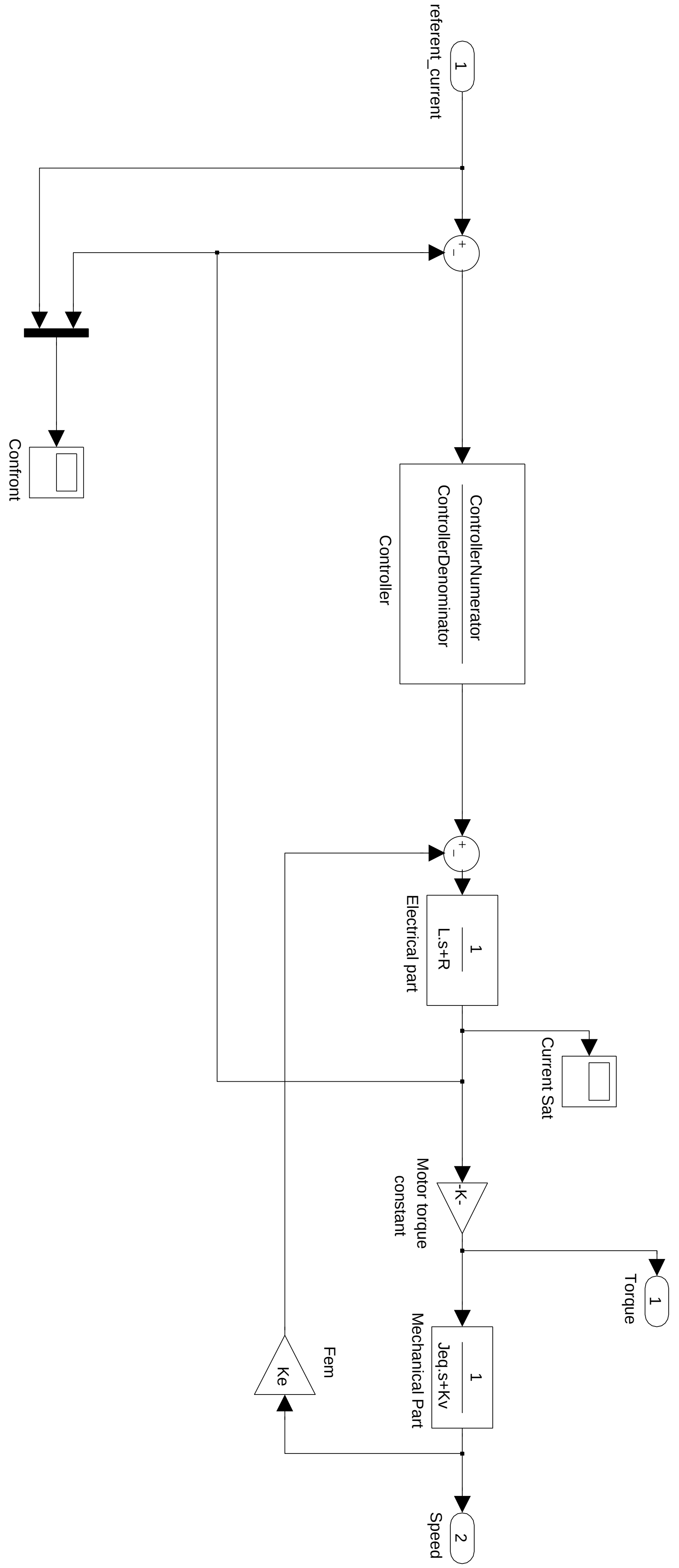


Figure 38: Model of the motor(more in detail)

5 WORTEX 2012 AND BIBE 2012

5.1 INTRODUCTION

The works and the results found have been tested in a event created with the collaboration of more institutes. This events is called "Wortex 2012"(WOrld Wide Robotised Tele-Echography eXperiment - 2012) and it is born in order to give demonstrations of remote tele-echography system in order to bring this technology into states or countries where it is not yet widespread and, at the same time, to advertise the product so that it is used in hospitals in the future by increasing the level technology in the medical department. At this demonstration have participated many institutes and people:

- France, Bourges, Prisme Laboratory: in this place the manipulation has been effectuated in a part of the laboratory with a robot Otelo and the passive probe; the team in this institute was composed from Chiccoli Marco, the candidate of this document; Juan Sebastian Sandoval, a colombian intern and also with the help of Nicola Carretta, a italian intern; Noura Ayadi, a tunisian PhD and Nicola Morettes, a french researcher.

- Cyprus, Limassol, CUT : in this place the manipulation has been performed from Sotos Voskarides, a professor of the University of Limassol and from Sotiris Avgousti, a researcher at the same university.

- USA, Vermont, University of Vermont: in this place the manipulation has been effectuated from Cyril Novales and Laurence Josserand, two professor of IUT of Bourges, with the help of the local professor in Vermont, Tobey Clark.

- Peru, Lima, Lima Maternity : in this place the manipulation has been effectuated from Pierre Vieyres, a professor of IUT of Bourges, with the help of the local professor Rossana Rivas.

- Perù, Arequipa, Arequipa Maternity: in this place the manipulation has been effectuated always from Pierre Vieyres with the help of the local staff.

The demonstration has been executed in the week from 6th june to 13th june; in the next chapter will be shown the total program of the demonstration. Moreover, in this chapter will be showed also the article written from the team of laboratory Prisme that it will be presented in the conference in Cyprus called Bibe 2012. The article is entitled "A predictive control approach and interactive GUI to enhance distal environment rendering during robotized tele-echography(Interactive platform for robotized telechography)" and it is composed by three parts: the predictive control(Juan Sebastian Sandoval, Laboratory Prisme), the graphic user interface(Chiccoli Marco, Laboratory Prisme) and the lines of communication(Sotiris Avgousti, CUT Cyprus).

Figure 39: Wortex 2012 Logo

5.2 WORTEX 2012

5.2.1 *Connection of system*

As first thing it will be shown how it connects the global system to effectuate the test and the demonstration. The connection can be of two types:

1. Ethernet: the lines of communication between the expert station and the network is composed by a ethernet cable and also the communication between the patient station and the network is composed by a ethernet cable.

2. Satellite: the lines of communication between the expert station and the network is composed by a satellite while the communi-

cation between the patient station and the network is composed by a ethernet cable.

To render the explication more easy are shown the two figures that represent the scheme of connection with the cable ethernet(fig.40) and with the satellite(fig.41).

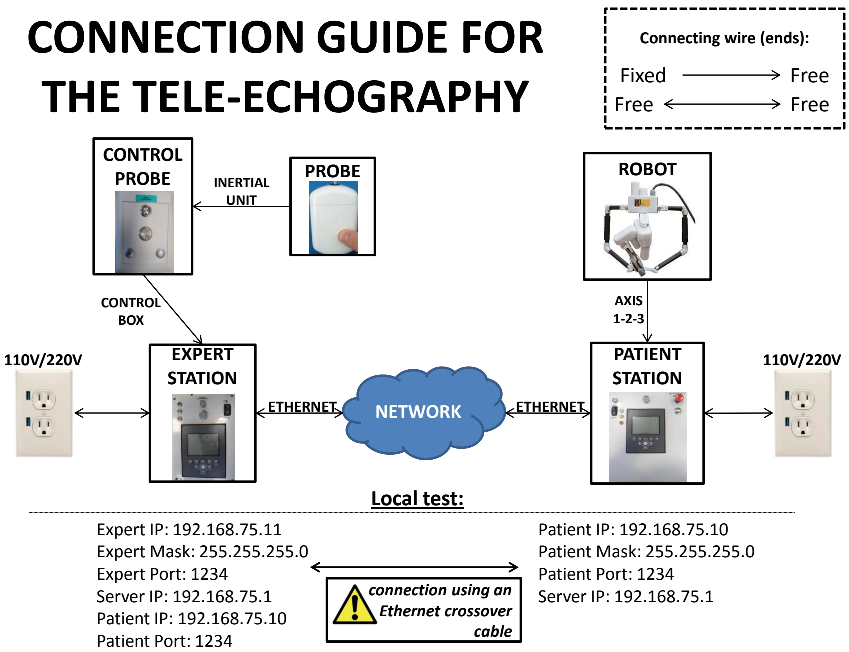


Figure 40: Connection by cable ethernet

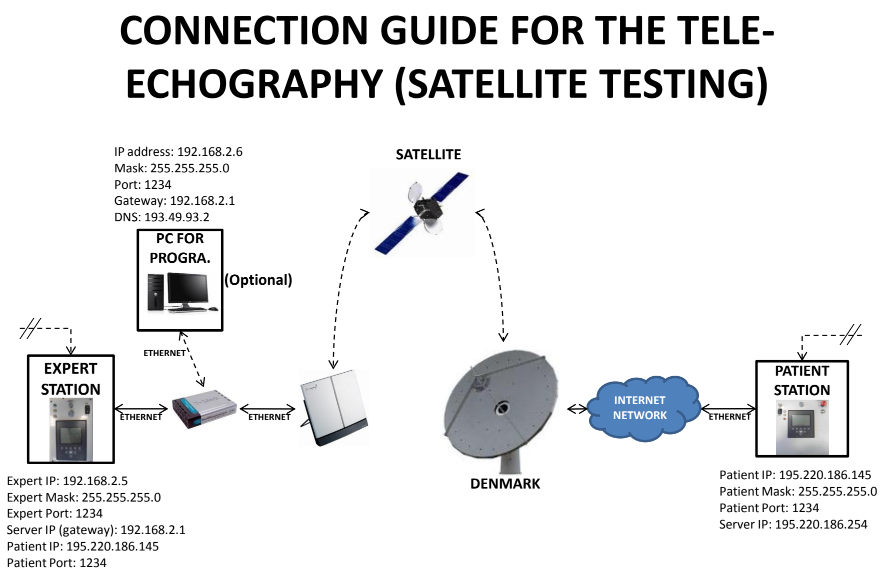


Figure 41: Connection by satellite

5.2.1.1 *Connection scheme by cable ethernet*

In this configuration there are two stations, the expert and the patient, connected through the network ethernet. In the expert site there is a passive probe, connected to the box "Control Probe" which in turn is connected to the box "Expert Station". The box "Control Probe" serve to control the movement of the probe. The "Expert Station" serve to set all the IP address of :

- Expert Station
- Expert Mask
- Expert Port
- Server IP
- Patient IP
- Patient Port

The expert station is feeded from the voltage 220V in Europe and 110V in USA. In the patient site there is the robot that is connected to box "Patient Station"; the latter serves to set the IP address of:

- Patient IP
- Patient Mask
- Patient Port
- Server IP

5.2.1.2 *Connection scheme by satellite*

The connection scheme with the satellite is a bit different from the previously. The "Expert Station" is connected always to the part for the passive probe and serves to set the same IP address of before. In this scheme there is also a switch ethernet: this one is connected to a user PC to set its configuration put the IP address of the satellite, the mask, the port, the gateway and the DNS. The switch is connected to an antenna that transmits the data to the satellite. Then the satellite transmits all information to another antenna in Denmark, that, through the internet connection, send the information to the patient site. In the patient station there is again the box "Patient Station" that serves to set the same IP address of before.

5.2.2 *Zones interesting of the body for the test*

To execute the test and to have the same results in all the place it has been adopted a common criterion: the echography have to be done in specific parts of the patient's body. In the next fig.42 is represented a scheme of the human's body and the name of the parts that the medic has to visit. There are three main zones; the their specific names are:

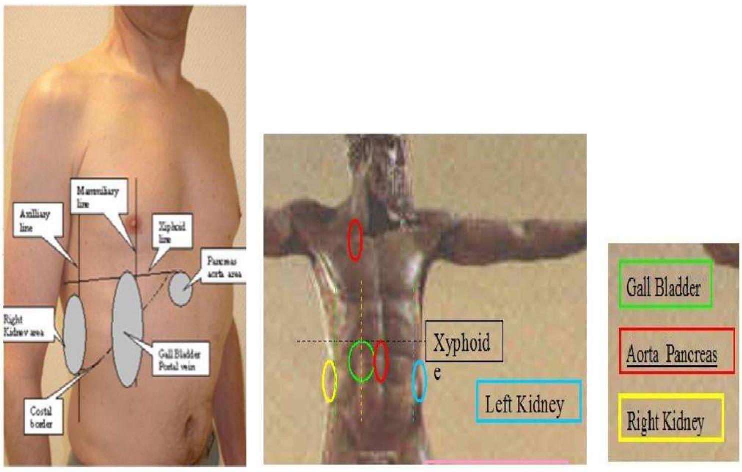


Figure 42: Body'sParts that the medical expert has to visit

1. Gall Bladder
2. Aorta Pancreas
3. Right Kidney

5.2.3 *Calendar of the tests*

To do all tests in mode ordered and in order to respect the time zone of the various countries it has been done a time table to be met in order to work without to have overlapping of manipulations. After many discussions, it is arrived to a common solutions to all the sites.

5.2.4 *Results of tests and pubblicity of the event*

The results of these tests have been very important: in fact this technology has been known in countries of the world where was totally unknown. The test have had success without big problems. The only problem has been the orientation of the passive probe: in fact when the system is on for the first time the robot executes a handling to initialize its three axis in a initial position; in this case the passive probe has to have the same orientation of the end-effector of the robot in order that, when the expert begins the manipulation, the same orientation is maintained throughout the test. During the echography, when the expert finds a interesting point, can stop his passive probe pushed a button in the box "control probe"; when he rehabilitates the functionality of the probe with the same button it loses the orientation between the probe and the end-effector. Therefore, the expert must

(a) *Pierre Vieyres with the doctors in Arequipa.* (b) *The doctor from the Bourges's hospital.*

Figure 43: Demonstration of tele-echography; Expert : Bourges(France), Patient: Arequipa (Peru)

(a) *The patient site in Bourges.* (b) *The patient site in Vermont.*

Figure 44: Two sites patient: Bourges(France) and Vermont(USA)

be able to recover the orientation of the probe looking the end-effector of the robot. All the test have been followed from many agency advertising agencies of various countries. The tests were accompanied by interviews conducted by Zoe Morrison and Ann Robertson from University of Edinburgh on this new technology adopted, each respondent's opinion on this technology and future prospects. In the next fig.43,44 are shown the images that represented the main moments of the tests.

5.3 bibe 2012

5.3.1 *Introduction*

Bibe is a IEEE 12th International Conference on BioInformatics and BioEngineering and it will be set in Cyprus, Larnaca, in the November 11-13[2].

5.3.1.1 *Aim and Scope*

The annual IEEE International Conference on Bioinformatics and Bioengineering cover complementary disciplines that hold great promise for the advancement of research and development in complex medical and biological systems, agriculture, environment, public health, drug design, and so on. Research and development in these two areas are impacting the science and technology of fields such as medicine, food production, forensics, etc. by advancing fundamental concepts in molecular biology and in medicine, by helping us understand living organisms at multiple levels, by developing innovative implants and bio-prosthetics, and by improving tools and techniques for the detection, prevention and treatment of 1diseases. The BIBE series provides a common platform for the cross fertilization of ideas, and to help shape knowledge and scientific achievements by bridging these two very important and complementary disciplines into an interactive and attractive forum. Keeping this objective in mind, BIBE solicits original contributions in the following non exclusive lists of areas.

5.3.2 *The laboratory Prisme's article*

The team of laboratory Prisme has decided to participate at the conference with a article. The article is divided in three main parts all relevant to the robotized tele-echography concepts: after a little introduction, the first part has been written by Juan Sebastian Sandoval and talks about his project on the predictive model control for the architecture in the bilateral system; the second part has been written by Chiccoli Marco and talks about the GUI in the tele-echography system and the project in the laboratory; the third part has been written by Sotiris Avgousti and talks about the line of communication. The final result is that the article has been accepted for the conference. In the appendix A is shown the complete article.

CONCLUSION

Finally, two main works are developed in this document:

1. The design of a graphic user interface with a new functioning for the medical expert.

2. A study on a new haptic device.

The graphic user interface has been developed with the software nokia Qt and is divided in two subparts: one technical, where the engineer expert can set up and test all system; and one medical, where there are the ultrasound image and the interface of the probes. For this interface the candidate has developed two 3D models, a model of the haptic probe and a model of the real probe in the patient site. To execute a tele-echography, the data have to be sent in the line of transmission clashing with, in some case, delay times that can be fixed or variable, note or estimated, and, they can influence the performances (transparency/stability) of the teleoperation system. For this reason, when the medical expert moves his haptic probe, due to these delay times, the echography image that he is seeing may not correspond to the same point on the patient's body that instead he thinks to see. To help the medical expert to overcome this difficulty the candidate has created this interface of the probe, that allows to confront the two positions of the haptic probe and the end-effector to the robot, in order to understand the exact point that he is seeing in that precise instant. As second work the candidate has done a study on a new haptic device: after a introduction on the mechanical and electrical structure, it have been studied the dynamic model and the response time of the probe. The movement of the two 3d models of the haptic probe and of the end-effector of the robot has not been tested because there were some problems downstream of this work: the work has been thought for the robot PROSIT 1 (the test have been effectuated with the platform PROSIT 2 due to some problem on the platform PROSIT 1) and therefore there were many things to change; moreover, the direct geometric model didn't response in correct mode, giving the Angle of Euler incorrectly; and, as last thing, the connection between the expert site and the patient site had some problems of communication, particularly in the protocol TCP/UDP, but after, this problem has been solved. However, one good local test has been executed between the GUI and the robot PROSIT 2: it has been tested the correct connection with the GUI and the robot, and the correct orientation of the 3d model of the haptic probe driving the robot directly by the GUI with a specific tools created ad hoc. The test has been very good because the orientation of the 3d models and the orientation of the robot were the same. Moreover the candidate has done some confidential video of this test. Also the study of the haptic device has led to good results. After to have created a dynamic model of the probe, it has been calculated the necessary torque to reproduce the force on the medic's hand and it has been developed a control current loop of the motor. The model responds very well; in fact, the graphic to compare the force generated by the motor and the force modelized as slave force (feedback force) have been shown highlighting the correct functioning. Then, it has been sized the system in order to obtain a damped response and the fastest possible. A good result have been found showing as the response time changes depending on the mobile mass used. Finally, the works developed have been valued in two events: WORTEX 2012, born in order to give demonstrations of remote tele-echography system in order to bring this technology into states or countries where it is not yet widespread and, at the same time, to advertise the product so that it is used in hospitals in the future by increasing the level technology in the medical department; BIBE 2012, a IEEE 12th International Conference on BioInformatics and BioEngineering in the November 2012, in Cyprus. There are some future works that should be implemented; hereafter the candidate will propose some ones:

1. If it will be possible, to test the correct orientation of the two 3D model during a complete tele-echography demonstration.
2. To improve the model of the probe:
   - To modelize the stiffness coefficient taking into account also the breathing of the patient.
   - We must verify the real possibility of achieving a damper inside the probe.
   - It is necessary to change the scheme of DC motor with a AC motor brushless, with also the control of current and the command of inverter(PWM).
3. After some discussion with the medical team, it's possible to improve the echography exam adding a new axis on the robot: a horizontal translation. Obviously, this movement should be imposed and then sampled in the expert site; the haptic probe has a central inertial that gives the his orientation but not the traslation. To have also the traslation sampled it has been thought to use a touch pad device, where the medical expert will move the haptic probe to execute his exam, in order to have both the measure of the orientation both the measure of the traslation. On this device the candidate has began to work configuring and testing the touch pad with a good results. The next step will be interface the touch pad with the tele-operation chain.

APPENDIX

A

THE ARTICLE IEEE

The article has been accepted for the conference and it will be presented in the november 2012 in Cyprus. The article is shown in the next page.

A predictive control approach and interactive GUI to enhance distal environment rendering during robotized tele-echography

Interactive platform for robotized telechography

Pierre Vieyres, Laurence Josserand, Marco Chiccoli, Juan Sandoval, Cyril Novales, Aicha Fonte
Laboratory Prisme, University of Orleans
63 ave de Lattre de Tassigny
18000 Bourges, France

Soteris Avgousti, Sotos Voskarides, Takis Kasparis
Telemedicine research laboratory
Cyprus University of Technology (CUT)
30 arch Kyprianou str. 3036 Limassol, Cyprus

***Abstract*—Performing a robotized telemedicine act via specific networks brings forth two issues. One is transparency in order to enable the operator, e.g. the medical ultrasound specialist, to safely and accurately perform bilateral tele-operation tasks despite the long time delays inherent to the communication link. To counter these effects, two strategies are combined to improve, at the operator site, the rendering of the interactions between the remote robotic systems with its environment (i.e. the patient), and the control of the robot's orientation. The first approach is the development of a new control architecture based on an internal model providing an anticipated value of the distant environment stiffness; it is complemented with a graphic user interface (GUI) which provides the expert with the real-time relative position of the haptic probe with the robot's end effector for a better tele-operated control. These combined strategies provide the expert with an improved interactive tool for a tele-diagnosis.**

Keywords—bilateral architecture; tele-echography; medical robotic; GUI; force feedback

I. INTRODUCTION

Robotized tele-echography has been developed since the mid-nineties to compensate for the insufficient numbers of ultrasound (US) experts and their regrouping in large medical care facilities centers, resulting in a growing number of medically isolated areas. Ultrasound imaging is a low cost, reliable and non-invasive technique used routinely in hospitals; however, it is an expert-dependent technique, meaning that the experts can only make a diagnosis when they are able to combine anatomical knowledge with the current orientation of the probe and its position on the patient's body and the analysis of the ultrasound images. In [1], Arbeille has shown that robotized tele-echography offers radiologists a solution to making real time diagnoses for remotely located patients with results comparable to diagnoses using standard local echography. This robotized approach is more reliable than the basic tele-ultrasound modality presented in the 90's [2, 3]. Several research teams [4-11] have been developing, with various constraints and objectives (e.g. ultrasound scan, needle insertion, prostate biopsy), tele-operated robotized tele-

a b

Figure 1: (a) positioning of the light-weight (< 3 kg) Prosit[1] robot on the patient (b) haptic probe held by the expert for the orientation of the remote ultrasound probe holder robot

echography systems using portable ultrasound devices via standard communication networks (i.e. internet and satellite). During the tele-echography procedure, a paramedical assistant positions the probe-holder serial robot on a predetermined anatomical location and maintains it on the patient's body (Fig. 1a). The medical expert, using a haptic probe developed by Essomba et al. [12], achieves a position open-loop control of the remotely located robot orientations (Fig. 1b). The role of the haptic probe is two fold: one is the rendering of the contact force applied by the real ultrasound probe on the patient's skin, the other is the recording of the medical expert's gestures to be reproduced by the remote robot's end effector. The patient's ultrasound image is the only information fed back to the medical specialist who analyzes it in real time and provides a diagnosis.

However, performing a robotized tele-echography via specific networks raises two issues. One is transparency in order to enable the operator, i.e. the medical ultrasound specialist, to perform precisely and safely bilateral tele-operation tasks despite time delays inherent in the chosen communication link (e.g. Wi-Fi, satellite); therefore, a bilateral control architecture is necessary to feedback at the expert site the contact force between the remote robot and its environment (i.e. the patient). The second issue is knowledge

This work was supported by PROSIT ANR-08-CORD-017, the S2E2 Smart Electricity Cluster, Cyprus university research Fund, Tototheo Group Cyprus, Cyprus and UK Inmarsat Satellite services

of the accurate positioning of the robot's end-effector with respect to the haptic probe's current position, which effects the analysis of the received ultrasound images on which the expert relies to give a diagnosis; this information has to be embedded in the Expert Graphic User Interface (GUI) to provide better information about the remote robot's behavior.

This paper addresses with these two issues and presents a new control approach to compensate for the long time delays of the communication links, to provide an anticipative force feedback to the expert site, combined with an interactive GUI to provide the expert with an accurate understanding of the orientations of the remote ultrasound probe in contact with the patient's body. The proposed combined approaches are designed to improve the performances of tele-echography robots when using long time delays communication links. The paper is organized as follows: the telemedicine communication protocols network is discussed in part II, part III presents the robotized tele-echography concept, and the needs for a protocol for its safe functioning with respect to patients. Part IV introduces the new control architecture developed to anticipate the contact force between the robot and its environment and the simulation results. Finally, the GUI virtual tool, that provided a complementary tool for the medical expert to give a diagnosis, is presented.

II. TELEMEDICINE COMMUNICATION NETWORK

Telemedicine applications can be categorized as requiring low, medium or high bandwidth transmission. The range of network choices for telemedicine in recent years have included wired communications technologies (plain telephone lines), ISDN and ADSL. Nowadays options include more modern technologies, digital land-lines or cellular/wireless, broadband networks such as broadband Integrated Services Digital Network (BISDN) with the Asynchronous Transfer Mode (ATM), as well as satellite networks, Wireless Local Area Network (WLAN) and Bluetooth, allow the operation of ambulatory and mobile telemedicine systems [13]. It should be noted that when considering telemedicine and telecommunications technologies, it is important to evaluate not only their capabilities and cost/performance trade-off but also the general technical development [14]. Regarding the transmission of medical data there are no theoretical bandwidth requirements. The range and complexity of telecommunication technology requirements vary with the specificity and characteristics of the given telemedicine application; generally a lack of bandwidth is interpreted as a longer transmission time [15]. Table I gives the bandwidth needed for some telemedicine real time transmission.

Standard GSM can only provide data-transfer speeds of *up* to 9.6 kbps, that only allows real-time small data transmission. GPRS and Edge theoretically allows transfer up to 171 Kb/s and 384 kb/s respectively, but all users in a communication cell share the same bandwidth. It can reliably be used for real-time small data transfer or at least a reasonable quality ambiance video [16]. The 3G cell phone technologies, based on UMTS may support up to 1.75Mb/s, and can support 384kb/s transmissions for medical images [17].

TABLE I. MINIMUM BANDWIDTH REQUIREMENT FOR REAL-TIME TRANSMISSION

Data Type	Minimum Bandwidth Required for Real-time transmission
Digital blood pressure	8 Kb/s
Digital thermometer	8 Kb/s
Oxygen saturation meter	8 Kb/s
Electrocardiogram	16 Kb/s
Ultrasound	320 Kb/s
Dermatology (high resolution and color)	384 Kb/s
Scanned x-ray	384 Kb/s
Mammogram	384 Kb/s
CT	384 Kb/s
Video + audio conference	320+64 Kb/s

Wi-Fi Lan (802.11g, 802.11n) can support from 54Mb/s up to 300Mb/s bandwidth,but must be very well controlled to avoid too many collisions and have real limitations in terms of mobility and coverage [18]. Nowadays, wired communications using dedicated digital lines with high baud rates are present in all hospitals and can be used for telemedicine applications between two hospitals. The connection to mobile or isolated area may use commercial links with high baud rate, such as *ASDL*, or high cost geostationary satellite communications.

In all cases, transmission using IP protocols is highly recommended. The protocol remains always the same, whatever the communication links used. Connection oriented protocols, such as Transmission Control Protocol (TCP/IP), can be used for basic connection (i.e. control data to ensure the devices connections) but is not suitable for images or videos transmission [19, 20]. Intrinsically, this kind of protocol reduces the network rate when the transmission reaches the speed limitation of one of the link of the network chain due to the repetition of packet loss. Connectionless protocols such as User data Protocol (UDP) are better adapted for images or videos, and are not sensitive to the latter problem; the down side is that it does not offer any guarantees that information will be delivered to the final destination. For real-time telemedicine, the choice is simple: it is better to loose one image than to overload and block a communication link. In the robotized tele-echography application presented in this paper, we used a connection-oriented protocol to ensure the connection between the distant sites, and connectionless protocol for data/images transmission. When a control loop is required through the network (e.g. control a remote robot), data are sent using UDP. For this specific medical tele-operated robotic application, to avoid a mechanical divergence of the robot when data loss occurs and for the patient safety, data sent are not robot joint velocity or torque set points but position set points. Hence in case of data loss, the robot remains oriented in the last received data position. Specific protocol (e.g. hybrid connected–connectionless) are developed to control robots through the network, but are not normalized [21]. Once the protocol is well defined for a given telemedicine application, the users have to face with the time varying delay inherent to the chosen communication link and its consequences on the stability and transparency properties for the tele-operated robotic system; Solutions to maintain transparency and stability properties during the tele-echography medical act are discussed in the following sections.

III. PRESENTATION AND ISSUES OF THE ROBOTISED TELE-ECHOGRAPHY SYSTEM

A. The tele-echography platform

The system includes three parts linked to each other: the expert station (i.e. the operator), the patient station and the communication link that enables data exchange between the two stations (Fig. 2.).

- *The expert (Master) station:* the specialist uses a haptic fictive ultrasound probe to control the robot end-effector holding the real ultrasound probe. The haptic device allows the expert to feel the interactions between the distal environment (i.e. the patient's body) and the real ultrasound probe.

- *The patient (Slave) station:* the robot carrying the real ultrasound probe is positionned and maintained on the patient by an assistant. The robotic probe holder system reproduces the medical expert's gestures during the tele-echography procedure thanks to the haptic device.

- *The communication link:* the communication network links the expert station to the patient station. Depending on the requirements and availability at the sites involved, a tele-operation system may use different types of communication links (e.g. WLAN, Satellite). However, these network links may generate transmission varying time delays, which hinder the overall robotic tele-operation and force feedback rendering at the expert site. In addition, a videoconferencing system is used for visual and auditory interactions between the expert, the patient and the assistant, and to transfer the ultrasound images from the patient station to the expert station.

B. Constraints of the tele-operated medical procedure

A robotic tele-echography system must provide the specialist with the best working conditions for a remote medical consultation and ensure safety for the patient. However, the use of the communication links between Expert and Patient stations introduces variable time delays that may lead to disturbances in the system stability and deterioration in the reference trajectory tracking generated by the expert [22]. Figure 3 shows different transmission delays round trip time measured using an Inmarsat satellite communications from different European locations. The average time of these transmission delays is about 1 second with an average variation of 200ms. These long time variable delays indicate a need the development of specific control architecture to maintain stability and transparency in the tele-operated system.

This paper presents communication protocols to be used for a tele-operated procedure and the combined solutions developed to provide an interactive and efficient system for the medical expert to perform the tele-operated actions.

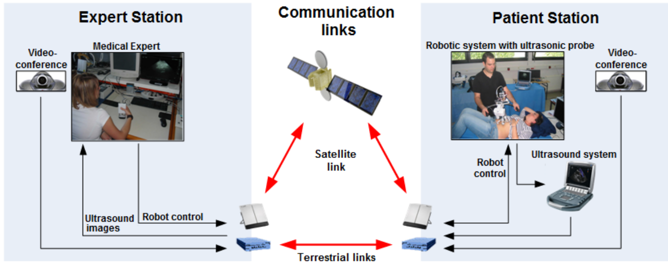


Figure 2: General scheme of a robotic tele-echography system (from [22])

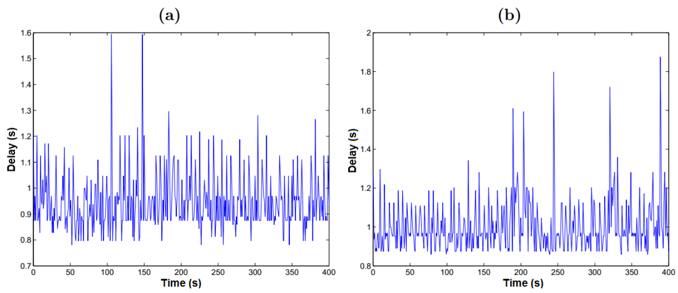


Figure 3: Transmission delays (round trip time) measured when using Inmarsat satellite link from (a) Bourges, France and from (b) Limassol, Cyprus

A comparison of TCP and UDP protocols is done with reference to telecommunication link characteristics such as variable time delay, bandwidth and reliability, which are of great importance to carry out a tele-operated telemedicine task.

The protocol choice and the network will impact the development of the bilateral control architecture needed to ensure: the transparency of the system, i.e. the force exerted by the robot on the patient's body should be equivalent to the force fed back to the expert, and the trajectory accuracy of the medical gesture, whatever the transmission delays. Designing a GUI will allow the expert to compare the orientation of the fictive probe with respect to that of the ultrasound probe. This functionality offers reassurance to the specialist about the ultrasound image analysis taking into account the communication delays.

IV. BILATERAL CONTROL ARCHITECTURE

In order to preserve the stability and transparency of the system, we developed and compared two bilateral control architectures. The system is transparent under two conditions: the robot end effector holding the real probe reproduces the same orientations as the one generated by the expert manipulating the haptic probe (fictive probe) and the force rendered by the haptic probe of the expert is the same as the force exerted by the probe on the patient's body.

A. Wave variables architecture

Keeping in mind the objectives of stability and transparency in the tele-operated system, we used a modified passive bilateral control architecture (position/ force) added with a specific controller on the slave side, based on the (velocity/force) formalism of wave variables introduced by Niemeyer and Slotine [23]. This architecture is referred to as passive as it has no internal energy generation and was used as reference architecture in this work. This ensures the stability

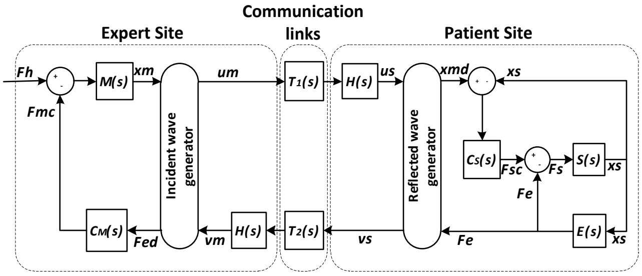


Figure 4: Scheme of the wave variables architecture

of the system regardless of the values of time delays caused by communication links. This architecture is composed of three blocks (Fig. 4). The expert site comprises the operator whose force action is represented by F_h, the model of the haptic probe *M(s)* which enables the remote control of the robot's orientations, the master controller $C_M(s)$ that provides the force feedback F_{mc} to the haptic probe from the measured delayed environment force F_{ed}, finally the incident wave generator constructs the incident wave u_m. The patient site consists of the slave robot *S(s)* interacting with the distant environment *E(s)*, the slave controller $C_S(s)$, and the reflected wave generator. Using the incident waves u_s and the force of the environment, this generator provides the references in position x_{md} and the reflected wave v_s. The communication links transmit the incident and reflected waves between the expert site and the patient site. The delays inherent to communication links between the two sites are represented by T1 and T2, respectively. Figure 5 shows the internal structure of the incident and reflected wave generators. These generators are bijective operators defining u and v waves from a linear combination of the position x and the force f. b is the characteristic impedance of the wave variables and was intitialize to the mean value of the body's stiffness as given in section B. The force F_{ed} fed back to the operator (i.e. the medical expert) depends on the time delays, the received incident wave variable Us and the input position generated by the operator as follows:

$$F_{ed}(t) = bx_m(t) - \sqrt{2b} \cdot u_s(t - T_2(t)) + 2F_e(t - T_2(t)) \qquad (1)$$

The first term of the equation (1) allows an immediate rendering of the force back to the operator as soon as the tele-operated task starts. The second term is a reflection produced upon the transformation of waves and must be filtered before entering the wave generator at the patient site. The last term contains the delayed contact force between the robot and the environment. The simulation results show the position of the robot's end effector with respect to the input position x_m of the haptic probe. x_s represents a sinusoidal displacement of the ultrasound probe in contact with the patient's body (Fig. 6); x_s signal is delayed by a delay time variable T_1 (0.7s). We can note a conservation of the amplitude and frequency of the input position. Figure 7 shows the anticipated force feedback F_{ed} rendered to the expert in comparison to the force F_e generated by the robot's end effector on the patient's body.

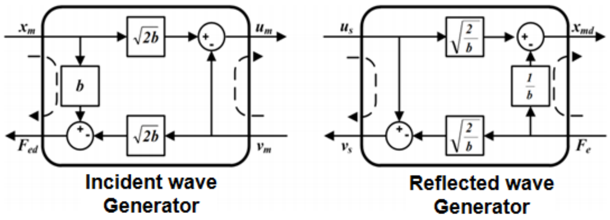


Figure 5: Internal architecture of the two wave generators

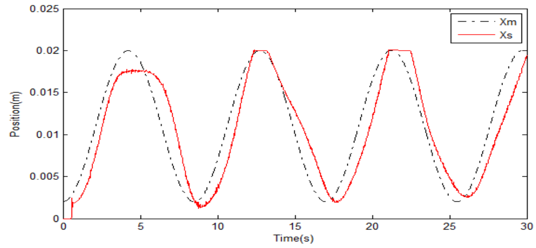


Figure 6: Comparison between the input position x_m and the signal x_s received by the robot's end effector using the wave variables architecture

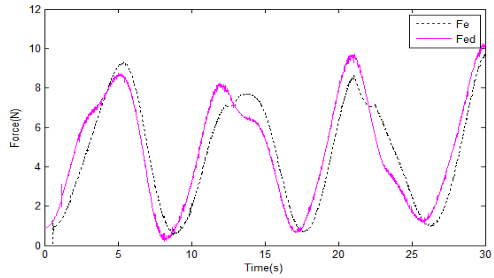


Figure 7: Comparison between the force F_e generated by the robot's end effector in contact with the environment and the force F_{ed} fed back at the expert site using the wave variables architecture

In this case, it was considered a constant patient breathing period of 4.8 seconds. The results validate the anticipative ability of this architecture to ensure the transparency property of the tele-operated system even for long time varying delays.

B. *Stiffness control architecture*

The second bilateral control architecture proposed is based on the stiffness control (Fig. 8). This architecture is based on the theory of internal model control. The expert site uses the patient site model (i.e. robot and environment systems) and the haptic probe position to obtain an estimate of the robot's behavior, anticipated with respect to the reference of the expert time. In addition, the stiffness of the patient's body, with which the robot is in contact, is measured and adjusted with respect to the "expert time" reference. The position and stiffness adjusted to the time reference of the expert allow the reconstruction of the force feedback. The measured compliance Kmes(t) of the patient's body received at expert, given by equation 2, has an average stiffness Kmoy. A sinusoidal signal is added to represent the patient's breathing pattern.

$$K_{mes}(t) = K_{moy} + Amp \cdot \sin(2\pi \frac{t}{T} + \Phi) \qquad (2)$$

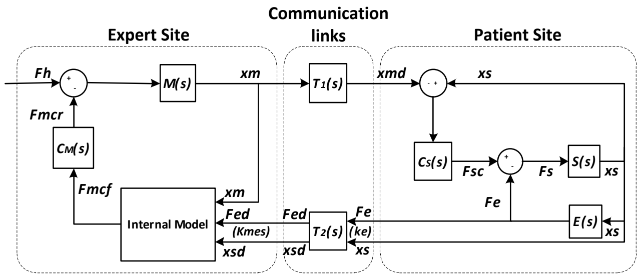


Figure 8: Scheme of the stiffness control architecture

This model is a first approximation of the real phenomenon, including breathing, which is generally more difficult to characterize than by using a simple sine function. However, it is justified by the fact that our approach is robust as it uses an internal model. In addition, for Kmoy, we consider primarily the stiffness of the abdomen in humans, known to be relatively homogeneous (450 N/m + ou – 50 N/m)[24][25]. Hence, four parameters allow a full identification of the measured stiffness K_{mes}; Kmoy, *Amp*, *T* and *Φ*: the average stiffness, the breathing amplitude, the breathing period and its phase, respectively. Kmes(t) is reconstructed and referenced to the expert time and in real time, providing an estimate of the current stiffness K_{est}. The time delays are known as they are evaluated also in real time hence avoiding a possible instability. The four parameters identification was carried out using basic calculation methods in order to validate the concept. However, to refine the results, the Levenberg-Marquardt algorithm was used [26]. This minimization numerical algorithm works as an iterative procedure to minimize a criterion *J* (Equation (3)). In this case, the criterion J represents the square of the difference between the measured stiffness K_{mes} and estimated stiffness K_{est}.

$$J = \sum \{K_{mes}(t) - K_{est}(Amp, K_{moy}, T, \Phi, t)\}^2 \qquad (3)$$

The force feedback F_{mcf} received by the expert is calculated using the estimated stiffness K_{est} and the robot position shifted with respect to the expert time reference (fig. 9). The simulation results show the conservation of the amplitude and frequency of the reference position x_m sent by the expert, and reproduced by the robot on the patient's body, where x_s represents a sinusoidal movement of the ultrasound probe in contact with the patient's body (Fig. 10). As in the previous architecture, these signals are delayed by a variable delay time T_1 (mean 0.7s).

Considering a patient with a breathing period of 4.8 seconds, a breathing amplitude of 100N/m, a breathing phase of 0.52 rd and a K_{moy} of 440N/m, figure 11 shows the anticipated force feedback F_{mcf} compared with the force F_e generated by the robot on the patient's body. This result demonstrates the good transparency provided by the proposed architecture. The results show that overall the two approaches anticipate the contact force between the robot and its environment, and thus satisfy the notion of transparency. However, we note that there are errors in amplitude in both cases of less than 15%, more pronounced in the case of wave variables. Figure 12 illustrates these errors between the input force sent from the robot and the forces rendered at the expert for each of the developed architectures (Fig. 12).

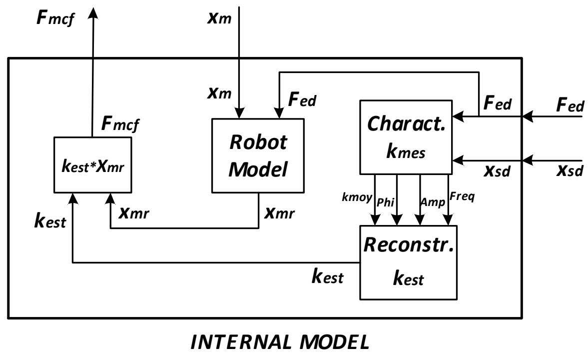


Figure 9: detail of the proposed internal model

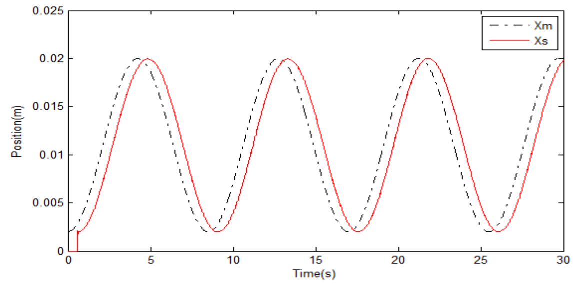


Figure 10: Comparison between the input reference position x_m and the signal x_s received by the robot using the stiffness control architecture

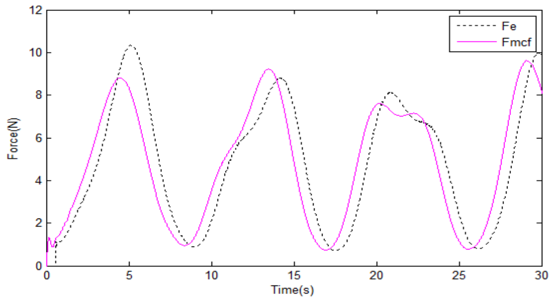


Figure 11: Comparison between the input force F_e generated by the robot and the force feedback F_{mcf} at the expert site using the stiffness control architecture

V. THE GRAPHIC USER INTERFACE AND VIRTUAL PROBES

When using communication links with variable long time delays, the medical specialists will always receives the patient's ultrasound images with a time lag T_2 with respect to the current position of the haptic probe they are manipulating. To provide the medical expert with a full rendering of the force feedback, it is very important that the expert knows exactly the position of the haptic probe in relation to the position of the robot's end-effector corresponding to echography images being received. In order to compensate for the time lag and improve the haptic rendering of the remote environment, the control architecture developed in the previous section is complemented with a dedicated graphic user interface (GUI) including a virtual probe. In this section, we present a joint 3-D model of the haptic probe and of the robot's end-effector, representing their respective orientations in a unique GUI in order to help the expert to perform his remote medical act with the best

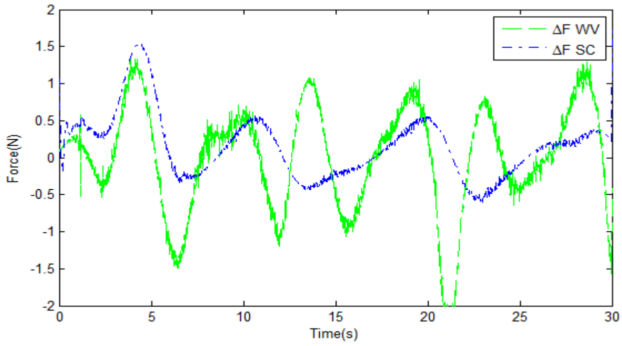


Figure 12: Comparison between the errors of the force feedback Δ*F WV* (wave variables architecture - green dotted line) and Δ*F SC* (stiffness control architecture – blue dotted line)

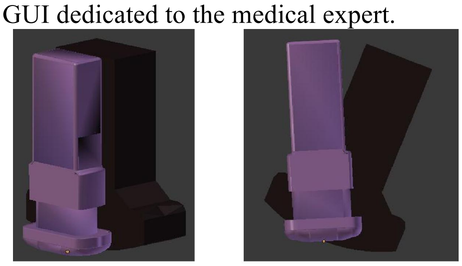
rendering of the distal interactions between the robot and the patient. The 3D model was built with Blender, an open source software used to design objects and 3-D animations; the GUI was developed using Qt, an open source software developed by Nokia company and designed for developing GUI. Within the GUI, the virtual probe and the virtual robot's end-effector, are orientated, in real time, thanks to the orientations data provided by the robotic Prosit; orientations are given with respect to Euler angles frame.

A. Virtual haptic probe

The virtual haptic probe consists of three parts: the mobile element held by the medical expert, the end-tip of the probe representing the ultrasound sensor capturing the ultrasound images and one element joining the two previous ones. The blender is used to design the surface elements of the virtual probe. The final result of the virtual probe is showed in Fig.13a and b.

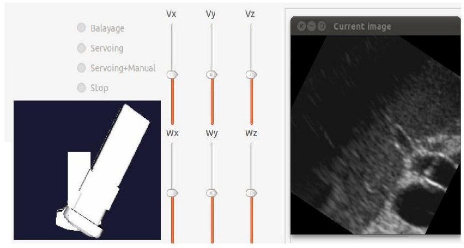
B. Comparison with the robot's end-effector position

A model of the robot's end-effector holding the real ultrasound probe has also been designed using Blender. This second virtual model is added in the GUI to the virtual haptic probe model. The virtual haptic and the end-effector models orientations are extracted and displayed on the GUI, in real time, from data provided by the real haptic probe localization sensors and the robot's end-effector proprioceptive sensors, respectively. Figure 13 presents the GUI display of the two virtual probes 3-D models in their initial position (Fig 13a) and during the tele-echography act when positions are not synchronized due to the time lag generated by the communication link (Fig. 13b). This display is included in the general GUI dedicated to the medical expert.



(a) (b)

Figure 13: comparison between the virtual haptic probe (front purple model) and virtual end-effector (black model) position (a) initial position (b) during the tele-echography act under long time delay



a b

Figure 14: Zoom on the graphic user interface (a) the virtual haptic probe and robot's end-effector orientations difference, (b) the 2-D ultrasound image received at the expert site

C. The Graphic User Interface (GUI)

The GUI developed for the ANR-Prosit is divided in two sub-screens: one for the technical part, which allows an engineer to set and to evaluate the various status of the local and remote systems such as connection protocols, tele-operated modes, proprioceptive sensors, data loss, this part is not presented here; the second GUI display is dedicated to the medical experts and help them to make a diagnosis when performing the tele-consultation. The so-called medical GUI is composed of three parts: the dynamic 2-D ultrasound image, a reconstructed 3D ultrasound image available post consultation, and the virtual probes display discussed in the previous section. Figure 14 represents a zoom of the medical GUI.

VI. CONCLUSION

The GUI improves the tele-operated act of the medical expert and offers in combination with the haptic input device a better rendering of the distal interactions between the robot and its environment (i.e. the patient). Simulation results validate the transparency quality provided by the two architectures for the tele-operated system; they provide the expert a force feedback equal to the force exerted by the robot. That is, the two architectures minimize the negative effects of variable transmission delays on the transparency of the system. It is necessary to validate experimentally the two control architectures will be shortly validated on the tele-echography platform via Inmarsat satellite link.

REFERENCES

[1] P. Arbeille, J. Ayoub, V. Kieffer, P. Ruiz, B. Combes, A. Coatrieux, P. Herve, S. Garnier, B. Leportz, E. Lefebvre, F. Perrotin : Realtime tele-operated abdom- inal and fetal echography in 4 medical centres, from one expert center, using a robotic arm & ISDN or satellite link. In: IEEE Int. Conf. on Automation, Quality and Testing, Robotic (AQTR 2008). vol. 1, pp. 45–46 (May 2008)

[2] G. Kontaxakis, S. Walter, G. Sakas: EU-TeleInViVo: an integrated portable telemedicine workstation featuring acquisition, processing and transmission over low-bandwidth lines of 3D ultrasound volume images. In: IEEE EMBS Int. Conf. on Information Technology Applications in Biomedicine. pp. 158–163 (2000)

[3] W.J. Chimiak, R.O. Rainer, N.T. Wolfman, W. Covitz : Architecture for a high- performance tele-ultrasound system. In: Jost, R.G., III, S.J.D. (eds.) Medical Imag- ing 1996: PACS Design and Evaluation: Engineering and Clinical Issue. vol. 2711, pp. 459–465. SPIE (1996)

[4] G. Fichtinger, J.P. Fiene, C.W. Kennedy, G. Kronreif, I. Iordachita, D.Y. Song, E.C. Burdette, P. Kazanzides: Robotic assistance for ultrasound-guided prostate brachytherapy. Medical Image Analysis 12(5), 535 – 545 (2008)

[5] K. Masuda, E. Kimura, N. Tateishi, K. Ishihara: Three dimensional motion mechanism of ultrasound probe and its application for tele-echography system. In: IEEE/RSJ Int. Conf. on Intel. Robots and Systems. vol. 2, pp. 1112 –1116 vol.2 (2001)

[6] P. Vieyres, G. Poisson, F. Courrèges, C. Novales, N. Smith-Guerin, Ph. Arbeille, C. Brù: A tele-operated robotic system for mobile tele-echography : The OTELO project in Chapter 23 ; *M-health, Emerging Mobile health Systems book,* Springer publisher, ISBN : 0-387-26558-9, pp. 461-474, 2006

[7] L. Bassit, G. Poisson, P. Vieyres: Kinematics of a dedicated 6 DOF robot for tele-echography. In: Int. Conf. on Advanced Robotics (ICAR). pp. 906–910 (2003)

[8] F. Najafi, N. Sepehri: A novel hand-controller for remote ultrasound imaging. Mechatronics 18(10), 578–590 (2008)

[9] Z. Neubach, M. Shoham, M.: Ultrasound-Guided Robot for Flexible Needle Steering. IEEE Trans. Biomed. Eng. 57(4), 799 –805 (Apr 2010)

[10] A. Vilchis, J. Troccaz, P. Cinquin, K. Masuda, F. Pellissier: A new robot architecture for tele-echography. IEEE Trans. Robot. Autom. 19(5), 922–926 (Oct 2003)

[11] K. Ito, S. Sugano, H. Iwata, : Portable and attachable tele-echography robot sys- tem: FASTele. In: IEEE Int. Conf. on Engineering in Medicine and Biology Society (EMBC). pp. 487 –490 (2010)

[12] T. Essomba, M. A. Laribi, J.P. Gazeau, S. Zeghloul, G. Poisson: Contribution to the Design of a Robotized Tele-Ultrasound System, in: 2nd IFToMM International Symposium on Robotics and Mechatronics, (ISRM). Shangai, Chine (2011).

[13] Pattichis, C., Kyriacou, E., Voskarides, S., Pattichis, M., Istepanian, R. & Schizas, C. 2002, "Wireless telemedicine systems: an overview", *Antennas and Propagation Magazine, IEEE,* vol. 44, no. 2, pp. 143-153.

[14] Irini RELJIN1, B.R. 2001, "Telecommunication Requirements in Telemedicine", .

[15] Sachpazidis, I., Ohl, R., Kontaxakis, G. & Sakas, G. 2006, "TeleHealth networks: Instant messaging and point-to-point communication over the internet", *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment,* vol. 569, no. 2, pp. 631-634.

[16] Istepanian, R. & Chandran, S. 1999, "Enhanced telemedicine applications with next generation telecommunication systems", BMES/EMBS Conference, 1999. Proceedings of the First JointIEEE, , pp. 708 vol. 2.

[17] Mueckenheim, J., Schacht, M. & Kettschau, J. 2004, "On system level modelling of UMTS downlink shared channel scheduling", 3G Mobile Communication Technologies, 2004. 3G 2004. Fifth IEE International Conference onIET, , pp. 392.

[18] Zhang, Y., Ansari, N. & Tsunoda, H. 2010, "Wireless telemedicine services over integrated IEEE 802.11/WLAN and IEEE 802.16/WiMAX networks", *Wireless Communications, IEEE,* vol. 17, no. 1, pp. 30-36.

[19] Slama, T., Trevisani, A., Aubry, D., Oboe, R. & Kratz, F. 2008, "Experimental analysis of an internet-based bilateral teleoperation system with motion and force scaling using a model predictive controller", *Industrial Electronics, IEEE Transactions on,* vol. 55, no. 9, pp. 3290-3299.

[20] Xiaohui, X., Zhijiang, D. & Lining, S. 2003, "The design and implementation of real-time Internet-based telerobotics", *Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003 IEEE International Conference on*IEEE, , pp. 815.

[21] Wirz, R., Marin, R., Ferre, M., Barrio, J., Claver, J.M. & Ortego, J. 2009, "Bidirectional transport protocol for teleoperated robots", *Industrial Electronics, IEEE Transactions on,* vol. 56, no. 9, pp. 3772-3781.

[22] G. Charron: *Contribution à la commande bilatérale et à la gestion des configuration singulières pour le suivi de trajectoire d'un système télé-opéré: application à la télé-échographie robotisée par satellite*. PhD thesis, university of Orléans France, 29 nov 2011.

[23] Niemeyer, G. and Slotine, J.-J. 1997, «Designing force reflecting teleoperators with large time delays to appear as virtual tools», in Proceedings of the IEEE International Conference on Robotics and Automation, vol. 3, p. 2212–2218.

[24] G. J Tortora, N.P.Anagnostakos, Principles of Anatomy and Physiology, 6e édition, New York: Harper-Collins, (1990).

[25] L. Sherwood, Fundamentals of Physiology : A Human Perspective, Thomson Brooks/Cole,(2006)

[26] D. Marquardt : An Algorithm for Least-Squares Estimation of Nonlinear Parameters, SIAM Journal on Applied Mathematics, vol. 11,pp. 431–441 (1963).

B

THE CODE OF GRAPHIC USER INTERFACE

B.1 THE CODE FOR THE HANDLING OF THE PROBE HAPTIC

```
void usgqtgui::newProbePos()
{

  /*** CHOOSE THE COMAND MODE ***/
  /******************************/


  if(modeControl) // Chiccoli Marco
  {
    haptiqueProbePos[0] += (ui->verticalSliderWx->value())*(
        double)period/(1000*5);
    haptiqueProbePos[1] += (ui->verticalSliderWy->value())*(
        double)period/(1000*5);
    haptiqueProbePos[2] += (ui->verticalSliderWz->value())*(
        double)period/(1000*5);
  }
  else //Chiccoli Marco
  {
            // set the matrix size and init the value to 0
            AngleEulerZXZ.resize(3);
            AngleEulerZYZ.resize(3);

            mutexDonnees.lock(); //protection the write
            //try to init the value read how to Z-X-Z euler
                angles
            AngleEulerZXZ[0]=CurrentHaptEulerPos[0];  //alpha
            AngleEulerZXZ[1]=CurrentHaptEulerPos[1];  //beta
            AngleEulerZXZ[2]=CurrentHaptEulerPos[2];  //gamma
            mutexDonnees.unlock();

            //define the rotation matrix
            vpMatrix rotatZ1;
            vpMatrix rotatX;
            vpMatrix rotatZ2;
            vpMatrix TransfMatrix;

            rotatZ1.eye(4,4);
            rotatX.eye(4,4);
            rotatZ2.eye(4,4);
            TransfMatrix.eye(4,4);

            //calculate the three rotation matrix
            //rotation Z
```

```
            rotatZ1[0][0]=cos(AngleEulerZXZ[0]);
            rotatZ1[1][0]=-sin(AngleEulerZXZ[0]);
            rotatZ1[0][1]=sin(AngleEulerZXZ[0]);
            rotatZ1[1][1]=cos(AngleEulerZXZ[0]);

            //rotation X
            rotatX[1][1]=cos(AngleEulerZXZ[1]);
            rotatX[2][1]=-sin(AngleEulerZXZ[1]);
            rotatX[1][2]=sin(AngleEulerZXZ[1]);
            rotatX[2][2]=cos(AngleEulerZXZ[1]);

            //rotation Z
            rotatZ2[0][0]=cos(AngleEulerZXZ[2]);
            rotatZ2[1][0]=-sin(AngleEulerZXZ[2]);
            rotatZ2[0][1]=sin(AngleEulerZXZ[2]);
            rotatZ2[1][1]=cos(AngleEulerZXZ[2]);

            //calculate the transformated matrix
            TransfMatrix=rotatZ1*rotatX*rotatZ2;

            /*** ALGORITMH TO HAVE THE THREE ANGLE Z-Y-Z ***/
            /*** It use note geometric rules ***************/

            if(TransfMatrix[2][0] || TransfMatrix[2][1])
              {
                AngleEulerZYZ[0]=atan2(TransfMatrix[1][2],
                    TransfMatrix[0][2]);  //phi
                AngleEulerZYZ[1]=acos(TransfMatrix[2][2]);
                                         //theta
                AngleEulerZYZ[2]=atan2(TransfMatrix[2][1],-
                    TransfMatrix[2][0]); //gamma
              }
            else //there are infinity solutions; it choose one
                solution
              {
                if(TransfMatrix[0][0]>0 && TransfMatrix[1][0]>0)
                    quadrant=1;
                else if(TransfMatrix[0][0]<0 && TransfMatrix
                    [1][0]>0)
                   quadrant=2;
                else if(TransfMatrix[0][0]<0 && TransfMatrix
                    [1][0]<0)
                   quadrant=3;
                else  quadrant=4;

                angle=acos(TransfMatrix[0][0]); //solve the cos(x
                    )
                //std::cout << "Angle1: "<< angle << std::endl;

                if(quadrant==3 || quadrant==4) //add o sub the
                    value according to quadrant
                angle=2*PI_GRECO-angle;
```

```
    AngleEulerZYZ[0]=angle/2;
    AngleEulerZYZ[1]=0;
    AngleEulerZYZ[2]=angle/2;
    //std::cout << " The AngleEulerZYZ in the line is
        (other solution): "<< AngleEulerZYZ.t() <<
        std::endl;
  }
/******************************/
/******************************/

//calculate the haptic probe final position
vpRzyzVector haptiqueProbe;
haptiqueProbe[0] = AngleEulerZYZ[0];
haptiqueProbe[1] = AngleEulerZYZ[1];
haptiqueProbe[2] = AngleEulerZYZ[2];

//return directly position of end-effector because
    the parameter is a variable pointed
vpThetaUVector TU(haptiqueProbe);

/* Change the referement with the rotation matrix */
/***********************************************/

//define the angle rotation
double angleX1=-90.0;
double angleY1=90.0;

//ausiliar vector
vpColVector XYZ2;
vpColVector XYZ3;
XYZ2.resize(4);
XYZ3.resize(4);

//init values
XYZ2[0]= TU[0];
XYZ2[1]= TU[1];
XYZ2[2]= TU[2];

//define the rotation matrix
//vpMatrix rotatZ;
vpMatrix rotatYY;
vpMatrix rotatXX;
vpMatrix Transformation2;
vpMatrix Transformation3;

//rotatZ.eye(4,4);
rotatYY.eye(4,4);
rotatXX.eye(4,4);
Transformation2.eye(4,4);
Transformation3.eye(4,4);
```

```
            //calculate the three rotation matrix
            //rotation Z
            //rotatZ[0][0] = cos(vpMath::rad(AngleEulerZXZ[0]));
            //rotatZ[1][0] = -sin(vpMath::rad(AngleEulerZXZ[0]));
            //rotatZ[0][1] = sin(vpMath::rad(AngleEulerZXZ[0]));
            //rotatZ[1][1] = cos(vpMath::rad(AngleEulerZXZ[0]));

            //rotation Y
            rotatYY[0][0] = cos(vpMath::rad(angleY1));
            rotatYY[2][0] = sin(vpMath::rad(angleY1));
            rotatYY[0][2] = -sin(vpMath::rad(angleY1));
            rotatYY[2][2] = cos(vpMath::rad(angleY1));


            //rotation X
            rotatXX[1][1] = cos(vpMath::rad(angleX1));
            rotatXX[2][1] = -sin(vpMath::rad(angleX1));
            rotatXX[1][2] = sin(vpMath::rad(angleX1));
            rotatXX[2][2] = cos(vpMath::rad(angleX1));

            //calculate the transform matrix to change the
                coordinates in the new referement
            Transformation2 = rotatXX * rotatYY;

            //calculate the transform matrix to change t in the
                new referement (remember T^-1=T^t)
            Transformation3=(Transformation2.t());

            //calculate the coordinate in the new system
            XYZ3 = Transformation3 * XYZ2;

            //print the value
            //std::cout << " The NewCoordinate in the line is:
                "<< XYZ3.t() << std::endl;

            /*****************************************/
            /*****************************************/


            //calculate the haptic probe final position
            haptiqueProbePos[0] = vpMath::deg(XYZ3[0]);
            haptiqueProbePos[1] = vpMath::deg(XYZ3[1]);
            haptiqueProbePos[2] = vpMath::deg(XYZ3[2]);
            std::cout << "haptique pos : " << haptiqueProbePos.t
                () <<std::endl;

        }

    vpMatrix rotationX;
    vpMatrix rotationY;
    vpMatrix rotationZ;
    vpMatrix rotRez;
```

```
    rotationX.eye(4,4);
    rotationY.eye(4,4);
    rotationZ.eye(4,4);
    rotRez.eye(4,4);

    rotationZ[0][0] = cos(vpMath::rad(haptiqueProbePos[2]));
    rotationZ[1][0] = -sin(vpMath::rad(haptiqueProbePos[2]));
    rotationZ[0][1] = sin(vpMath::rad(haptiqueProbePos[2]));
    rotationZ[1][1] = cos(vpMath::rad(haptiqueProbePos[2]));

    rotationY[0][0] = cos(vpMath::rad(haptiqueProbePos[1]));
    rotationY[2][0] = sin(vpMath::rad(haptiqueProbePos[1]));
    rotationY[0][2] = -sin(vpMath::rad(haptiqueProbePos[1]));
    rotationY[2][2] = cos(vpMath::rad(haptiqueProbePos[1]));

    rotationX[1][1] = cos(vpMath::rad(haptiqueProbePos[0]));
    rotationX[2][1] = -sin(vpMath::rad(haptiqueProbePos[0]));
    rotationX[1][2] = sin(vpMath::rad(haptiqueProbePos[0]));
    rotationX[2][2] = cos(vpMath::rad(haptiqueProbePos[0]));

    rotRez = rotationX * rotationY * rotationZ;

    for(int i=0; i<4; i++)
    {
        for(int j=0; j<4; j++)
        {
          //vtk_oMprobe_obj2->Element[i][j]=rotationX->GetElement
              (i,j);
            vtk_oMprobe_obj2->Element[i][j]=rotRez[i][j];
        }
    }

    probe_obj_transform2->SetMatrix(vtk_oMprobe_obj2); //
        initialize position transformation for probe CAD  model
    probeActor2->SetUserTransform(probe_obj_transform2);

    renWin->Render();

    }
}
```

B.2 THE CODE FOR THE HANDLING OF THE END-EFFECTOR

```
void usgqtgui::affichageEffecteurPosition() //program similar to
    the code for the probe
{
    if(ui->actionRobot->isChecked()){
              // set the matrix size and init the value to 0
              AngleEulerZXZ1.resize(3);
              AngleEulerZYZ1.resize(3);
```

```
mutexDonnees.lock();//protection the write
//try to init the value read how to Z-X-Z euler
    angles
AngleEulerZXZ1[0]=CurrentEulerPos[0]; //alpha
AngleEulerZXZ1[1]=CurrentEulerPos[1];  //beta
AngleEulerZXZ1[2]=CurrentEulerPos[2]; //gamma
mutexDonnees.unlock();

std::cout << "AngleEulerZXZ1 : " << AngleEulerZXZ1.
    t() << std::endl;

//define the rotation matrix
vpMatrix rotatZZ1;
vpMatrix rotatXX;
vpMatrix rotatZZ2;
vpMatrix TransfMatrix;

rotatZZ1.eye(4,4);
rotatXX.eye(4,4);
rotatZZ2.eye(4,4);
TransfMatrix.eye(4,4);

//calculate the three rotation matrix
//rotation Z
rotatZZ1[0][0] = cos(vpMath::rad(AngleEulerZXZ1[0])
    );
rotatZZ1[1][0] = -sin(vpMath::rad(AngleEulerZXZ1
    [0]));
rotatZZ1[0][1] = sin(vpMath::rad(AngleEulerZXZ1[0])
    );
rotatZZ1[1][1] = cos(vpMath::rad(AngleEulerZXZ1[0])
    );

//rotation X
rotatXX[1][1] = cos(vpMath::rad(AngleEulerZXZ1[1]))
    ;
rotatXX[2][1] = -sin(vpMath::rad(AngleEulerZXZ1[1])
    );
rotatXX[1][2] = sin(vpMath::rad(AngleEulerZXZ1[1]))
    ;
rotatXX[2][2] = cos(vpMath::rad(AngleEulerZXZ1[1]))
    ;

//rotation Z
rotatZZ2[0][0] = cos(vpMath::rad(AngleEulerZXZ1[2])
    );
rotatZZ2[1][0] = -sin(vpMath::rad(AngleEulerZXZ1
    [2]));
rotatZZ2[0][1] = sin(vpMath::rad(AngleEulerZXZ1[2])
    );
```

```
rotatZZ2[1][1] = cos(vpMath::rad(AngleEulerZXZ1[2])
    );

//calculate the transformated matrix
TransfMatrix=rotatZZ1*rotatXX*rotatZZ2;

/*** ALGORITMH TO HAVE THE THREE ANGLE Z-Y-Z ***/
/*** It use note geometric rules **************/

 if(TransfMatrix[2][0] || TransfMatrix[2][1])
  {
    AngleEulerZYZ1[0]=atan2(TransfMatrix[1][2],
        TransfMatrix[0][2]);  //phi
    AngleEulerZYZ1[1]=acos(TransfMatrix[2][2]);
                                 //theta
    AngleEulerZYZ1[2]=atan2(TransfMatrix[2][1],-
        TransfMatrix[2][0]); //gamma
  }
else //there are infinity solutions; it choose one
    solution
  {
    if(TransfMatrix[0][0]>0 && TransfMatrix
        [1][0]>0)
        quadrant1=1;
    else if(TransfMatrix[0][0]<0 && TransfMatrix
        [1][0]>0)
       quadrant1=2;
    else if(TransfMatrix[0][0]<0 && TransfMatrix
        [1][0]<0)
       quadrant1=3;
    else  quadrant1=4;

    angle1=acos(TransfMatrix[0][0]); //solve the
        cos(x)
    std::cout << "Angle1: "<< angle1 << std::endl;

    if(quadrant1==3 || quadrant1==4)//add o sub the
         value according to quadrant
       angle1=2*PI_GRECO-angle1;

 //   std::cout << " Quadrant1: "<< quadrant1 <<
     std::endl;

    AngleEulerZYZ1[0]=angle1/2;
    AngleEulerZYZ1[1]=0;
    AngleEulerZYZ1[2]=angle1/2;

  }

/*********************************/
/*********************************/
```

```
  //calculate the haptic probe final position
  vpRzyzVector EndEffector;
  EndEffector[0] = AngleEulerZYZ1[0];
  EndEffector[1] = AngleEulerZYZ1[1];
  EndEffector[2] = AngleEulerZYZ1[2];
  //std::cout << " The EndEffector in the line is:
      "<< EndEffector.t() << std::endl;

  //return directly position of end-effector because
      the parameter is a variable pointed
  vpThetaUVector TU1(EndEffector);
  //std::cout << " The element in the line is: "<<
      TU1.t() << std::endl;

/* Change the referement with the rotation matrix */
/**************************************************/

//define the angle rotation
double angleX=-90;
double angleY=90;

//ausiliar vector
vpColVector XYZ;
vpColVector XYZ1;
XYZ.resize(4);
XYZ1.resize(4);

//init values
XYZ[0]= TU1[0];
XYZ[1]= TU1[1];
XYZ[2]= TU1[2];

//define the rotation matrix
//vpMatrix rotatZ;
vpMatrix rotatY;
vpMatrix rotatX;
vpMatrix Transformation;
vpMatrix Transformation1;

//rotatZ.eye(4,4);
rotatY.eye(4,4);
rotatX.eye(4,4);
Transformation.eye(4,4);
Transformation1.eye(4,4);

//calculate the three rotation matrix
//rotation Z
//rotatZ[0][0] = cos(vpMath::rad(AngleEulerZXZ[0]));
//rotatZ[1][0] = -sin(vpMath::rad(AngleEulerZXZ[0]));
//rotatZ[0][1] = sin(vpMath::rad(AngleEulerZXZ[0]));
//rotatZ[1][1] = cos(vpMath::rad(AngleEulerZXZ[0]));
```

```
        //rotation Y
        rotatY[0][0] = cos(vpMath::rad(angleY));
        rotatY[2][0] = sin(vpMath::rad(angleY));
        rotatY[0][2] = -sin(vpMath::rad(angleY));
        rotatY[2][2] = cos(vpMath::rad(angleY));


        //rotation X
        rotatX[1][1] = cos(vpMath::rad(angleX));
        rotatX[2][1] = -sin(vpMath::rad(angleX));
        rotatX[1][2] = sin(vpMath::rad(angleX));
        rotatX[2][2] = cos(vpMath::rad(angleX));

        //calculate the transform matrix to change the
            coordinates in the new referement
        Transformation = rotatX * rotatY;

        //calculate the transform matrix to change t in the
            new referement (remember T^-1=T^t)
        Transformation1=(Transformation.t());

        //calculate the coordinate in the new system
        XYZ1 = Transformation1 * XYZ;

        //print the value
        //std::cout << " The NewCoordinate in the line is:
            "<< XYZ1.t() << std::endl;

        /*********************************/
        /*********************************/

        //calculate the haptic probe final position
        EndEffector[0] = vpMath::deg(XYZ1[0]);
        EndEffector[1] = vpMath::deg(XYZ1[1]);
        EndEffector[2] = vpMath::deg(XYZ1[2]);


    vpMatrix rotationXX;
    vpMatrix rotationYY;
    vpMatrix rotationZZ;
    vpMatrix rotRez1;

    rotationXX.eye(4,4);
    rotationYY.eye(4,4);
    rotationZZ.eye(4,4);
    rotRez1.eye(4,4);

    rotationZZ[0][0] = cos(vpMath::rad(EndEffector[2]));
    rotationZZ[1][0] = -sin(vpMath::rad(EndEffector[2]));
    rotationZZ[0][1] = sin(vpMath::rad(EndEffector[2]));
    rotationZZ[1][1] = cos(vpMath::rad(EndEffector[2]));
```

```
        rotationYY[0][0] = cos(vpMath::rad(EndEffector[1]));
        rotationYY[2][0] = sin(vpMath::rad(EndEffector[1]));
        rotationYY[0][2] = -sin(vpMath::rad(EndEffector[1]));
        rotationYY[2][2] = cos(vpMath::rad(EndEffector[1]));

        rotationXX[1][1] = cos(vpMath::rad(EndEffector[0]));
        rotationXX[2][1] = -sin(vpMath::rad(EndEffector[0]));
        rotationXX[1][2] = sin(vpMath::rad(EndEffector[0]));
        rotationXX[2][2] = cos(vpMath::rad(EndEffector[0]));

        rotRez1 = rotationXX * rotationYY * rotationZZ;

        for(int i=0; i<4; i++)
        {
            for(int j=0; j<4; j++)
            {
              //vtk_oMprobe_obj1->Element[i][j]=rotationX->
                  GetElement(i,j);
                vtk_oMprobe_obj1->Element[i][j]=rotRez1[i][j];
            }
        }

        probe_obj_transform1->SetMatrix(vtk_oMprobe_obj1); //
            initialize position transformation for probe CAD model
        probeActor1->SetUserTransform(probe_obj_transform1);

        renWin->Render();
      }

}
```

BIBLIOGRAPHY

[1] Prosit anr project website, 2011. URL http://w4.anr-prosit.fr/.

[2] Ieee 12th international conference on bioinformatics and bioengineering, 2012. URL http://bibe2012.cs.ucy.ac.cy/.

[3] Start in 3d with blender, 2012. URL http://www.siteduzero.com/tutoriel-3-11714-debutez-dans-la-3d-avec-blender.html.

[4] Get blender, 2012. URL http://www.blender.org/download/get-blender/.

[5] Qt (framework), 2012. URL http://en.wikipedia.org/wiki/Qt_(framework).

[6] F. Buzan and T Sheridan. A model-based predictive operator aid for telemanipulators with time delay. *in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 1: 138–143, 1989.

[7] G. Charron. Contribution à la commande bilatérale et à la gestion des configurations singulières pour le suivi de trajectoire d'un système télé-opéré : Application à la télé-échographie robotisée par satellite, Thesis, University of Orleans, 2011.

[8] A. Fonte, T. Essomba, P. Vieyres, J. Canou, Fraisse, S. Zeghloul, A. Krupa, and Arbeille. Robotic platform for an interactive tele-echographic system: The prosit anr-2008 project. *The Hamlyn Symposium on Medical Robotics*, 2010.

[9] B. Hannaford. Stability and performance tradeoffs in bi-lateral telemanipulation. *in Proceedings of the IEEE International Conference on Robotics and Automation*, 3:1764–1767, 1989b.

[10] D. Lawrence. Stability and transparency in bilateral teleoperation. *IEEE Transactions on Robotics and Automation*, 9(5):624–637, 1993.

[11] G Niemeyer and J.-J. E. Slotine. Designing force reflecting teleoperators with large time delays to appear as virtual tools. *in Proceedings of the IEEE International Conference on Robotics and Automation*, 3:2212–2218, 1997a.

[12] G Niemeyer and J.-J. E. Slotine. Using wave variables for system analysis and robot control. *in Proceedings of the IEEE International Conference on Robotics and Automation*, 2:1619–1625, 1997b.

[13] G Niemeyer and J.-J. E. Slotine. Telemanipulation with time delays. *The International Journal of Robotics Research*, 23(9):873–890, 2004.

[14] G Niemeyer and J.-J. E. Slotine. Telemanipulation with time delays. *The International Journal of Robotics Research*, 23(9):873–890, 2004.

[15] Passama R. *Architecture du robot PROSIT 1*, Rapport project ANR(confidential), 2012.

[16] G. Raju, G. Verghese, and T. Sheridan. Design issues in 2-port network models of bilateral remote manipulation. *in Proceedings of the IEEE International Conference on Robotics and Automation*, 3: 1316–1312, 1989.

[17] J. S. Sandoval Arévalo. Étude et validation des architectures bilatérales en robotique appliquée à la télé-échographie, Master Thesis, University of Tours, 2012.

[18] J. S. Sandoval Arévalo, P. Vieyres, C. Novales, L. Josserand, and A. Fonte. Architectures bilatérales pour un système de télé-échographie robotisé avec retour d'effort. *CNR IUT: National Conference. Tours*, March 2012.

[19] L. Sherwood. *Fundamentals of Physiology : A Human Perspective*. 2006.

[20] G. J. Tortora and N. P. Anagnostakos. *Principles of Anatomy and Physiology*. 6 edition, 1990.

[21] C. Zanella. *Modelli Geometrici*. 2 edition, September 2010. URL http://zanella.objectis.net/libro-modelli-geometrici/modelli-geometrici.