POLITECNICO DI MILANO

FACOLTÀ DI INGEGNERIA DEI SISTEMI
Corso di Laurea in Ingegneria Biomedica

Dynamic Tool Compensation in a Bilateral Telemanipulation System for Neurosurgery

Relatore:
Elena DE MOMI, PhD

Correlatori:
Danilo DE LORENZO, PhD
Dott.ssa Elisa BERETTA

Tesi di Laurea di:
Veronica PENZA
Matricola 766232

Anno Accademico 2012-2013

Dynamic Tool Compensation in a Bilateral Telemanipulation System for Neurosurgery

Veronica Penza

October 3, 2013

Abstract

Minimally invasive surgery (MIS) procedure has influenced modern surgery by decreasing invasiveness, minimizing patient recovery, time and cost. However, surgical procedures using long tool inserted through small ports on the body deprive surgeons of the depth perception, dexterity, sense of touch, and straightforward hand-eye coordination that they are accustomed in open surgery. Robotic Minimally Invasive Surgery (RMIS) via Telemanipulation is a promising approach to overcame this drawbacks. Telemanipulation system provides the capacity to perform surgery on a patient at a remote location. It enables a human operator to interact with a remote environment and perform his task with the aid of master and slave devices.

Telemanipulation offers several key potential advantages over traditional MIS:

- improves accuracy filtering of unnecessary or unwanted surgeon movements, especially hand tremor;
- provides advantages giving the surgeon the option of scaling down macroscopic hand motions or scaling up remote forces;
- provides articulated wrist-like micro-robotic or actuated instruments attachment at the end of the instruments for conducting micro-nano scale surgery with additional degrees of freedom (DoFs) that give back full dexterity inside the human body;
- The surgeon may become less tired during long surgery when sitting at a comfortable and ergonomic console rather than having to stand by the bedside;

Currently the research on the telemanipulation system in neurosurgery focuses the attention on the force feedback and of how important it is for the surgeon to perceive the sensation of touch away from the surgical site. Loss of haptic feedback in RMIS procedure could be a disadvantage to surgeon, since they are conventionally used to palpating tissue to diagnose tissues as normal or abnormal. Moreover, without environmental force feedback, the surgeon would need to interpret the tissue deformation and behaviour to judge the force exerted on the environment, which consumes time and causes mental fatigue. Therefore, the need exists to incorporate force feedback into Telemanipulation Surgery and provides an excellent opportunity to improve the quality of the surgical procedure. In the current robotic neurosurgical system, only neuroArm enhances haptic feedback.

In a telemanipulation system, the time delays in the communication is an important issue subject to research, due to the exchange of information between the master and slave devices, in terms of pose or force. Delays exist for information transport and data processing. Surgeon must be aware that communication delays may introduce problems to their system. As results the performance of the teleoperated system may deteriorate drastically and may possibly become unstable.

In this project we present a bilateral telemanipulation system, which consists of a 6 Degrees of Freedom (DoFs) haptic interface (PHANToM Omni, Geomagic, Morrisville, NC 27560, USA) as master device and 7 DoFs robotic arm (LightWeightRobot 4+, KUKA, Augsburg, Germany) as slave device. A surgical instrument is placed on the slave device and the external force sensor (Nano25, ATI Industrial Automatation, Goodworth Dr. | Apex, NC 27539 USA) has the purpose of measuring interaction forces and torques of the surgical instrument with the tissues. These forces are reproduced by the haptic interface on the hand of the surgeon. The force sensor cannot be placed on the tip of the surgical tool to prevent electronics from entering the patient and for reasons of size and sterility. Hence, in the bilateral telemanipulation system presented it is located between the end-effector of the slave arm and the surgical instrument. However, in this configuration the force sensor, during a surgical procedure, does not feel only the contact forces of the surgical tool with the tissues, but in addition it feels also the weight of the tool, which is dynamically exerting force on the force sensor. The overall forces measured (weight of the tool and interaction with the environment) are reproduced by the haptic interface.

The aim of the presented thesis is to dynamically compensate the weight of a generic tool measured by the force sensor during a telemanipulation procedure, in order to feel on the haptic interface only the contact forces of the instrument with the tissues. The dynamic tool compensation algorithm described estimates the force and torque exerted by a known geometry tool attached to the force sensor and removes them form the overall force and torque measured by the force sensor. The equations used to estimate the forces and torques generated by a generic surgical instrument are derived from the basic laws of dynamics that describe the motion of a rigid body due to external forces and torques. Once the force measured by the force sensor is compensated, it is sent to the haptic interface, in order to be reproduced at the hand of the surgeon.

The analysis executed on the telemanipulation system evaluates the communication delays and it is carried out recording the positon of both slave and master end-effector with an optical tracking system. It evaluates the communication delay between the movement controlled by the haptic interface and the movement tracks by the slave robot.

Experimental trials on dynamic tool compensation are executed in static and dynamic condition. In the dynamic trials the slave robot with the force sensor and the tool attached to the end-effector, performs linear and angular movements. The force and torque estimation error is evaluated in both conditions. It is also evaluated, in static condition, the presence of correlation between the estimation error committed and the orientation of the force sensor and, in dynamic condition, the correlation between the acceleration of the movement and the estimation error.

Results show that the presented bilateral telemanipulation system allows the dynamic compensation of a generic tool with estimation errors negligible considering the accuracy of the force sensor. While it is not possible to define a correlation between the estimation error and the force sensor orientation in static condition, there is an high positive correlation between the estimation error and the acceleration of the movements in dynamic condition. The more is high the acceleration, the more is the estimation error. The communication delays measured is below the 0.50ms.

This work was realized at Nearlab, Politecnico di Milano.

Keywords: dynamic tool compensation, telemanipulation system, haptics, communication

Sommario

La chirurgia mininvasiva ha influenzato la moderna chirurgia introducendo molti vantaggi, tra cui la minor invasività, riduzione del tempo di ricovero del paziente e conseguente diminuzione dei costi. Tuttavia, le procedure chirurgiche in cui si ultilizzano strumenti lunghi inseriti in piccole aperture sul corpo del paziente impediscono al chirurgo di percepire il senso di profondità, di tatto, la destrezza nell'eseguire il movimento e la coordinazione mano occhio, sensazioni invece comuni nella chirurgia aperta.

La chirurgia mininvasiva robotica, intesa come Telemanipolazione, è vista come una promettente soluzione ai problemi sopracitati. Un sistema di telemanipolazione robotico permette al chirurgo di eseguire il task chirurgico sul paziente da remoto, attraverso l'utilizzo di un robot *master* ed di un robot *slave*, che interagisce con l'ambiente chirurgico esattamente come gli viene comandato dal robot *master*.

La telemanipolazione offre potenziali vantaggi rispetto alla tradizionale chirurgia mininvasiva, tra cui:

- aumento dell'accuratezza del movimento filtrando movimenti della mano del chirurgo non necessari o non voluti, come ad esempio il tremore della mano;
- possibilità di scalare il movimento o la forza di ritorno all'interfaccia aptica: riducendo movimenti mascroscopici della mano o aumentando forze di interazione con i tessuti;
- possibilità di aggiungere strumenti chirurgici che incrementano i gradi di libertà, consentendo maggiore destrezza nel movimento o che permettono di eseguire micro-nano chirurgia;
- il chirurgo ha maggiore ergonomia e comfort nell'eseguire lunghi interventi seduto in una stazione di lavoro, piuttosto che restare molte ore accanto al lettino operario. Questo gli permette un minor affaticamento fisico.

Attualmente la ricerca sui sistemi di telemanipolazione in neurochirurgia focalizza l'attenzione sul ritorno di forza e su quanto sia importante per il chirurgo percepire a distanza la senzazione del tatto.

La mancanza di ritorno aptico in chirurgia mininvasiva robotica è uno svantaggio per il chirurgo, poichè convenzionalmente è abituato a palpare il tessuto per valutare la patologia e come agire di conseguenza.

Inoltre, senza ritorno di forza il chirurgo deve interpretare la deformazione e il comportamento del tessuto con cui è a contatto per capire quali forze sta esercitando, inducendo così fatica mentale e consumo di tempo.

Per i motivi precedentemente illustrati, si sente la necessità di introdurre e valorizzare il ritorno di forza in sistemi di telemanipolazione, mirando a migliorare la qualità della procedura chirurgica.

Attualmente, in neurochirurgia è stato sviluppato e testato su essere umano solo un sistema con ritorno di forza, neuroArm.

In un sistema di telemanipolazione, però, lo scambio di informazione tra *master* e *slave*, in termini di posa o forza, può essere accompagnato da un ritardo di comunicazione, dovuto sia alla trasmissione dei dati su rete locale, che all'elaborazione dei dati stessi. Il chirurgo deve essere a conoscenza di questo ritardo, in quanto questo può comportare deterioramento drastico del sistema di telemanipolazione e probabile instabilità del sistema.

In questo progetto presentiamo un sistema di telemanipolazione costituito da: un'inter faccia aptica a 6 gradi di libertà (gdl) con il ruolo di robot *master* (PHANToM Omni, Geomagic, Morrisville, NC 27560, USA), e un braccio robotico a 7 gdl che si comporta da robot *slave* (LightWeightRobot 4+, KUKA, Augsburg, Germany). Lo strumento chirurgico usato per eseguire il task è posto sul robot *slave* e il sensore di forza (Nano25, ATI Industrial Automatation, Goodworth Dr. | Apex, NC 27539 USA) ha la funzione di misurare le forze e le coppie di interazione dello strumento chirurgico con i tessuti circostanti. Queste forze sono riprodotte dall'interfaccia aptica sulla mano del chirurgo. Il sensore di forza non può essere posto sulla parte terminale dello strumento chirurgico per motivi di dimensione, sicurezza e sterilità. Così, nel sistema di telemanipolazione presentato, è situato prima dello strumento chirurgico. Tuttavia, in questa configurazione, durante una qualsiasi procedura chirurgica telemanipolata il sensore di forza misura non solo le forza di interazione dello strumento con i tessuti, ma anche il peso stesso del tool, che genera delle force sul sensore durante qualsiasi movimento. L'insieme di queste forze (dovute al peso dello strumento e al contatto con i tessuti) sono riprodotte dall'interfaccia aptica. Lo scopo di questo progetto è stimare le forze dovute al peso del tool misurate dal sensore di forza durante una procedura di telemanipolazione e compensarle dinamicamente, con il fine di far percepire al chirurgo solo le forze di interazione con l'ambiente. L'algoritmo di compensazione dinamica sviluppato stima forze e coppie esercitate da un generico strumento rigido sul sensore di forza in condizioni dinamiche e sottrae queste alle forze misurate dal sensore di forza. Una volta che le forze vengono misurate e compensate, sono pronte per essere inviate e riprodotte dall'interfaccia aptica.

L'analisi eseguita per valutare il ritardo di comunicazione del sistema di telemanipoazione prevede che un sistema di localizzazione ottico registri la posizione dello strumento chirurgico posto sullo *slave* e la posizione dell'organo terminale del *master*.

Test sulla compensazione dinamica del tool sono stati eseguiti sia in condizioni statiche che dinamiche. In condizioni dinamiche gli esperimenti prevedono movimenti lineari ed angolari dello *slave* con il sensore di forza e lo strumento chirurgico posti sull'organo terminale. L'errore della stima della forza e della coppia è valutato in entrambe le condizioni. Inoltre è stato valutato, in condizioni statiche, la presenza di correlazione tra l'errore della stima commesso e l'orientamento del sensore di forza e, in condizioni dinamiche, una correlazione tra l'errore di stima e l'accelerazione del movimento eseguito.

Risultati mostrano che il sistema di telemanipolazione presentato permette di eseguire procedure di telemanipolazione con ritorno di forza e compensazione dinamica di un generico strumento chirurgico, con un errore di stima trascurabile, inferiore all'accuratezza del sensore di forza. Mentre non è possibile stabilire una correlazione tra l'errore di stima e l'orientamento del sensore di forza in condizioni statiche, è invece molto forte la correlazione tra l'errore di stima e l'accelerazione del movimento: all'aumentare del valore assoluto dell'accelerazione, aumenta l'errore. La mediana del ritardo misurato è 50 ms.

Il lavoro di tesi è stato svolto presso il laboratorio Nearlab, Politecnico di Milano.

Parole chiave: compensazione dinamica dello strumento, sistema di telemanipo-

lazione, interfaccia aptica, comunicazione

Contents

List of Figures

List of Tables

List of Acronyms

MIS	Minimally Invasive Surgery
RMIS	Robotic Minimally Invasive Surgery
DoF	Degree of Freedom
3-D	3-Dimensional
PP	Position-Position control architecture
FP	Force-Position control architecture
PF	Position-Force control architecture
FF	Force-Force control architecture
ee	end-effector
HI	Haptic Interface master device
LWR	slave device
FS	Force Sensor
ROS	Robot Operating System
OROCOS	Open RObot COntrol Software
API	Application Programming Interface
FRI	Fast Research Interface
UDP	User Datagram Protocol
HDAPI	Haptic Device API
fft	fast fourier trasform
Best fit FOAW	Fit-First Order Adaptive Windowing Filter
DRF	Dynamic reference Frame
rmse	root mean square error

Chapter 1

Introduction

1.1 Robotic Surgery: telemanipulation scenario

Robots are becoming increasingly relevant to neurosurgeons, in particular extending neurosurgeon's physical capabilities in minimally invasive intervention.

Many are the advantages of MIS interventions compared to open surgery, such as reduced pain and trauma, shorter hospitalization, shorter rehabilitation time and cosmetic advantages.

However, surgical procedures using long tool inserted through small ports on the body deprive surgeons of the depth perception, dexterity, sense of touch, and straightforward hand-eye coordination that they are accustomed in open surgery[6][7].

Robotic Minimally Invasive Surgery (RMIS) via Telemanipulation is a promising approach to overcame this drawbacks.

"Telemanipulation is the extension of human sensing and manipulation capabilities by coupling it to remote artificial sensors and actuators." Sheridian (1992)*[8]* It extends the human capability to manipulate object remotely by providing the operator with similar conditions as those at the remote location. This is achieved via installing a similar manipulator or joystick, called *master*, at the human's hand that provides motion commands to the *slave*, which is performing the actual task. In a general setting, the human imposes a force on master manipulator which in turn results in a displacement that is transmitted to the slave, which mimics that movement. [9]

The human operator, sitting in the work-station, is always in the control loop, he should be able to interpret the task to be executed, makes the decisions and defines the control strategies for the manipulation. He has to learn, to understand and to interpret the dynamics of the teleoperator, its environment and the possible resulting disturbances. Furthermore, via display, visual/ auditory and/or tactile, the operator is able to get information and to see the consequences of actions to be taken.

The Figure 1.1 shows a telemanipulation system scenario.

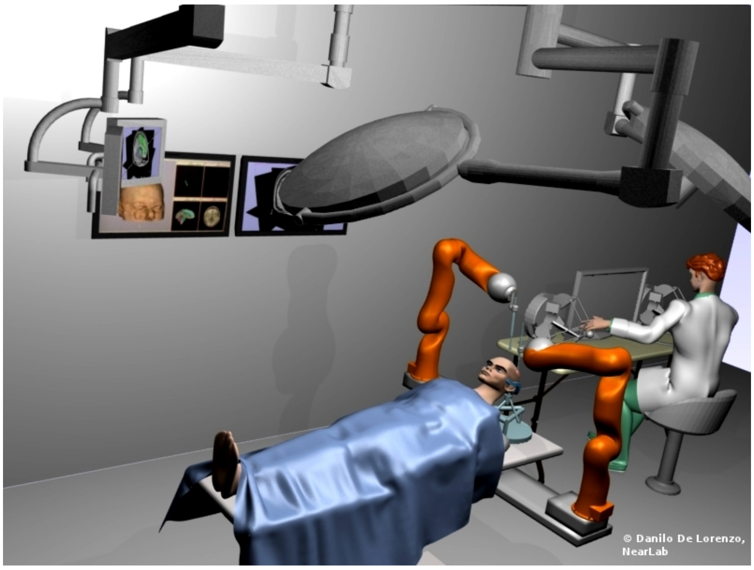


Figure 1.1: Example of telemanipulation system in operating room: the surgeon sits at the workstation and works on the surgical field through the master devices at distance. The slave manipulators, tracking the movements of the master manipulators, performs the surgery. http://www.active-fp7.eu/

Telemanipulation offers several key potential advantages over traditional MIS[10][6]:

- Filtering of unnecessary or unwanted surgeon movements, especially hand tremor, improves accuracy;
- Giving the surgeon the option of scaling provides advantages: scaling down of local macroscopic hand motions can result in remote microscale (or even nanoscale) movements; scaling up of remote forces (using haptic) could theoretically make brain tissue feel macroscopic, e.g., feel like moving rocks, to the surgeon if desired ;
- Teleoperation offers the option of using articulated wrist-like micro-robotic attachment at the end of the instruments for conducting micro-nano scale surgery [11] or actuated instruments with two additional Degrees of freedom (DoFs) that give back full dexterity inside the human body;
- The surgeon may become less tired during long surgery when sitting at a comfortable and ergonomic console rather than having to stand by the bedside;

Using robotics also give one the ability to interface with virtual environments and other software and computer techniques such as intraoperative imaging, which facilitates planning, training, and navigation tasks [12].

Improvements in RMIS system will lead to significant societal impacts through better patient care, reduced morbidity, shorter hospital stays, reduced trauma, faster recovery times, and lower health care cost.[7]

In spite of the aforementioned advantages, currently in telemanipulation tasks several issues are subject to research [8]:

- the lack of tactile and touch informations, and consequently the mismatch with proprioceptive feedback.

- the lack of information for depth perception, since visual display usually generate 2D data.
- due to the fact that manipulations or operations have to be performed at distance, there is always a time delay in the human operator-telemanipulator control loop, which may yield severe problems with reference to the stability of the control loop.

1.2 Aim of the work: dynamic tool compensation

When a surgeon performs surgery at the workstation of the bilateral telemanipulation system, he wants to feel as if it was directly on the surgical field. The force feedback allows him to recognize tissue inflammation, to detect solid or hollow organ masses, and has the aim of reproduce exactly the force involved in the contact of the surgical instrument with the environment. The haptic interface and the force sensor play a significant role. A problem may be the positioning of the force sensor: it cannot be placed on the tip of the surgical tool to prevent electronics from entering the patient and for reasons of size and sterility. Hence, it can be located between the end-effector of the slave arm and the surgical instrument, as shown in Figure 1.2.

However, in this configuration, the force sensor, during a surgical procedure, does not feel only the contact forces of the surgical tool with the tissues, but in addition it feels also the weight of the tool, which is dynamically exerting force on the force sensor. The overall forces measured (weight of the tool and interaction with the environment) are reproduced by the haptic interface.

The aim of the presented thesis is to dynamically compensate the weight of a generic tool measured by the force sensor during a telemanipulation procedure, in order to feel on the haptic interface only the contact forces of the instrument with the tissues.

This work was realized at Nearlab, Politecnico di Milano.

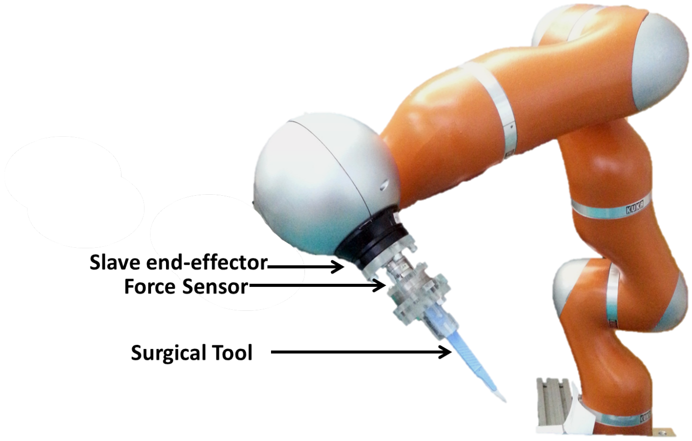


Figure 1.2: Example of surgical instruments (scalpel) attached to the distal part of the force sensor, placed at the end-effector of the slave devices.

1.3 Outline of the thesis

The thesis is organized as follows:

Chapter_2 shows the current bilateral telemanipulation system, enhances the role of force feedback and haptic interface in surgery and summarizes the state of the art in the estimation of the contact forces to reproduce on the master side to the surgeon at the work-station.

Chapter_3 presents the materials and the devices involved in the telemanipulation system and the developed methods used for the dynamic compensation of the surgical tool attached to the external force sensor placed on the slave arm, together with the experimental protocol and the performed analysis.

Chapter_4 reports the experimental results of the performed analysis.

Chapter_5 concludes the thesis and presents possible future developments for the presented work.

Chapter 2

State of the Art

2.1 Robotics in neurosurgery

The first reported use of a robot in neurosurgery was in 1985 by Kwoh and colleagues [13], who employed a Programmable Universal Machine for Assembly ([PUMA]; Advanced Research Robotics, Oxford, CT) industrial robot for holding and manipulating biopsy cannulae. Although the robot served only as a holder/guide, the potential value of robotic systems in surgery was evident. In 1991, Drake and coworkers [14] reported the use of a PUMA robot as a retraction device in the surgical management of thalamic astrocytomas. Despite their novel application, both systems lacked the proper safety features needed for widespread acceptance into neurosurgery. Beginning in 1987, Benabid and associates [15] experimented with an early precursor to the robot marketed as NeuroMate (Integrated Surgical Systems, Sacramento, CA). NeuroMate uses preoperative image data to assist with surgical planning and a passive robotic arm to perform the procedure. The NeuroMate system has been used in >1,000 cases.

These first neurosurgical robots relied on preoperative images to determine robotic positioning. As a result, surgeons could not dynamically monitor needle placement under image-guidance and were blind to changes such as brain shift. To satisfy the need for a real-time, image-guided system, Minerva was developed (University of Lausanne, Lausanne, Switzerland). The system consisted of a robotic arm placed inside a computed tomography (CT) scanner, thus allowing surgeons to monitor the operation in real-time and make appropriate adjustments to the trajectory as needed [16]. Despite considerable engineering challenges, the design and construction of magnetic resonance (MR)–compatible robotic systems soon followed.

The majority of time in neurosurgical cases is spent on micromanipulation. Currently, teleoperated system dedicated to neurosurgery, targeted to assist and improve the micromanipulation and capable of complex procedure are Neurobot [1], that does not used force feedback, ROBOCAST [17] and neuroArm [3][18], that include haptics. Of these, none are commercially available. Only NeuRobot and neuroArm have undergone human clinical trials, and only neuroArm continues doing.

2.1.1 NeuRobot

In Japan, Hongo and colleagues [19] develop NeuRobot, a robot platform for telecontrolled microneurosurgery through the portal of an endoscope.

NeuRobot is a master-slave manipulator system, and it is designed so that a surgeon operates the slave manipulator by controlling three levers on the operation-input device, master manipulator, while watching a three-dimensional monitor. The micromanipulator, part of which is inserted into the operating field, has a thin tubular cylinder of 10-mm diameter, and three micro-instruments of 1-mm tip are installed therein. Each microinstrument has three degrees of freedom: rotation, neck swinging, and forward/backward motion. There are five small holes in the tubuar cylinder available for irrigation or suction. Various type of microinstruments can be used as surgical tool including micro-forceps, micro-hook, micro-needle, micro-tip. Investigators perform neurosurgery on a cadaveric head and conclude that the system facilitates more accurate and less invasive surgery [19]. NeuRobot is subsequently used to remove a portion of a tumor from a patient with a recurrent, atypical meningioma [20].

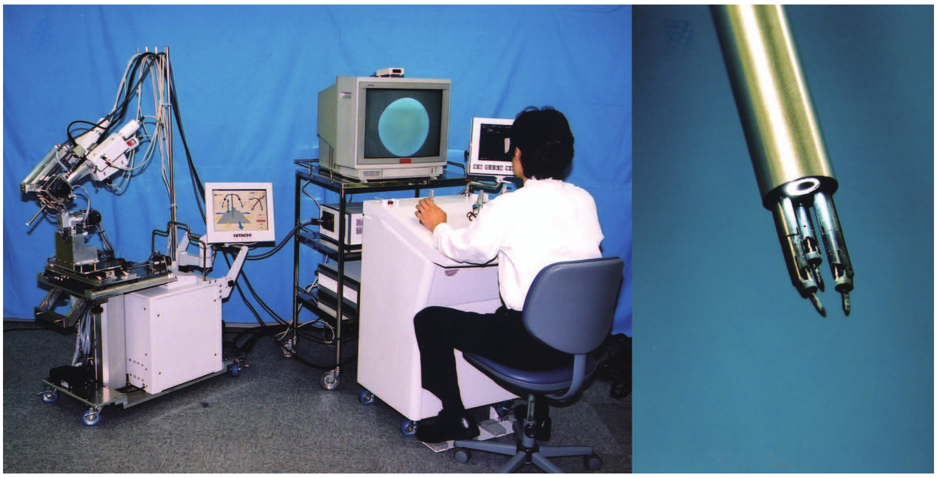
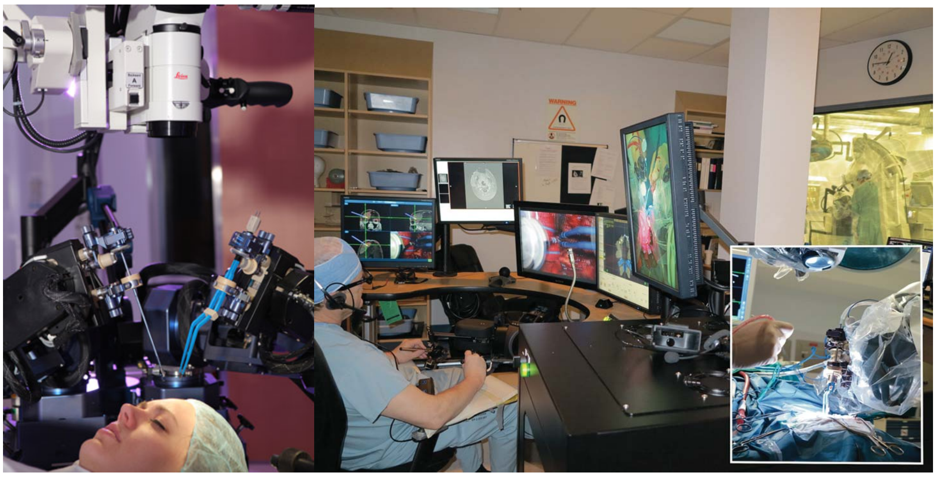


Figure 2.1: Photograph of the NeuRobot system consisting of operating console, monitor and slave manipulator fixed in the supporting device. [1]

2.1.2 neuroArm

nauroArm is a teleoperated magnetic resonance-compatible image-guided robot. The neuroArm system comprises two robotic arms capable of manipulating, both designed and existing microsurgical tools, connected via a main system controlled to a workstation with a sensory immersive human-machine interface. The surgeon is positioned at a workstation and uses the human-machine interface to interact with the surgical site. The human-machine interface provides both MR images and real-time high-definition three-dimensional (3-D) images of the surgical site (Figure 2.2). Modified haptic interface hand controllers enable the manipulators to emulate the surgeon's hand movements while providing the surgeon with haptic force feedback. The displacement of the hand controller is mimicks by the robot manipulators. The interaction force between the surgical tool and the environment, measured by two force sensors mounted on each arm, is applied to the surgeon's hand by the haptic device.

The neuroArm system allows the surgeon to access imaging data without interrupting surgical procedure while providing the surgeon with advanced sensory input available to facilitate the performance of surgery. Operating within real-time images means that surgeons can correct for brain shift and can ensure the complete removal of a tumor during operation. [3][18]



(*a*) (*b*)

Figure 2.2: (*a*) neuroArm structure [2] (*b*) The neuroArm workstation is remote from the operating room. Inset, the neuroArm robot with bipolar forceps in the right manipulator and suction in the left.[3]

2.2 Bilaterial Telemanipulation Control scheme

In telemanipulation system, there is an information exchange between the human operator and the remote environment. The information can be in the form of either position or force. Based on which information is exchanged, the control architectures are divided into four types, such as position-position control architecture (PP) (in Figure 2.3-(*a*)), force-position control architecture (FP) (in Figure 2.3-(*b*)), position-force control architecture (PF) (in Figure 2.3-(*d*)) and force-force control architecture (FF) (in Figure 2.3-(*c*)). All these four control architectures are also known as 'two channel control architectures'. The 2-channel architecture allows transmission of one signal and reflections of one signal in teleoperation, predominantly either PP or PF. If both position and force information are exchanged between master and slave, the control architecture is called 'four channel control architectures'.[21]

In a PP there are no force sensor measurements , and the controller tries to minimize the difference between the haptic interface and the robot manipulator (end-effector) position, thus reflecting a force proportional to this difference to the user once the slave makes contact with object. PP control achieves relatively good position tracking between the haptic interface and manipulator, but its force tracking performance is poor [10].

In a PF architecture, a force sensor is used to measure the interaction between the remote robot and the environment for reflection to the user while the robot tracks the position of the haptic interface. PF control achieves relatively good position tracking between master and slave while its force tracking performance is perfect [22]. Neither of the above schemes achieves full transparency.

The FF control architecture has the disadvantages of position-error accumulation between the master device and the slave device [23]. The position-force control architecture has practical difficulties in implementation and hence has not been used until now [23].

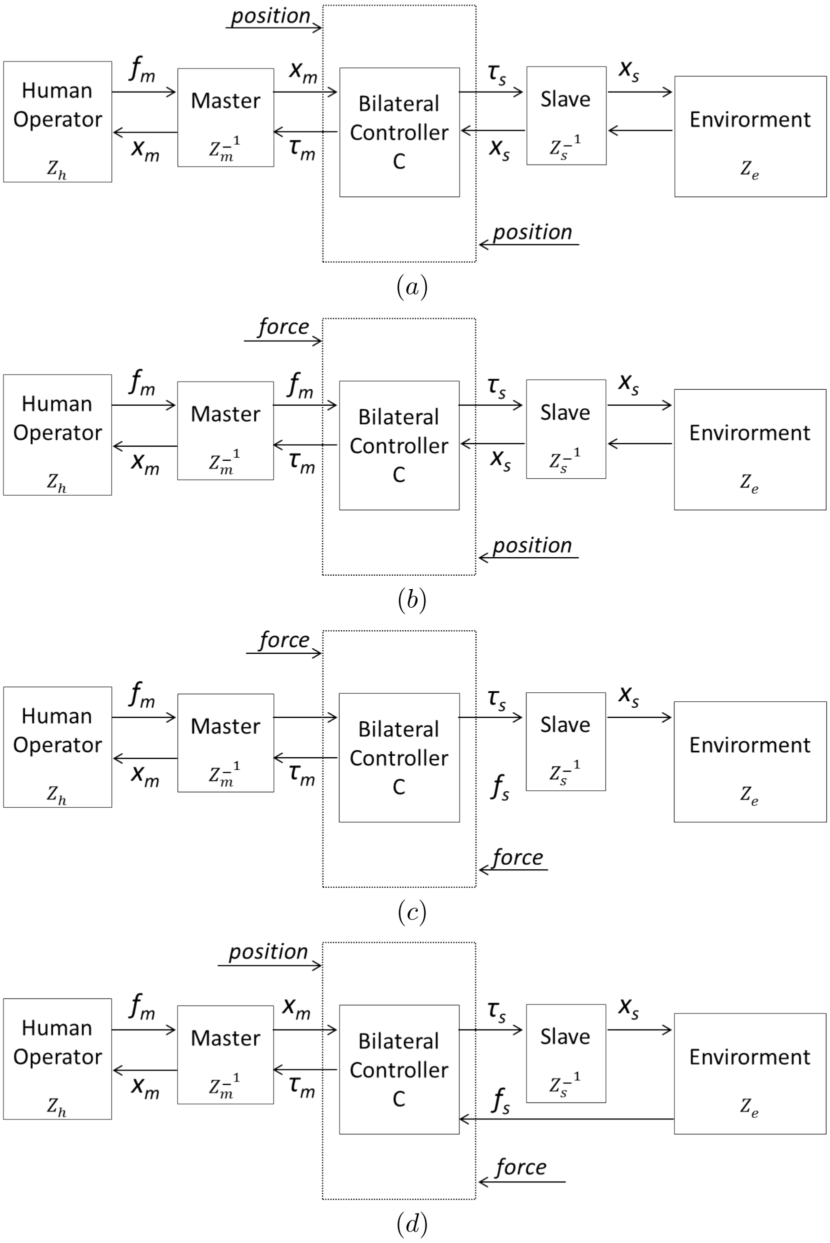
A scheme of the FP control architecture is shown in Figure 2.3; f_m, x_m, f_s, x_s are respectively the force of the master, the position of the master, the force of the slave, and the position of the slave. Z_m and Z_s represent the dynamic models of master and slave, and the control input torques to the master and the slave are τ_m andτ_s .Impedance models of the human operator and the environment are represented as Z_h and Z_e.

However, the exchange of information can lead to time delays in the communication between local and remote manipulator: a challenge to engineers and surgeons alike. [9]. Surgeon must be aware that communication delays may introduce problems to their system. In a telesurgical system, delays exist for information transport and processing. Information from continuous-output sensors must be digitized before computer processing, and even information in digital form must be re-encoded for efficient transmission.

Signal-processing techniques for accomplishing this with minimal impact on signal quality are well established, and the effects (like aliasing, in which a continuous signal cannot be properly reconstructed from its digital representation) should be unnoticeable in commercial systems.

However, the practical impact of filtering and computer encoding/ decoding of signals is the introduction of significant time delays, even in nearby teleoperation systems. In long-distance teleoperation, even larger transport delays occur as a result of the limited speed of the electric signals and limited capacity of wires or wireless systems for encoding information.

Thus, a telesurgical system compounds delays inherent to the “reaction time” of both the electromechanical system itself and the surgeon who operates it. As the total delay in the system increases, it becomes increasingly difficult to control the slave. If the delay becomes too large, then the surgeon must adopt a “move and wait” strategy that increases workload [24].



(a)

(b)

(c)

(d)

Figure 2.3: Control signal flow diagrams for the two-channel control architectures. In (*a*) position-position architecture, (*b*) position-force architecture, (*c*) force-force architecture, (*d*) force-position architecture.

2.3 The sense of touch: Haptic Interface

The FP control requires force information to feedback to the surgeon, which also would provide haptic cues to the surgeon for effective surgery.

The terms haptics relates to the sense of touch. The sense of touch gives us information on the material properties of an object, including stiffness (elasticity), texture, and weight, as well as shape properties such as size, orientation, and curvature. In the active exploration of objects, humans identify texture through lateral motion, hardness by applying pressure, temperature through static contact, weight by unsupported holding, global shape and volume through enclosure by fingers, and exact shape by following the object contours.

In surgery field, the surgeons need this sense, in fact in open surgery have traditionally relied on the combination of their tactile and visual abilities to diagnose tissue as normal or abnormal. However, current robotic teleoperator surgical systems provide features for visual feedback, but are incapable of providing force feedback. Surgeons who operate using these robotic assistants do not have force and tactile feedback from the operative site as they are indirectly in contact with the surgical site instrumented tool.

Hence, the need to incorporate force feedback capabilities into robotically assisted procedures, provides an excellent opportunity to improve the quality of surgical procedures. This opportunity is allowed by the haptic interface.

Haptic interface [25] is a human-machine interface, that enables a person to explore an environment indirectly through a robotic proxy. It is able to display both position and force in 3D. In a physical environment, the proxy consists of a mechanical device capable of effecting changes in its surroundings, often in the form of an electrically powered robotic manipulator with a tool gripper.

Most haptic bilateral configurations used in surgery to date transmit commanded movements to the remote proxy and reflected forces back to the haptic master interface [26].

2.4 The role of force feedback in surgery

In the context of a robot-assisted surgery, various studies confirm that haptic feedback generally improves performance in terms of task success rate, completion time, economy of exerting force, and less trauma to tissue [27].

These haptic feedback benefits are:

Scaling Enhances Precision The precision of the surgeon limits endoscopic, stereotactic, and microneurosurgical procedures, determined by his inherent visual acuity and fine motor control. Not only do these qualities differ from person to person, but their maximum bounds limit the level of precision with which a surgeon can interact with delicate, minute cranial structures. Neurosurgeons operating with telesurgical systems, however, routinely achieve sub millimeter precision when their movements are scaled down for the robotic manipulator compared with the millimeter precision generally achieved by hand [28].

Filtering Reduces the Effects of Hand Shake Even though neurosurgeons are highly trained, finely tuned professionals, it is difficult to always perform all procedures without making any unnecessary or unwanted hand movements within the surgical corridor. Physiological tremors in particular range naturally from 8 to 12 Hz, which presents an impediment to microsurgery [29]. Telesurgical systems can recognize and remove such undesirable movements, improving surgeon performance and increasing patient safety.

Virtual Fixtures Reduce Surgeon Workload and Increase Accuracy Some procedures or portions of procedures require a surgeon's motion to be restricted to a single direction, e.g., in biopsies when the needle must be inserted directly in a straight line from the cranial opening to the abnormal tissue to best preserve surrounding structures. Telesurgical systems semi automate such procedures by ignoring all movements of the surgeon's hands that do not follow the predetermined straight-line trajectory. This so-called z-lock function, which is also referred to as the incorporation of virtual fixtures, allows the surgeon to focus on one direction at a time without concern for extraneous movement, resulting in increased accuracy and decreased workload [30].

The absence of haptic feedback can lead to underestimated or unrecognized tissue inflammation, inability to detect solid or hollow organ masses, or accidental puncturing of blood vessels or tissue damage, and it has quite often been argued that visual information alone is highly subjective and can deteriorate because of fluids from the patient's body on the camera lens, which makes it a potential safety concern.[21]

2.5 The need of surgical instrument compensation

Bilateral control methods in teleoperation are effective, and human can feel reaction force from the environment through the haptic interface.

One way out is to estimate the interaction force between the slave robot and the environment so that it can feedback to the haptic master interface to provide haptic cues to the surgeon.

There have been few studies to estimate the environment force in the telerobotic systems. Katsura et al. [31] used disturbance observers for estimating environmental force for teleoperators, pointing out its advantage due to the large bandwidth requirement for force transmission during teleoperation. However, their method requires an accurate nominal robot model and assumes negligible modelling error. Also, the robot is assumed to operate under constant velocity, to neglect other uncertainties in the robot and the friction is assumed to have been accurately identified, which are very big assumptions.

One common way of estimating environmental interaction force is based on the deflection produced on a known environmental model. Haddadi et al. [32] propose a method for the online estimation of the Hunt-Crossley model, which can estimate the non-linear properties of the soft tissues. Yamamoto et al. [33] compare seven possible mathematical tissue models, using self-validation and cross-validation studies, and claim that the Hunt–Crossley model can best describe the soft tissue characteristics. They also compare various estimation techniques and claime that the recursive least-squares method is highly appropriate for surgical environments [?]. The demerit of the above methods is that different models that can describe the characteristics of the varied soft and hard environments need to be chosen, and a common method for dealing with all kinds of environments is difficult to apply.

Another innovative way to feedback that forces is the use an external force sensor placed at the end-effector of the slave. Any kind of surgical tools can be attached at the tool side of the force sensor, and the forces measured are replicated on the master side.

Placement of force and/or tactile sensors at the tool tip provides the most accurate way to quantify forces and sensations. However, this requirement for the sensors to enter a patient's body demands a level of advancement not yet found in sensor technology. The design of these sensors must address biocompatibility and sterilizability issues and, in the context of minimally invasive surgery (and particularly micro neurosurgery), must conform to rigorous constraints on size, weight, shape, and sensitivity.

The solution is to placed the force sensor between the end-effector (ee) and the surgical instruments.

The fact that the surgical tool is located after the force sensor, leads to a problem: the human operator feels not only the contact force/torque of the tool with the environment, but also the force/torque exerted on the force sensor due to the mass of the tool. That force interferes with the manipulation of the object when the mass is too large.

Zemiti et al. [34] propose a method to measure the environmental interaction force using a force-sensor placed outside the body of the patient and claim that it was not affected by the unwanted friction effects. However, their method was based on an assumption that the dynamic wrench is negligible and complete gravity compensation is possible. [21].

Disturbance observers have also been used for gravity estimation [35], in which gravity, as already mentioned, interferes with the manipulation of large objects. However, the position error between the master and slave robots may emerge in this case.

The on-line dynamic compensation of the tool's weight acting on the force sensor solves the problem, allowing the system to reflect back to the master the real force of interaction with tissues.

Chapter 3

Materials and Methods

The proposed bilateral telemanipulation system consists of an haptic interface master device (HI), the PHANToM Omni (Geomagic, Morrisville, NC 27560, USA), a slave device (LWR), the LightWeightRobot 4+ (KUKA, Augsburg, Germany), and an external force/torque sensor (FS), the Nano25 (ATI Industrial Automatation, Goodworth Dr. | Apex, NC 27539 USA).

The Figure 3.1 shows the telemanipulation system scenario. LWR is placed beside a surgical table on which the surgery is performed, while the master is on a remote workstation. The force feedback is allowed by a force sensor attached to the end effector (ee) of the slave: it measures the interaction force of the surgical instrument with the tissue. We suppose that the surgeon at the workstation should feel as if he were running a task directly on the surgical field, in terms of pose and force information.

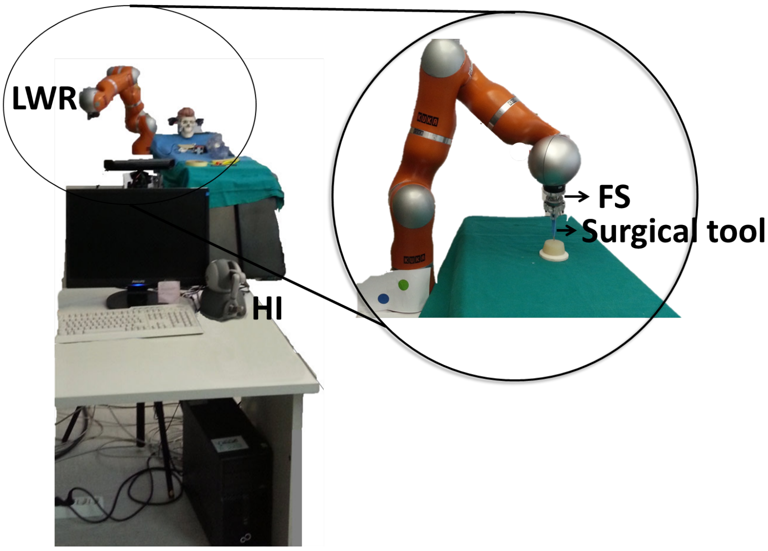


Figure 3.1: Telemanipulation scenario: the workstation consists of the HI, while the LWR works on the operating table. The FS is placed between the ee and the scalpel.

3.1 Telemanipulation system architecture

The overall system is organized as shown in Figure 3.2, the grey blocks represents the hardware components, master and slave devices and the force sensor, while the other blocks are the software components, connected using a distributed architecture; there are many applications, cooperating with each other, that reside on multiple computing nodes. As shown in 3.2, the software interfaces, *Haptic Interface*, *FRI*, *Force Sensor*, allow the communication with the hardwares, read pose information from master and slave devices, force/torque information from force sensor and set force on the haptic interface. The other software components, *Tool Dynamic Compensation*, *Telemanipulation Manager*, *Position Controller* process the data necessary in a bilateral telemanipulation system: compensate dynamically the weight of the surgical tool attached at the ee, manage the mapping involved in the telemanipulation and control the robot.

The software components are developed in C++, using Robot Operating System (ROS) and Open RObot COntrol Software (Orocos) framework (Section 3.3.1 and 3.3.2), which provide libraries and tools to help software developers to create robot application. The communication protocols differ depending on the framework and hardware used. The Nano25 give data through a RS-232 serial port, using an asynchronous serial protocol; the lwr4+ communicates via ethernet through the UDP protocol; the PHANToM Omni communication interface is IEEE-1394 Fire Wire port. The different communication protocols are shown in Figure 3.2.

The architecture runs on Linux operating system, because ROS and Orocos currently only runs on Unix-based platforms. Morover, we use Xenomai, a real-time development framework cooperating with the Linux kernel, that provides an hard real-time support to the applications.

In the next paragraphs will be described in details the hardware (Section 3.2) and software (Section 3.3) components of the bilateral telemanipulation architecture presented.

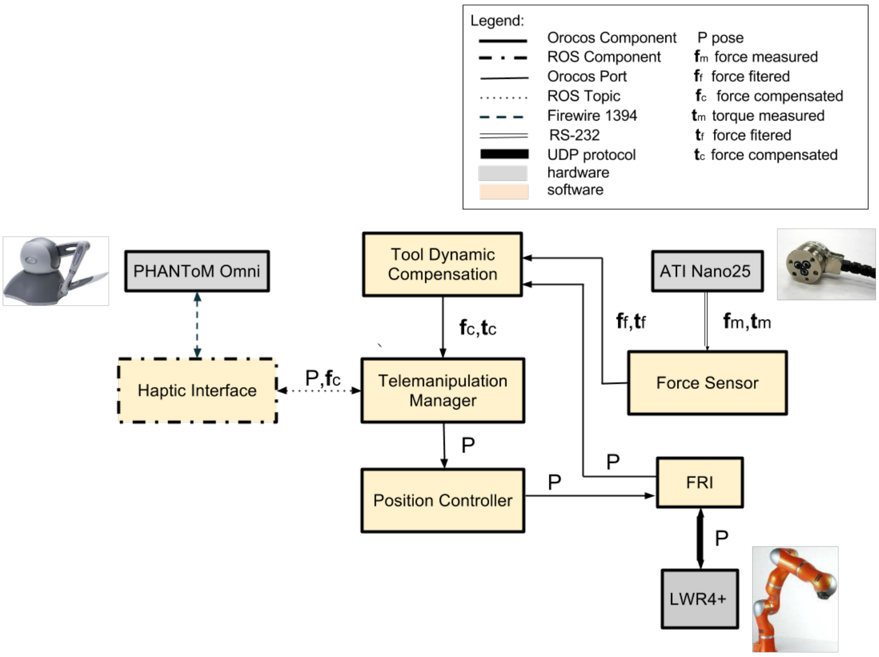


Figure 3.2: Telemanipulation system architecture: blocks represent the software components that manage the telemanipulation system. The software interfaces for exchanging data with the hardwares (grey blocks) are *Haptic Interface*, *Force Sensor* and *FRI*. The other software components *Tool Dynamic Compensation*, *Telemanipulation Manager*, *Position Controller* process the data necessary in a bilateral telemanipulation system. Each line is associated with the framework or communication protocol used.

3.2 Telemanipulation System: hardware components

The hardware architecture, as mentioned above, consists of a slave robot arm placed beside an operating table and a workstation on which the surgeon works . A force/torque sensor is placed on the end-effector (ee).

The work-station is composed by a master robot arm and a screen, on which are displayed the scene captured by the camera on the surgical field. The overall system is shown in Figure 3.1.

In the next paragraphs each device is explained in details.

3.2.1 Master Haptic Interface - Phantom omni (Geomagic)

The Master device of the bilateral telemanipulation system is a serial robot, the PHANToM OMNI haptic interface (HI) (Geomagic, Morrisville, NC 27560, USA), shown in Figure 3.3.

PHANToM Omni is a six-degrees-of-freedom (DoFs) positional sensing robotic arm, with a range of motion like hand movement pivoting at wrist. The wrist movement are reproduced by the gimbal angles show in Figure 3.4. The most important feature is that it is a motorized device that applies force feedback on the user's hand, allowing them to

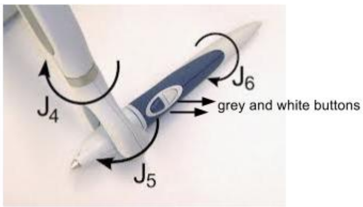


Figure 3.4: PHANToM Omni: gimbal angles and buttons

feel virtual objects and producing true-to-life touch sensations. It is actuate only on 3 DoFs and it is able to reproduce up to a end-effector (ee) nominal force of 3.3 N [36].

In Table 3.1 are shown complete technical specifications [37].

Hence, the role of PHANToM Omni as master device in the bilateral telemanipulation system is to manage the translation and rotation movements of the slave arm at distance, and even more to reproduce the interaction force between the surgical tool attached at the slave arm and the environment in which is performing the surgery.

The two buttons placed on the stylus of the haptic interface can be used to allow a safe enablement of the telemanipulation task (Figure 3.4).

Figure 3.3: Phantom Omni Haptic Interface - Sensable

PHANToM OMNI Technical Specification	
Workspace	> 160 W x 120 H x 70 D mm
Range of motion	Hand movement pivoting at wrist
Nominal position resolution	˜0.055 mm
Maximum exertable force at nominal (orthogonal arms) position	3.3 N
Stiffness	x-axis > 7.3 lb/in (1.26 N/mm) y-axis > 13.4 lb/in (2.31 N/mm) z-axis > 5.9 lb/in (1.02 N/mm)
Force feedback (3 Degrees of Freedom)	x, y, z
Interface	IEEE 1394 FireWire ®

Table 3.1: PHANToM Omni Technical Specifications

3.2.2 Slave Arm Robot - LightWeightRobot 4+ (KUKA)

The Slave device of the bilateral telemanipulation system serial arm Robot, the Light Weight Robot 4+ (LWR4+), (KUKA, Augsburg, Germany), shown in Figure 3.5.

The LWR4+ is a 7 DoFs robotic arm,which model is supposed to be based on the human model of an arm, aiming at a weight-to-payload ratio of 1:1.

The main characteristics are [4]:

- rate payload of 7kg, with a mass of 15kg.
- kinematic redundancy, i.e. seven degrees of freedom which made possible, for example, an elbow motion while maintaining the pose of the end-effector;
- joint-integrated power and signal processing electronics including torque measurements in all joint by means of torque sensors;
- actively controllable, programmable compliance at joint and cartesian level. Joint and cartesian stiffness and damping parameters can be adjusted between zero and the maximum value corresponding to position control;

	Stiffness [Nm/rad]		Damping [N*s/rad]	
Joint	Min	Max	Min	Max
1:7	0.01	5000	0.1	1.0

- The velocity by which the KUKA Robot Controller manages the point-to-point trajectory has a trapezoidal profile. As shown in Figure 3.22, when the robot has to get to a point in cartesian space it reaches the costant velocity set with an acceleration and stops with a deceleration equal and opposite to the acceleration. Depending on the cartesian velocity set, the acceleration/deceleration changes.

The role of the LWR4+ is to physically replace the arm of the surgeon and tracks the movements executed at distance by the HI. Due to the presence of this device on the operative field, the LWR4+ has the advantages of a kinematic redundancy and slender structure, so that is not too cumbersome and doesn't obstacle the small field of view of the surgeon during the surgical gesture or other kind of movement in the operating room.

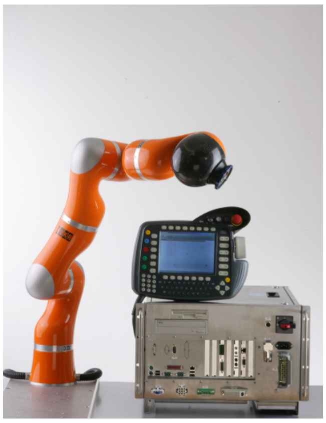


Figure 3.5: KUKA Lightweight Robot 4+ (lwr4+) with KUKA Robot Controller (KRC) and KUKA Teach Pendant [4]

3.2.3 Force Sensor - Nano25 (ATI Industrial Automatation)

The force/torque sensor used to feel the interaction forces/torques of the slave arm with the environment is a Nano25, from ATI Industrial Automatation.

The ATI Multi-Axis Force/Torque Sensor system measures all six components of force and torque. The system consists of a transducer, shielded high-flex cable, and intelligent data acquisition system, F/T controller. The communication with the F/T controller is managed through a serial port. The small dimension (Figure 3.6) (Table 3.3), allows to attach it directly at the end-effector of the slave arm.

Technical specifications are shown in Table 3.2 [38].

The role of the force/torque sensor in the bilateral telemanipulation system is to measure the interaction force of the surgical tool with the environment of the operative field, in order to allow the surgeon to feel that forces by the HI.

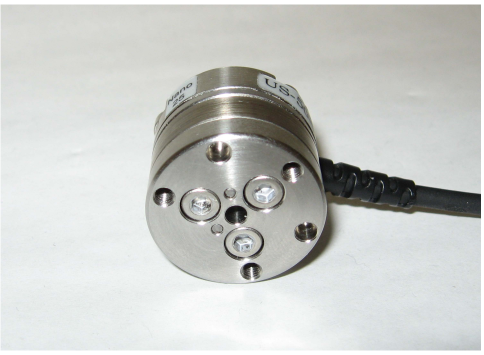


Figure 3.6: Force/Torque Sensor Nano25 - ATI Industrial Automatation

Technical Specification	Fx,Fy	Fz	Tx,Ty	Tz
sensing ranges	125 N	500 N	3 Nm	3 Nm
resolution	1/12 N	1/12 N	1/660 Nm	1/660 Nm
accuracy	1.25 N	5 N	0.03 Nm	0.03 Nm
Single-Axis Overload	±2300 N	±7300	±43 Nm	±63 Nm
Resonant Frequency	3600 Hz	3800 Hz	3800 Hz	3600 Hz

Table 3.2: ATI Force/Torque Sensor Nano25 - Technical Specifications

Dimensions	
Weight	0.0634 kg
Diameter	25 mm
Height	22 mm

Table 3.3: ATI Force/Torque Sensor Nano25 - Dimensions

3.3 Teleoperation System: software components

The software component described in the next paragraphs can be divided in:

- interface software components (Section 3.3.3), which allow the communication with the device using specific Application Programming Interface (API);

- processing components (Section 3.3.4), that elaborate the data read by the interface software components.

The overall applications are developed using ROS and Orocos framework, described in paragraph 3.3.1 and 3.3.2.

The software architecture is shown in Figure 3.7.

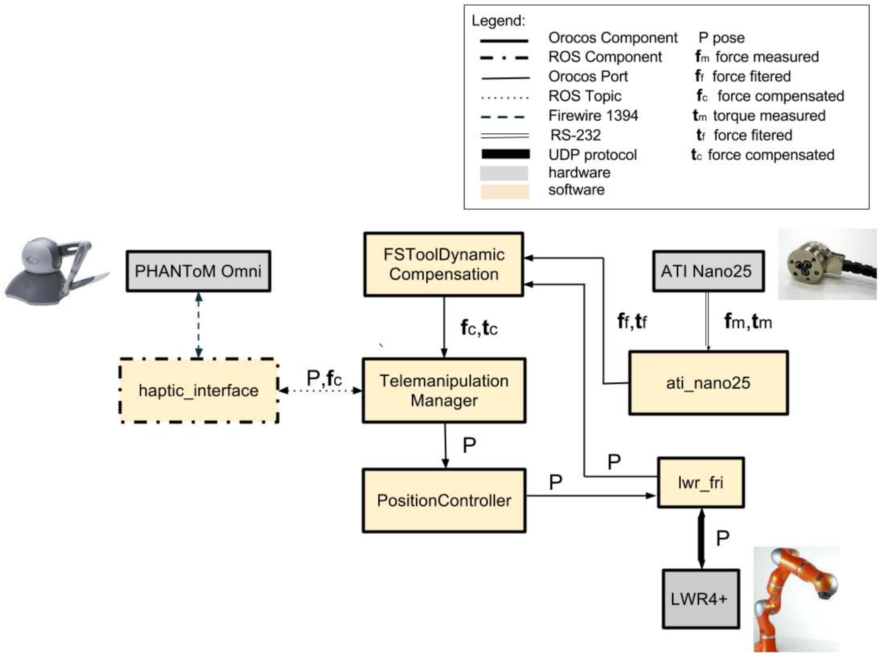


Figure 3.7: Architecture of software components in telemanipulation system. In details communication protocol and information exchange between components.

3.3.1 Robot Operating System - ROS

ROS is a distributed framework of processes that enables executables to be individually designed and loosely coupled at runtime. These processes are grouped into *Packages*.

Packages are the main unit for organizing software in ROS. It contains ROS runtime processes (*nodes*), a ROS-dependent library, datasets, configuration files, or anything else that is usually organized together.

ROS *nodes* are the core of the software, processes that perform computation. ROS is designed to be modular: a robot control system will usually comprise many nodes.

As shown in Figure 3.8, data flow between different *nodes* is done using *messages* which are being transported over *topics*, by a TCP/IP-based transport protocol, called TCPROS protocol. *Messages* are routed via a transport system with publish/subscribe semantics. A *node* sends out a *message* by publishing it to a given *topic*. The *topic* is a name that is used to identify the content of the *message*. A *node* that is intrested in a certain kind of data will subscribe to the appropriate *topic*. There may be multiple concurrent *publishers* and *subscribers nodes* for a single topic, and a single *node* may publish and/or subscribe to multiple *topics*. *publisher* and *subscribers* are aware of each others' existence and its connection is created by contacting the ROS *master*, this software component is responsible for the interconnection of all ROS components. Once a *publisher* and *subscriber node* are connected, the communication between both components is fully peer-to-peer without interaction with the ROS *master*. [**?**][39]

ROS is not a realtime framework, though it is possible to integrate ROS with realtime code developed using Orocos framework.

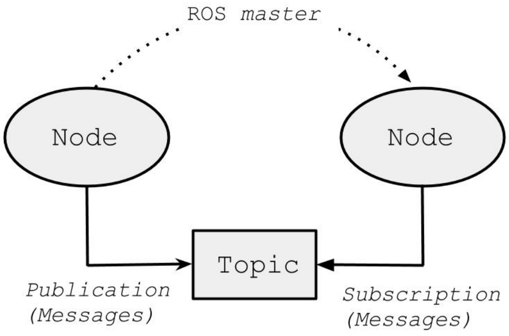


Figure 3.8: ROS architecture: nodes, the software core, communicate each other through messages, publishing or subscribing at a topic of interest. The communication is enabled at first time by the ROS master.

3.3.2 Open RObot COntrol Software - Orocos

Orocos is an application generating a set of components from which the skeleton of any robot control application can be constructed. As its name implies, its focus is hard realtime control of robotic components.

The Orocos Toolchain Library is referred also to a "middleware" because it sits between the application and the Operating System. It take care of the real-time communication and execution of software *components*.

Figure 3.9 shows the main structure of an Orocos *component*.

An Orocos *component* is a basic unit of functionality which executes one or more C++ programs in a single thread and delivers a service that other program can use. A component has *ports* to send or receive a stream of data. The algorithm inside the component writes *Output ports* to publish data to other components, while *Input ports* allow an algorithm to receive from other component. Each component defines its data exchange ports and connection transmit data from one port to another. In Orocos there is a clear decoupling between the component that writes or reads the data on a *port* and how the data is transported between different components through *messages*.

Moreover, *Operations* and *Properties* define respectively which function a component offers through the Service Interface and are intended to configure and tune a variable component with certain values through the Configuration Interface.

The interconnection with ROS packages is simple: using the ROS transport, any Orocos component's port can be connected to a ROS topic, without interfering with the real time behavior of Orocos component [40][5].

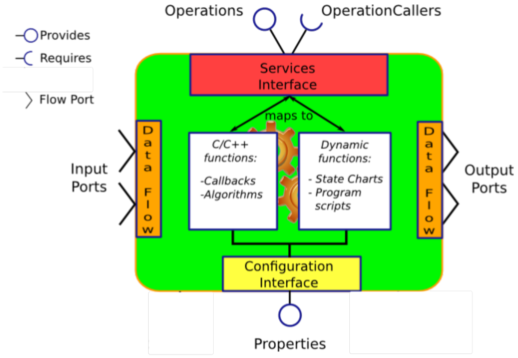


Figure 3.9: Orocos component architecture: Input - Output ports allow dataflow between components, Operations and Properties define component function and configuration [5]

3.3.3 Software interface component

3.3.3.1 "KUKA Fast Research Interface" (FRI) Component

The KUKA Fast Research Interface (FRI) is developed, from KUKA Roboter GmbH, to give direct low-level real-time access to the KUKA robot controller (KRC) at high rates of up to 1KHz.

It is an Ethernet-based connection between the KUKA KRC industrial robot controller and an external control unit, such as a standard desktop computer. The communication is based on simply binary user datagram protocol (UDP) messages. The FRI communication rate is user-configurable between 1 ms and 100 ms. If a rate lower than 1 ms is chosen, the external commands are interpolated in order to realize a smooth behaviour at the lower control level. Three control modes are supported: Joint Position Control, Joint Impedance Control, Cartesian Impedance Control. [41]

The communication with the LWR4+ through the FRI is done by a specially designed Orocos/RTT Component, that can in itself be connected to any other Orocos/ROS component in the control system (using any available data flow middleware).

The code can be found in the lwr_fri package which is available in the K.U.Leuven ROS package repository at: http://git.mech.kuleuven.be/robotics/kul-ros-pkg.git.[40]

3.3.3.2 "haptic_interface" Component

The *haptic_interface* component is a ROS interface software component which allows the communication with the PHANToM Omni haptic interface device.

The API (OpenHaptics Toolkit) provided by Geomagic includes the Haptic Device API (HDAPI). These libraries allow low-level access to the haptic device and enables haptics programmers to directly render force and get data.

In the *haptic_interface* ROS node the HDAPI is used to initialize the device, initialize the scheduler, start the scheduler, perform haptic commands using the scheduler, then exit when done.

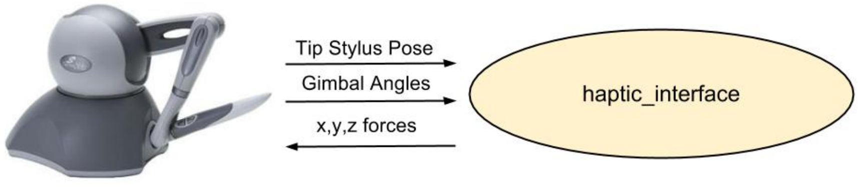


Figure 3.10: *haptic_interface* ROS package: data exchange with the device. The ROS *node* sets x,y,x forces and read stylus pose and gimbal angles.

"Device initialization" includes everything necessary to communicate with the device. This generally involves creating a handle to the device, enabling forces, and calibrating the device.

The scheduler callbacks allow the programmer to enter commands that will be performed within the servo loop thread. The scheduler manages a high frequency, high priority thread for sending forces and retrieving state information from the device. Typically, force updates need to be sent at 1000 Hz frequency in order to create compelling and stable force feedback [42]. HDAPI can be used to modify the rate of the servo loop. The servo loop refers to the tight control loop used to calculate forces to send to the haptic device.

Calls to the device typically involve managing state, setting or getting parameters, and sending forces. The necessary information include querying buttons, position, velocity, gimbal angles (Figure 3.4), and endpoint transform matrices.

Figure 3.10 shows the information exchange by the ROS node *haptic_interface* with the device. It reads the stylus pose, the gimbal angles of the last three joints used to control the slave arm, and set force read by the force sensor and compensated on the haptic interface.

3.3.3.3 "ATIForceSensor" Component

The ATIForceSensor Component is an Orocos component that reads force/torque from the ATI Nano25 F/T Sensor Controller.

The primary function of the F/T Sensor Controller is to convert strain gauge data to Cartesian force/torque. Hence, it is able to output resolved force/torque values in ASCII format through the RS-232 serial port, providing data in readable characters.

Therefore, the Orocos component that manages the communication with the force sensor consists of:

- setting the serial port parameters: 8-bit transmission with no parity and one stop bit, baud rate 38400 bit/s.
- reading data from the serial port and converting it in force/torque values.

  In ASCII mode one data record consists of 57 bytes if the output consists of six force/torque components; the first byte is the error flag followed by a comma and the force/torque data in the order of Fx, Fy, Fz, Tx, Ty, Tz. Error code at the beginning of the record is an indicators of strain gauge saturation or transducer damage.

Force and torque values are reported in counts. Counts are integer values set so one count is near the ideal resolution of the F/T system. The use of integers, instead of real numbers, produces faster output. The Nano 25 sensor has 12 counts per Kg related to force and 660 counts per Kg related to torque. For example, a force output of 128 counts would indicate a load of 10.60 Kg. 10.60 Kg = 128 count ÷ 12 counts/Kg.

- computing the sensor bias.

  Biasing eliminates the effects of gravity (tool weight) or other forces acting on the end-effector. When a sensor bias is performed, the Controller reads the forces and torques currently acting on the sensor and uses these readings as a reference for future readings. Future readings will have this reference subtracted from them before they are transmitted.

3.3.4 Processing component

3.3.4.1 Telemanipulation Manager

The Telemanipulation manager is an Orocos Component with the task of managing the control and transformations involved in the bilateral telemanipulation system. The telemanipulation control scheme used is a PF. The master controls the slave sending position and orientation information, while the force sensor on the slave gives forces to the master.

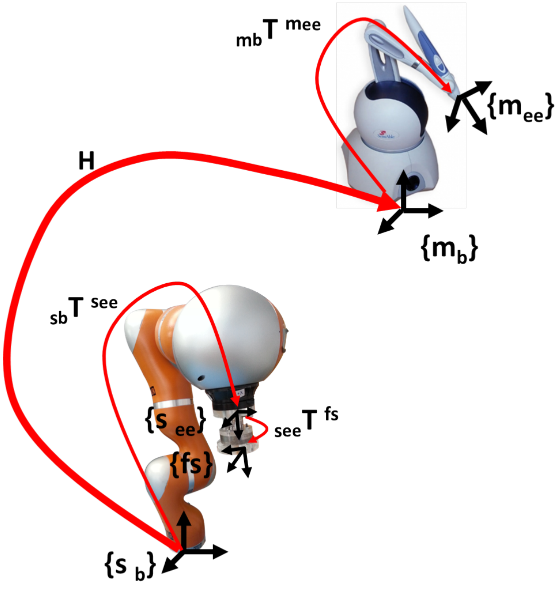


Figure 3.11: Transformations involved in the telemanipulation system

The Fig.3.11 shows the transformations between the master reference frame, in the base and in the end-effector $\{m_b\}\{m_{ee}\}$, and the slave reference frame, in the base and in the end-effector $\{s_b\}\{s_{ee}\}$. H is the transformation between $\{s_b\}$ and $\{m_b\}$ and has to be an identity if we want that the slave mimics the movement of the master.

When the surgeon is sitting at the work-station, he holds the stylus of the master robot. The stylus has two buttons used to enable the telemanipulation modes.

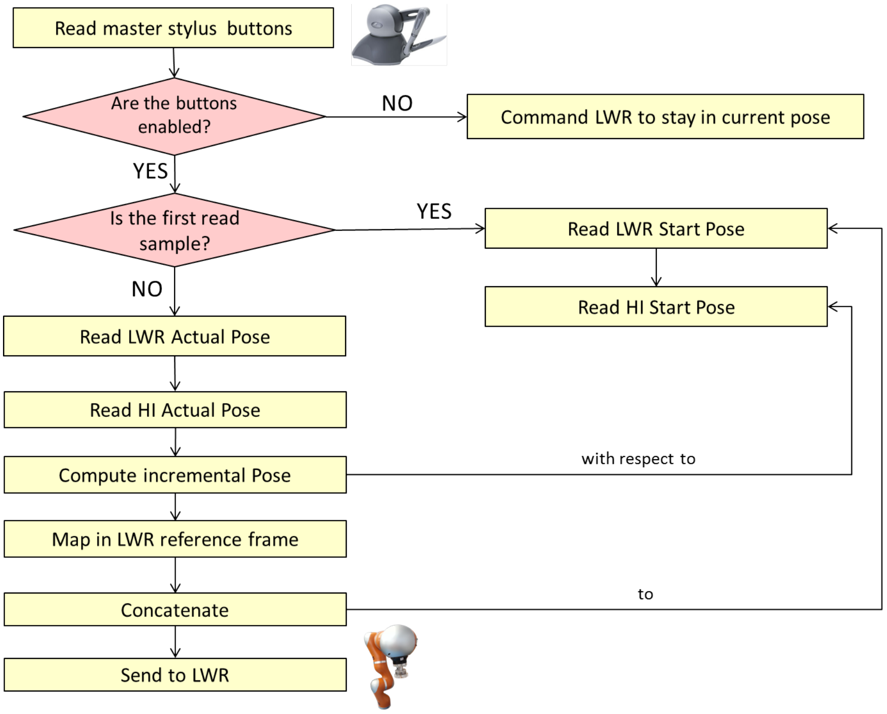


Figure 3.12: Flowchart of the Telemanipulation Manager component

As described in the flowchart in Figure 3.12, whenever the two buttons are pressed, the start pose of the slave and master robot are saved. Thus, all the following pose of the master are related to its start pose and concatenated at the start pose of the slave arm. The slave, starting from its start pose ${}_{\{s_b\}}T^{\{s_{ee}\}}_{start}$, move itself as much as the incremental movements of the master with respect its start pose ${}_{\{m_b\}}T^{\{m_{ee}\}}_{start}$, .

The movements executed by the remote surgical tool (attached at slave arm) have to be equal to the movements of the haptic interface stylus in terms of rotation and translation. Thus, for example, if we move the stylus towards the positive x axis, we want to see the surgical tool moving towards the same direction. H has to become an identity.

The master pose increments ${}_{m_b}d\mathbf{T}^{m_{ee}}$ with respect to its start pose is computed as:

$$ {}^{\{m_b\}}d\mathbf{T} = {}_{m_b}\mathbf{T}^{m_{ee}^{-1}}_{start} \cdot {}_{m_b}\mathbf{T}^{m_{ee}} \tag{3.1}$$

The incremental movements in the master reference frame $\{m_b\}$ are mapped in the slave reference frame $\{s_b\}$:

$$ {}^{\{s_b\}}d\mathbf{T} = \mathbf{H} \cdot {}^{\{m_b\}}\, d\mathbf{T} \tag{3.2}$$

The slave pose desired is computed as:

$$ {}_{s_b}\mathbf{T}^{s_{ee}}_{des} = {}_{s_b}\mathbf{T}^{s_{ee}}_{start} \cdot {}_{m_b}d\mathbf{T}^{m_{ee}} \tag{3.3}$$

Different features are enabled by pressing simultaneously or individually the grey and white buttons on the stylus of the master haptic interface:

grey-button The telemanipulation is managed only in translation.

white-button The telemanipulation is managed only in rotation.

grey&white-button The telemanipulation is managed in roto-translation.

Also changing Property of the Orocos Component enables different features:

- force feedback. The force **f** in $\{fs\}$ is reported in $\{s_b\}$ and then mapped in $\{m_b\}$.

$$^{\{s_b\}}\mathbf{f} =_{\{s_b\}} \mathbf{T}^{\{fs\}\{fs\}}\mathbf{f} \tag{3.4}$$

$$^{\{m_b\}}\mathbf{f} = \mathbf{H}\cdot^{\{s_b\}}\mathbf{f} \tag{3.5}$$

- scaling of rotation, translation and force

Finally, the desired cartesian pose of the slave arm is sent to an Orocos Component which manages the Joint Position Control, communicating with the robot through the lwr_fri component, described above.

The processing frequency is 500 Hz.

3.4 Dynamic compensation of the surgical instrument

The surgical tool is attached at the distal part of the FS, which measures the interaction force and torque with the environment in which the LWR is performing a task, in order to allow the force feedback . As mentioned above the FS can not be placed on the tip of the surgical tool to prevent electronics from entering the patient and for reasons of size and sterility, hence it is located between the slave ee and the instrument as shown in Figure 3.13.

However, the tool exerts forces and torques on the force sensor due do its weight and its dynamic behaviour during a motion. The main problem is that anyway the sensor measures force and torque due to:

- the weight of the surgical instrument both in static and dynamic conditions;

- the contact force and torque with the environment;

If these forces are reproduced on the stylus of the haptic interface held by the surgeon, he doesn't perceive only the contact force of the tool with the environment, but in addition he feels also the weight of the surgical tool.

The next paragraphs propose an algorithm able to dynamically compensate the weight of a generic tool attached at the distal part of the FS. After the dynamic compensation, the forces reproduced by the haptic interface are only the contact forces of the tool with the environment.

3.4.1 Force/Torque filtering

The force and torque measured by the force sensor are affected by noise, and need to be filtered. The main noise is induced by the engine vibration of the robot and by the friction of the rotational joint.

Thus, the first step is characterized by a frequency analysis of the force and torque signals. Both static and dynamic acquisitions of the force/torque signals is computed with the force/torque sensor attached on the robot end-effector and the robot engine on. The

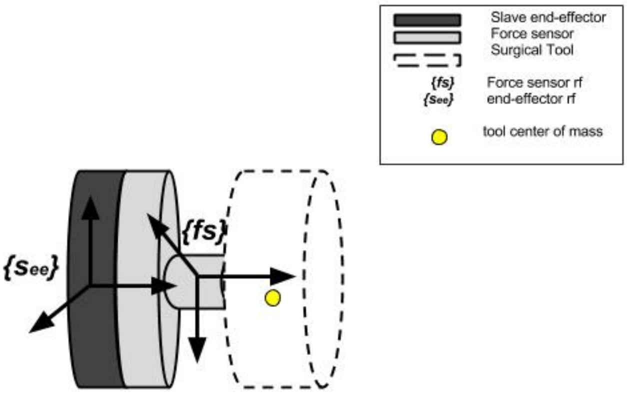
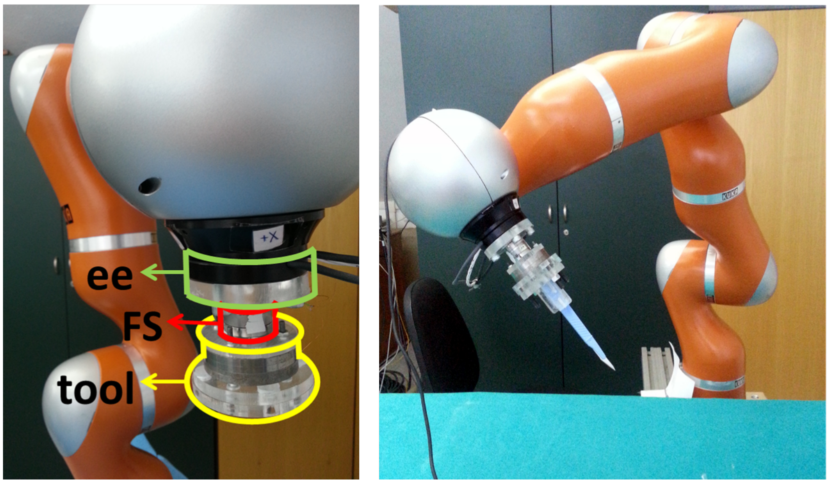


Figure 3.13: Force/ Torque Sensor at the slave end-effector with attached a generic tool



(a) (b)

Figure 3.14: (a) generic tool of known geometry attached to the force sensor (b) surgical tool (scalpel) attached to the force sensor

Notch Filter Parameters	
signal frequency	85 Hz
notch frequency	18Hz
bandwith	20 Hz
structure	Direct Form II
order	2
group delay	1 sample at 9 Hz
a1	-0.3251
a2	0.3662
b0	0.6831
b1	-0.3251
b2	0.6831

Table 3.4: Notch filter parameters

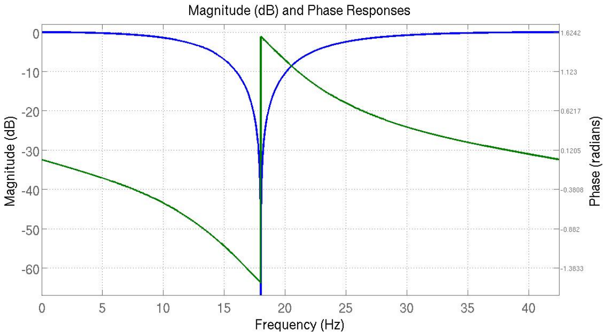


Figure 3.15: Magnitude and Phase Responses of notch filter

main noise frequency is identified via the frequency analysis, computing the fast fourier transform (fft) in MATLAB (Version 7.12.0 - R2011a).

A online single notch IIR filter (Eq. 3.6, 3.7) is implemented, in order to eliminate the major frequency components of noise. The parameters of the notch filters, shown in Table 3.4 and Figure 3.153.16, are defined using the Open Filter Design and Analysis Tool (fdatool) Toolbox of Matlab.

$$v(n) = x(n) - a_1 v(n-1) - a_2 v(n-2) \tag{3.6}$$

$$y(n) = b_0 v(n) + b_1 v(n-1) + b_2 v(n-2) \tag{3.7}$$

A post-filtering frequency analysis is also done to verify the effectiveness of the filtering.

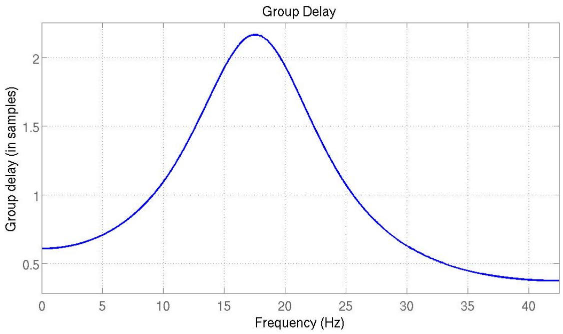


Figure 3.16: Group delay of notch filter

3.4.2 Force/Torque estimation

The online dynamic tool compensation needs to estimate the force and torque exerted by surgical tool attached at the distal part of the force/torque sensor during any type of movements. These force and torque to be estimated are due to to the dynamically distribution of the tool's weight on the three axis of the force sensor reference frame $\{fs\}$.

The compensation, of force and torque $\mathbf{f}_{\mathrm{c}}$ $\mathbf{t}_{\mathrm{c}}$, consists in removing the force/torque estimated $\mathbf{f}_{\mathrm{e}}$ $\mathbf{t}_{\mathrm{e}}$ from the force/torque filtered $\mathbf{f}_{\mathrm{f}}\mathbf{t}_{\mathrm{f}}$, Equations 3.8 and 3.9.

$$^{\{\mathrm{fs}\}}\mathbf{f}_c = {}^{\{\mathrm{fs}\}}\mathbf{f}_f - {}^{\{\mathrm{fs}\}}\mathbf{f}_e \tag{3.8}$$

$$^{\{\mathrm{fs}\}}\mathbf{t}_c = {}^{\{\mathrm{fs}\}}\mathbf{t}_f - {}^{\{\mathrm{fs}\}}\mathbf{t}_e \tag{3.9}$$

The equations used to estimate the force and torque generated by a generic surgical instrument are derived from the basic laws of dynamics. Based on the Newton-Euler approach, the motion of a rigid body due to external forces and torques is described by two vector equations, i.e. Equations 3.10 and 3.11, that are linear with regard to the unknown parameters, the linear and angular acceleration vector $\mathbf{a}$ and α, the angular velocity vector ω, the gravity vector $\mathbf{g}$, and the inertia matrix $\mathbf{I}$.[43]

The forces $\mathbf{f}_{\mathrm{e}}$and $\mathbf{t}_{\mathrm{e}}$ are estimated in $\{fs\}$ in order to be directly removed from the $\mathbf{f}_{\mathrm{f}}$, $\mathbf{t}_{\mathrm{f}}$measured in $\{fs\}$.

$$\mathbf{f}_e = m\mathbf{a} - m\mathbf{g} + \alpha \times m\mathbf{c} + \omega \times (\omega \times m\mathbf{c}) \tag{3.10}$$

$$\mathbf{t}_e = \mathbf{I}\alpha + \omega \times (\mathbf{I}\alpha) + m\mathbf{c} \times \mathbf{a} - m\mathbf{c} \times \mathbf{g} \tag{3.11}$$

where

$$\mathbf{I} = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yx} \\ I_{xz} & I_{yz} & I_{zz} \end{pmatrix} \tag{3.12}$$

The equation 3.13 relates the force/torque $\mathbf{f}_s$,τ_s, the variables $\mathbf{a}_s$, α_s, ω_s and $\mathbf{g}_s$ to the complete set of inertial parameters contained in φ_s.

$$\begin{pmatrix} \mathbf{f}_e \\ \tau_e \end{pmatrix} = \mathbf{V}\varphi_s \tag{3.13}$$

$$\varphi_s = [m,\, m\mathbf{c}_x,\, m\mathbf{c}_y, m\mathbf{c}_z,\, I_{xx},\, I_{xy},\, I_{xz},\, I_{yy},\, I_{yz},\, I_{zz}]^T \tag{3.14}$$

Equations 3.10and 3.11 are used to compose the matrix $\mathbf{V}$.

The parameter vector φ_s contains the complete set of inertial parameter of tool.

In the next paragraphs are described the methods for the estimation of *mass*, *center of mass* and *inertial matrix* of the tool.

3.4.2.1 Cartesian Velocity and Acceleration computation

We defined a method in order to compute on-line $\mathbf{a}$, α and ω in $\{fs\}$.

Starting from the informations on the joint angle position $\mathbf{q}$ it is computed the angular joint velocity $\dot{\mathbf{q}}$ with the Best Fit-First Order Adaptive Windowing Filter (Best fit FOAW) [44], a finite impulse response (FIR) velocity estimation technique. The explanation of the best fit FOAW is presented in the Appendix.

Using $\dot{\mathbf{q}}$, and applying Equation 3.15 is possible to compute the cartesian linear and angular velocity in $\{s_{ee}\}$.

$${}^{\{s_{ee}\}}\mathbf{vel} = \mathbf{J}\dot{\mathbf{q}} \tag{3.15}$$

In order to estimate the force/torque in $\{fs\}$ and to be comparable with the force/torque measured by the force sensor the cartesian velocity and acceleration is needed in $\{fs\}$.

Thus, the kinematics chain of the slave robot at which is attached the force sensor is modified adding a rigid link at the end-effector, as if the force sensor was the new end-effector of the slave arm.

With the new robot chain, applying the Equation 3.15 is computed the cartesian velocity ${}^{\{fs\}}\mathbf{vel}$ in the force sensor reference frame $\{fs\}$.

The cartesian acceleration is computed using the Best fit FOAW.

3.4.2.2 Identification of mass, center of mass and inertial parameters

The first of the inertial parameters 3.14 estimated are the mass m and the center of mass *cdm* of the surgical tool (Figure 3.13).

The m and *cdm* are computed in static condition, using respectively the equation 3.10 and 3.17; neglecting the components involved in dynamic become:

$${}^{\{\mathrm{fs}\}}\mathbf{f} = m^{\{fs\}}\mathbf{g} \tag{3.16}$$

$${}^{\{\mathrm{fs}\}}\mathbf{t} = \mathrm{m} \cdot \mathbf{cdm} \times^{\{\mathrm{fs}\}} \mathbf{g} \tag{3.17}$$

where ${}^{\{\mathrm{fs}\}}\mathbf{f}$ and${}^{\{\mathrm{fs}\}}\mathbf{t}$ are the force measured in $\{fs\}$ by the force sensor and $\mathbf{g}$ is the gravity vector in $\{fs\}$.

As $\mathbf{f}$ and $\mathbf{t}$ are measured in $\{fs\}$, also the contribution of the gravity has to be calculated in$\{fs\}$.

The transformation ${}_{\{s_b\}}\mathbf{T}^{\{s_{ee}\}}$ between $\{s_b\}$ and $\{fs\}$ has to be computed:

$${}_{\{s_b\}}\mathbf{T}^{\{fs\}} =_{\{s_b\}} \mathbf{T}^{\{s_{ee}\}} \cdot_{\{s_{ee}\}} \mathbf{T}^{\{fs\}} \tag{3.18}$$

${}_{\{s_b\}}T^{\{s_{ee}\}}$ is obtained applying the forward kinematics starting from the joint position $\mathbf{q}$, while ${}_{\{s_{ee}\}}\mathbf{T}^{\{fs\}}$ is determined.

Replacing the 3.19 into 3.16 and 3.17, the estimation of the m and *cdm* of mass is obtained solving least square problem in MATLAB 7.12.0 (R2011a).

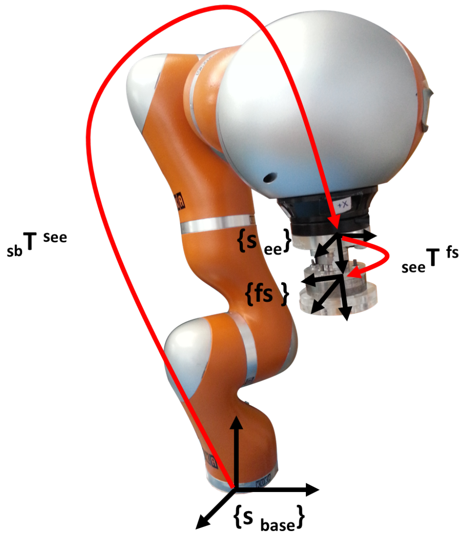


Figure 3.17: Representation of LWR reference frames and force sensor reference frame and transformations.

The data of force/torque and joint position are recorded with the robot put in 30 different static poses.

$$ {}^{\{fs\}}\mathbf{g} = {}_{\{s_b\}}\mathbf{T}^{\{fs\}^{-1}} . {}^{\{s_b\}}\,\mathbf{g} \tag{3.19} $$

The inertial matrix **I** is computed starting from the assumption that the surgical tool has a known geometry, similar to a composition of solid cylinders . Thus, the formulas used are:

$$ I_{x,y} = \frac{1}{12} m(3r^2 + h^2) \tag{3.20} $$

$$ I_z = \frac{mr^2}{2} \tag{3.21} $$

where r is the radius of cylinder and h is the height.

The computation of the correct transformation ${}_{\{s_{ee}\}}\mathbf{T}^{\{fs\}}$ between $\{s_{ee}\}$ and $\{fs\}$, shown in Figure 3.17, is described in the next paragraph.

3.4.2.3 Identification of the transformation between end-effector and force sensor reference frame

The force sensor is mounted on the end-effector of the slave so that $\{s_{ee}\}$ and $\{fs\}$ frame are coaxial along the z axis. Hence, the transformation ${}_{\{s_{ee}\}}\mathbf{T}^{\{fs\}}$ is characterized only by a rotation around the z axis.

It is known that

$$ {}^{\{fs\}}\mathbf{g} = ({}_{\{s_b\}}\mathbf{T}^{\{s_{ee}\}} \cdot {}_{\{s_{ee}\}}\,\mathbf{T}^{\{fs\}})^{-1} \cdot {}^{\{\mathrm{s_b}\}}\mathbf{g} \tag{3.22} $$

where ${}_{\{s_b\}}T^{\{s_{ee}\}}$ is the transformation between the base $\{s_{base}\}$ and the end-effector of the slave $\{s_{ee}\}$ and ${}_{\{s_{ee}\}}T^{\{fs\}}$ is the transformation between $\{s_{ee}\}$ and $\{fs\}$.

The gravity ${}^{\{fs\}}\mathbf{g}$ is measured with the force sensor, the gravity ${}^{\{s_b\}}\mathbf{g}$ is known , the transformation ${}_{\{s_{base}\}}\mathbf{T}^{\{s_{ee}\}}$ can be computed applying the forward kinematics starting from the joint position measurement.

The only variable unknown is the ${}_{\{s_{ee}\}}\mathbf{T}^{\{fs\}}$.

The algorithm used to compute the unknown rotation around the z axis wants to find the angle θ that minimizes the root mean square error (rmse) of the error (err_g) between ${}^{\{fs\}}\widehat{\mathbf{g}}$ estimated and the ${}^{\{fs\}}\mathbf{g}$ measured.

If

$$err_g = {}^{\{fs\}}\mathbf{g}_{x_i} - {}^{\{fs\}}\,\widehat{\mathbf{g}}_{x_i} \tag{3.23}$$

and

$${}^{\{fs\}}\widehat{\mathbf{g}} = ({}_{\{s_b\}}\mathbf{T}^{\{s_{ee}\}} \cdot {}_{\{s_{ee}\}}\,\mathbf{T}^{\{fs\}}(\theta))^{-1} \cdot {}^{\{s_b\}}\,\mathbf{g} \tag{3.24}$$

the angle $\theta_{optimum}$ is the angle tha minimizes the rmse of err_g.

This algorithm is performed off-line, analysing the data measured in MATLAB 7.12.0 (R2011a). The function used for the minimization is `lsqnonlin` , the algorithm used is Levengerg-Marquardt, and the solution is constrained between $\pm\pi$. The angle initialization is chosen measuring with a goniometer the approximate angle of rotation around the z axis.

The data of force/torque and joint position are recorded with the robot put in 30 different static poses.

Hence, the flow chart of the estimation algorithm, shown in 3.18, starts from the acquisition of the joint position, computes the joint velocity applying the best fit FOAW, finds the cartesian velocity considering the forward kinematics and the Jacobian and computes the cartesian acceleration with the best fit FOAW on-line. The remaining inertial parameters, mass, center of mass e inertial matrix of the tool are computed off-line. Once the overall unknown variable are identified the dynamic compensation of the tool is possible.

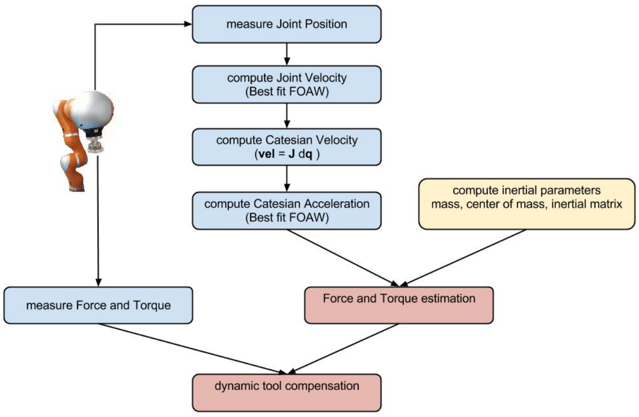

measure Joint Position
compute Joint Velocity
(Best fit FOAW)
compute Catesian Velocity
(vel = J dq)
compute Catesian Acceleration
(Best fit FOAW)
compute inertial parameters
mass, center of mass, inertial matrix
measure Force and Torque
Force and Torque estimation
dynamic tool compensation

Figure 3.18: Flowchart of the dynamic tool compensation algorithm

3.5 Experimental trials

The need to incorporate force feedback capabilities into surgical robotically assisted procedures provides an excellent opportunity to improve the quality of the surgery. The system proposed aims to improve the force feedback, trying to reproduce to the surgeon side exactly and transparently the interaction force.

In the next paragraphs the analysis on the communication delay involved in the bilateral telemanipulation system and on the force/torque estimation error involved in the dynamic tool compensation.

3.5.1 Telemanipulation delay validation

The analysis executed on the telemanipulation system evaluates the communication delay between the movement controlled by the HI and the movement tracks by the LWR.

The communication delay consists of:

- δ_r = time taken to read data from the slave and master devices on the local network;
- δ_w = time taken to send data to the slave devices on the local network;
- δ_c = computational time of the Telemanipulation Manager algorithm, in which position and orientation of the master devices are mapped in $\{s_b\}$ and the force are mapped in $\{m_b\}$, as described in Section 3.3.4.1 ;

Hence, the total delay δ_t, shown in Figure 3.19, is:

$$\delta_t = \delta_r + \delta_c + \delta_r \tag{3.25}$$

3.5.1.1 Protocol

The test consists in harmonic movements of the stylus of the HI, and resulting movements of the slave arm (LWR), performed 12 times, 30 seconds each, with a sampling frequency of 100 Hz. All the data are recorded under ROS/Orocos architecture, ensuring time synchronization.

The movements of the master and slave is recorded by an external system, the Optotrack Certus localization system (NDI, Canada), an optical tracking system that track 2 dynamic reference frames (DRFs) rigidly placed respectively on HI and LWR end-effectors (with an accuracy of 0.15mm). The set up is shown in Figure 3.20. In order to track exactly the tip of the HI stylus (ee) and the tip of the surgical instruments on the slave arm, a pivoting is computed in order to estimate the transformation between the DRFs and the tip of the stylus and the tool. In that way the data recorded by the localization system refers to the tip of the devices and not to the center of the DRFs.

δ_c is measured inside the C++ code, using the function getTicks() from <rtt/os/TimeService.hpp> orocos library.

For each trials, the communication delay δ_t between the master and slave movements is estimated computing the cross-correlation of the signals in MATLAB 7.12.0 (R2011a).

3.5.1.2 Analysis

The media, standard deviation, mediana and 25°-75° percentile of the delays δ_t and δ_c of the 12 tests are computed.

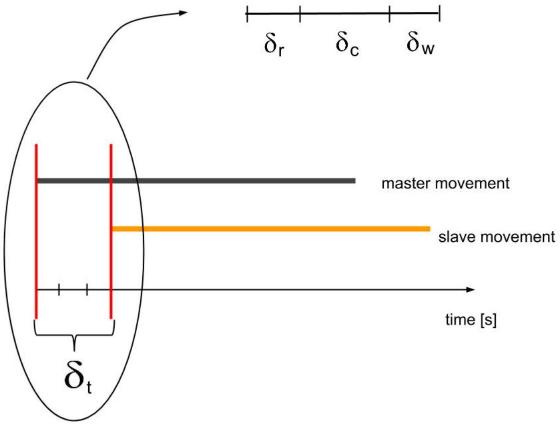


Figure 3.19: Delay between the movement controlled by the master and the movement tracks by the slave. It is divided in: time taken to read and send data on the local network, δ_r and δ_w, and the computational time of the Telemanipulation Manager algorithm, δ_c.

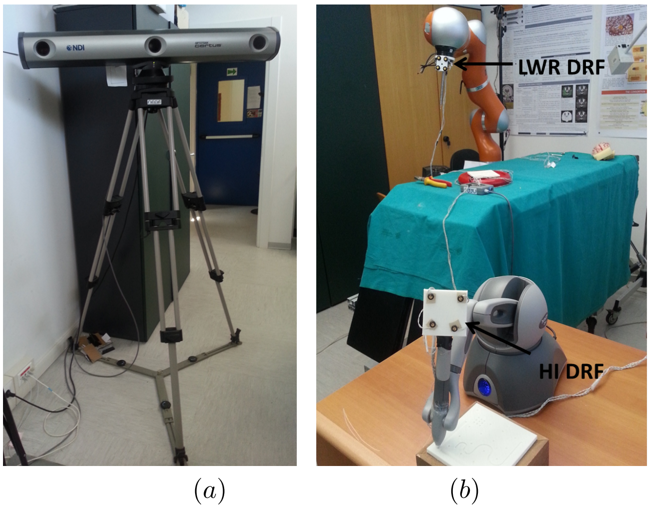


Figure 3.20: Setup of the communication delay estimation: (*a*) Optrotrack Certus localization system (*b*) PHANToM Omni and LWR4+ with DRFs tracked by the localization system.

3.5.2 Dynamic Tool Compensation Validation

The *FSToolDynamicCompensation* component has the aim of online estimate the force/torque exerted by the surgical tool on the force sensor and compensate it, in order to have available only the contact force between the surgical tool and the environment.

The analysis of the tool compensation is focused on the evaluation of the error ($\mathbf{err}_f$ and $\mathbf{err}_t$) between force and torque measured $\mathbf{f}_\mathrm{m}$, $\mathbf{t}_\mathrm{m}$ and estimated $\mathbf{f}_\mathrm{e}$, $\mathbf{t}_\mathrm{e}$ in different conditions.

3.5.2.1 Protocol

The tests are performed with the slave device (LWR), the force sensor (FS), and a tool of known geometry. The tool is placed on the distal part of the force sensor, which is attached to the end effector of the robot, as shown in Figure 3.14-(a).

The trials are executed in:

Static condition:

LWR with the FS and the tool is put in 40 different poses and is fixed in the reached pose for 20 seconds.

Dynamic condition*:*

LWR with the FS and the tool performs:
- *linear motion* along the x, y, z FS axes $\{fs\}$;
- *rotation* around the x, y, z FS axes $\{fs\}$;

In Figure 3.21 are shown examples of linear and angular movements in $\{fs\}$.

Each movements is executed at 10- 30- 50- 70- 100% of the maximum robot cartesian velocity. 250mm/s is the maximum cartesian velocity performing linear movement and 2rad/s is the maximum cartesian velocity performing angular movement.

In our trials for each of the 5 velocities there is a correspondent costant acceleration and deceleration, by reference to the trapezoidal profile of velocity used by the KUKA Controller to reach a point (see Section 3.2.2); hence, for the overall linear and angular trials, at 5 velocities, there are 5 costant linear and angular accelerations and decelerations:

maximum cartesian velocity [% of max vel]	10	30	50	75	100
linear deceleration/acceleration [m/s^2]	±0.023	±0.21	±0.57	±1.26	±2.23
angular deceleration/acceleration [rad/s^2]	±0.012	±0.16	±0.44	±0.98	±1.74

Table 3.5: Values of cartesian linear and angular acceleration and deceleration used by the KUKA Controller in the trapezoidal velocity profile to reach a point, associated to the cartesian velocity.

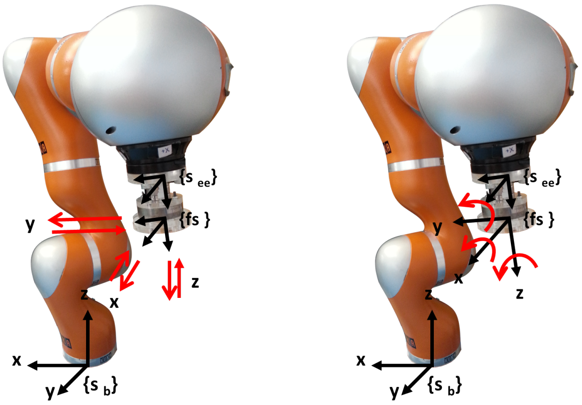
Datails are shown in Tables 3.6 and 3.7.

Dynamic condition				
vel (% of max vel)	linear motion in cartesian axis			# trials
	x [mm]	y[mm]	z[mm]	
10	±300	±300	±200	10
30	±300	±300	±200	10
50	±300	±300	±200	10
75	±300	±300	±200	10
100	±300	±300	±200	10

Table 3.6: Tool Compensation Validation Protocol - linear motion

Dynamic condition				
vel (% of max vel)	angular motion in cartesian axis			# trials
	x [deg]	y [deg]	z [deg]	
10	±90	±100	±60	12
30	±90	±100	±60	12
50	±90	±100	±60	12
75	±90	±100	±60	12
100	±90	±100	±60	12

Table 3.7: Tool Compensation Validation Protocol - angular motion



(*a*) (*b*)

Figure 3.21: (*a*) example of linear movements in FS reference frame $\{fs\}$ (*b*) example of angular movements in FS reference frame $\{fs\}$.

3.5.2.2 Measures

The force and torque measured, $\mathbf{f}_{\mathrm{m}}$ and $\mathbf{t}_{\mathrm{m}}$, are measured at 85 KHz and the force and torque estimated, $\mathbf{f}_{\mathrm{e}}$ and $\mathbf{t}_{\mathrm{e}}$, are computed at 500 Hz using the algorithm described in Section 3.4.

For each acquisition in static and dynamic condition, the analysis is based on the evaluation of the force/torque estimation error ($\mathbf{err}_f$, $\mathbf{err}_t$), computed as:

$$\mathbf{err}_{f_i} = \mathbf{f}_{\mathrm{m_i}} - \mathbf{f}_{e_i} \tag{3.26}$$

$$\mathbf{err}_{t_i} = \mathbf{t}_{\mathrm{m_i}} - \mathbf{t}_{e_i} \tag{3.27}$$

In static condition, $\mathbf{err}_f$ and $\mathbf{err}_t$are computed on the entire signals recorded with FS in a different orientations.

In dynamic condition, we want to evaluate the estimation error $\mathbf{err}_f$ and $\mathbf{err}_t$ when the tool is subject to acceleration. We compute the cartesian velocity of the slave in $\{fs\}$ during the dynamic motion and we find the part of velocity signals in which the acceleration is costant (see Figure 3.22). The estimation error is evaluated on the corresponding forces ($f1 : f2$), i.e. corresponding to a motion with constant acceleration ($\frac{a2-a1}{dT}$) and deceleration, as show in Figure 3.22.

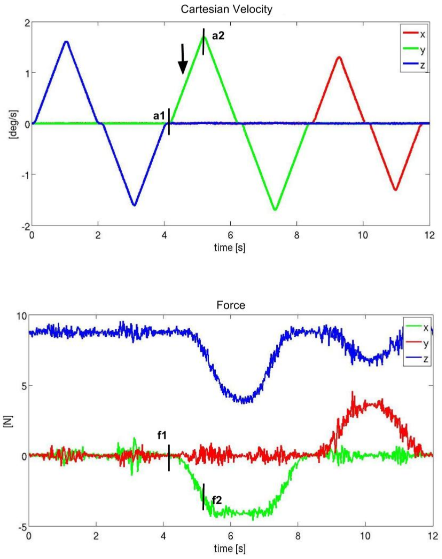


Figure 3.22: In the upper plot is shown the trapezoidal profile of the velocity in FS cartesian axes $\{fs\}$, the range $a1 : a2$ indicates the costant acceleration. In the lower plot is shown the corresponding values of force in FS cartesian axes $\{fs\}$, and the range $f1 : f2$ indicates the force exerted by the tool on the force sensor during a movement at costant acceleration.

3.5.2.3 Statistical Analysis

For each trial, root mean square errors ($\mathbf{rmse}_{\mathrm{f}}$, $\mathbf{rmse}_t$), medians ($\mathbf{median}_f$, $\mathbf{median}_{\mathrm{t}}$) and 25° - 75° percentiles (p_{25}, p_{75}) of the $\mathbf{err}_{\mathrm{f}}$ and $\mathbf{err}_{\mathrm{t}}$ are computed:

$$\mathbf{rmse}_{\mathrm{f}} = \sqrt{\frac{\sum_{i=1}^{N} \mathbf{err}_{\mathrm{f_i}}^{\mathbf{2}}}{N}} \tag{3.28}$$

$$\mathbf{rmse}_{\mathrm{t}} = \sqrt{\frac{\sum_{i=1}^{N} \mathbf{err}_{\mathrm{t_i}}^{\mathbf{2}}}{N}} \tag{3.29}$$

$$\mathbf{rmse}_{\mathrm{f_{x,y,z}}} = \sqrt{\frac{\sum_{i=1}^{N} \mathbf{err}_{\mathrm{f_{x_i}}}^{\mathbf{2}} + \mathbf{err}_{\mathrm{f_{y_i}}}^{\mathbf{2}} + \mathbf{err}_{\mathrm{f_{z_i}}}^{\mathbf{2}}}{N}} \tag{3.30}$$

$$\mathbf{rmse}_{\mathrm{t_{x,y,z}}} = \sqrt{\frac{\sum_{i=1}^{N} \mathbf{err}_{\mathrm{t_{x_i}}}^{\mathbf{2}} + \mathbf{err}_{\mathrm{t_{y_i}}}^{\mathbf{2}} + \mathbf{err}_{\mathrm{t_{z_i}}}^{\mathbf{2}}}{N}} \tag{3.31}$$

where i refers to the number of samples.

$\mathbf{rmse}_{\mathrm{f}}$ and $\mathbf{rmse}_{\mathrm{t}}$ are evaluated under:
Static condition
A Pearson's linear correlation coefficient between $\mathbf{rmse}_{\mathrm{f}}$ and $\mathbf{rmse}_{\mathrm{t}}$ on x, y, z axes and FS orientation is computed in MATLAB (Version 7.12.0 - R2011a). The correlation function computes p-values for Pearson's correlation.

Dynamic condition

A Pearson's linear correlation coefficient between the $\mathbf{rmse}_\mathrm{f}$ and $\mathbf{rmse}_\mathrm{t}$ and the costant acceleration/deceleration of the performed task is computed in order to understand if the error committed depends on the acceleration of the motion. The correlation function computes p-values for Pearson's correlation.

Chapter 4

Results

This chapter shows the results of the analysis computed on:

- communication delay involved in the telemanipulation system;

- the estimation of force $\mathbf{f}_e$ and torque $\mathbf{t}_e$ used for the dynamic tool compensation;

4.1 Telemanipulation delay results

In Figure 4.1 are shown the position of HI and LWR during execution of armonic movements with the stylus of the master using the proposed telemanipulation system.

The red and green line are respectively the master and slave position.

The range of velocity in which are performed the master movements is from -0.12 m/s to 0.14 m/s.

In Table 4.1 are presented the mean value, standard deviation, median and 25°-75° percentile of the delay, expressed in sample [# sample] and in second [s] between the master and slave position signals.

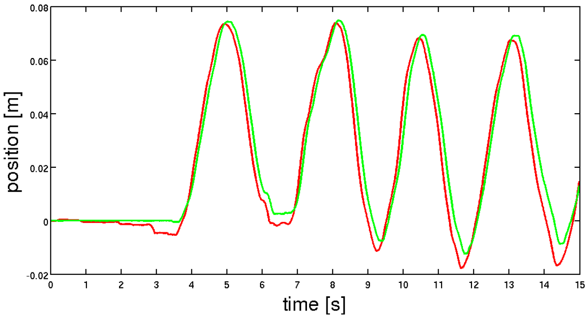


Figure 4.1: Example of master and slave position following. The red line shows the master position, while the green line shows the slave position.

	25° percentile	median	75° percentile	media	std
$\delta_r + \delta_w$ [# sample]	0	5	7.5	4.5	±3.8964
$\delta_r + \delta_w$ [second]	<10 ms	50 ms	75 ms	45 ms	±39 ms
δ_c [second]	7.0172e-06	7.0881e-06	7.3560e-06	-	-

Table 4.1: Statistical analysis on communication delay: median, 25° and 75° percentile, media and standard deviation

4.2 Force and torque filtering results

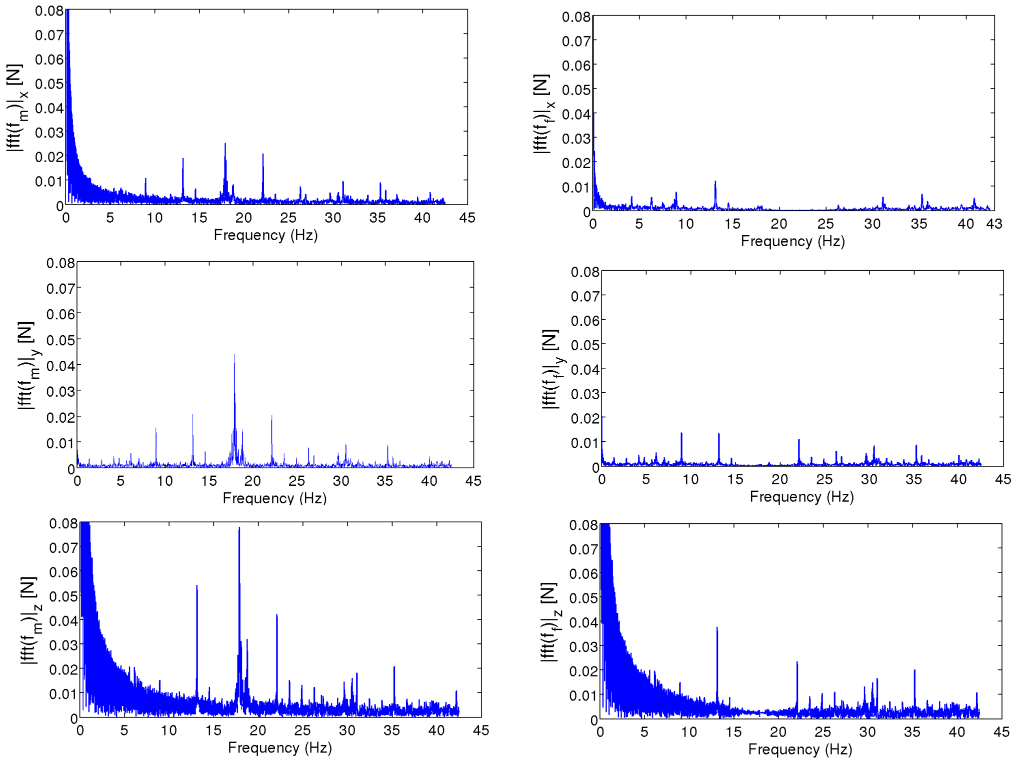
The first step in the dynamic tool compensation is to filter forces and torques measured by the FS.

This section show the frequency analysis of $\mathbf{f}_m$, $\mathbf{t}_m$ and of $\mathbf{f}_f$, $\mathbf{t}_f$ described in paragraph 3.4.1, to identify the frequency component of noise and to verify the effectiveness of the notch filter implemented. The analysis are implemented in static and dynamic condition.

At last the signal frequency component is evaluated computing the fft of the force estimated $\mathbf{f}_e$, assuming that this force is less affected by noise with respect to the force measured and is easier to identify the movement frequency component.

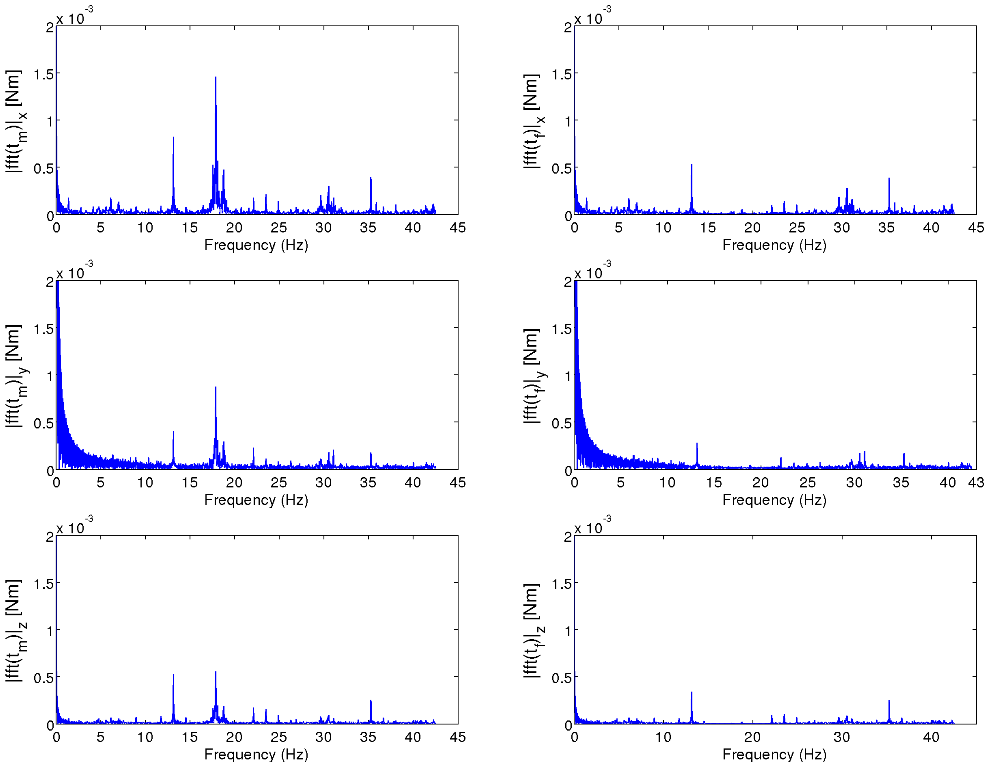
4.2.1 Frequency analysis of $\mathbf{f}_m, \mathbf{f}_f$ and in $\mathbf{t}_m, \mathbf{t}_f$ in static condition

Figures 4.2 and 4.3 show the frequency analysis of force/torque measured $\mathbf{f}_m$, $\mathbf{t}_m$ (in the left side (a)) and of force/torque filtered $\mathbf{f}_f$, $\mathbf{t}_f$ (in the right side (b)). The signals analyzed are recorded with the FS attached at the ee of the LWR in static condition, with the engine of the robot on. In (a) the fft of $\mathbf{f}_m$, $\mathbf{t}_m$ shows the picks that represent the frequency content of the noise in the three axes x, y, z from the upper to the lower plot. In (b) the same signals $\mathbf{f}_f$, $\mathbf{t}_f$ after applying the notch filtering is presented.



(*a*) (*b*)

Figure 4.2: (*a*) Frequency Analysis of force before filtering: identification of the noise frequency components in static condition with robot's engine on in the three axes x, y, z from the upper to the lower plot - (*b*) Frequency Analysis of force after filtering: identification of the noise frequency components in static condition with robot's engine on in the three axes x, y, z from the upper to the lower plot



(a) (b)

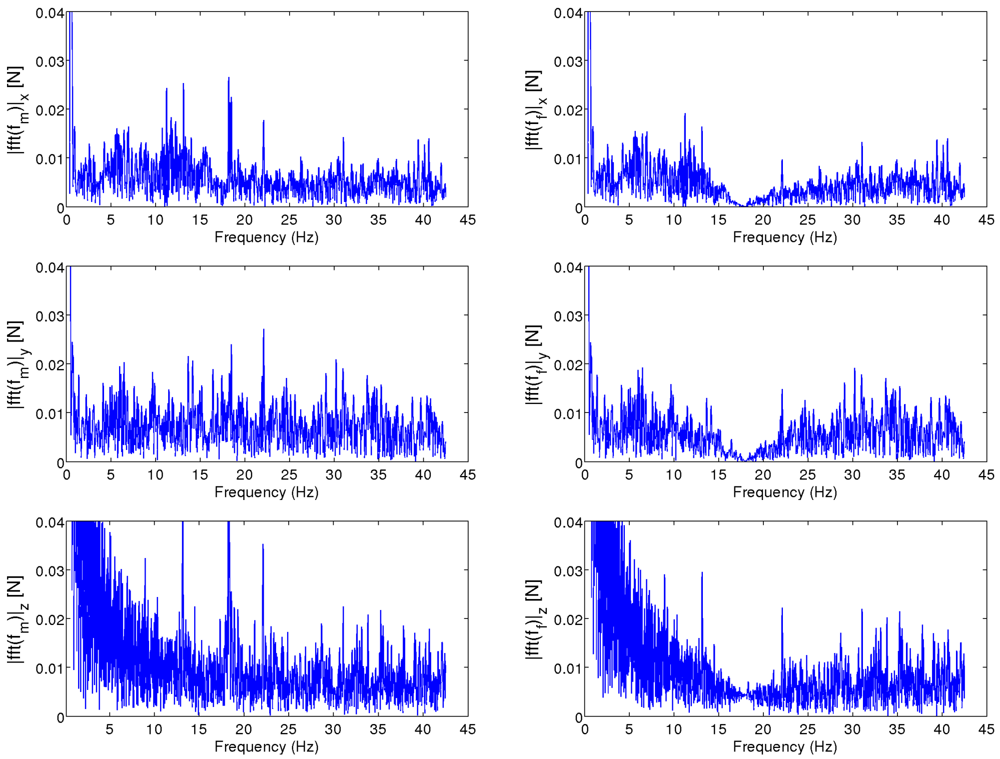
Figure 4.3: (a) Frequency Analysis of torque before filtering: identification of the noise frequency components in static condition with robot's engine on in the three axes x, y, z from the upper to the lower plot - (b) Frequency Analysis of torque after filtering: identification of the noise frequency components in static condition with robot's engine on in the three axes x, y, z from the upper to the lower plot

4.2.2 Frequency analysis of $\mathbf{f}_m, \mathbf{f}_f$ and in $\mathbf{t}_m, \mathbf{t}_f$ in dynamic condition

Figures 4.4 and 4.5 show frequency analysis of $\mathbf{f}_m$, $\mathbf{t}_m$ and of $\mathbf{f}_f$, $\mathbf{t}_f$. The signal analyzed is recorded with the FS attached to the ee of the LWR in dynamic condition, executing angular movement around $\{fs\}$ cartesian axes.

In (a) the fft of $\mathbf{f}_m$, $\mathbf{t}_m$ shows the picks that are the frequency content of the noise in all the three axes x, y, z from the upper to the lower plot. In (b) the same signals $\mathbf{f}_f$, $\mathbf{t}_f$ after applying the notch filtering is presented.

Figure 4.6 shows an example of the fft of $\mathbf{f}_e$. The picks represent the frequency components of a linear movement along the z axis in $\{fs\}$. In Table 4.2 are presented the value of the signal frequency component for linear and angular movements executed at minimum and maximum cartesian velocity.



(a) (b)

Figure 4.4: (a) Frequency Analysis of force before filtering: identification of the noise frequency components in dynamic condition in x, y, z axes from the upper to the lower plot - (b) Frequency Analysis of force after filtering: identification of the noise frequency components in dynamic condition in x, y, z axes from the upper to the lower plot

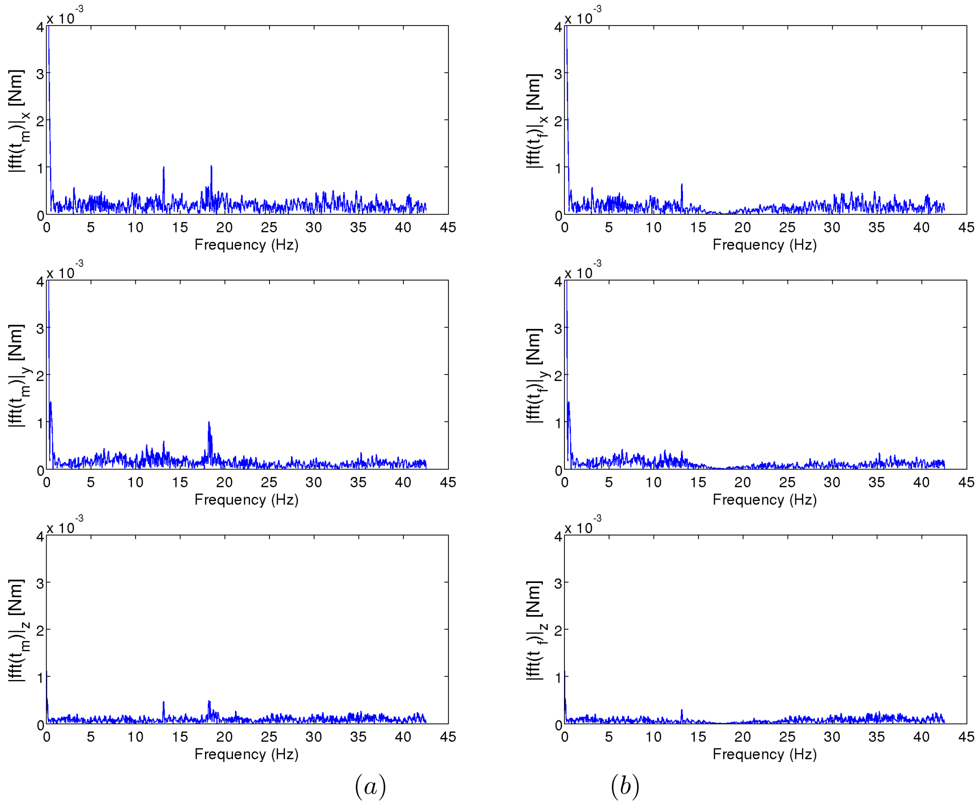


Figure 4.5: (a) Frequency Analysis of torque before filtering: identification of the noise frequency components in dynamic condition in x, y, z axes from the upper to the lower plot - (b) Frequency Analysis of torque after filtering: identification of the noise frequency components in dynamic condition in x, y, z axes from the upper to the lower plot

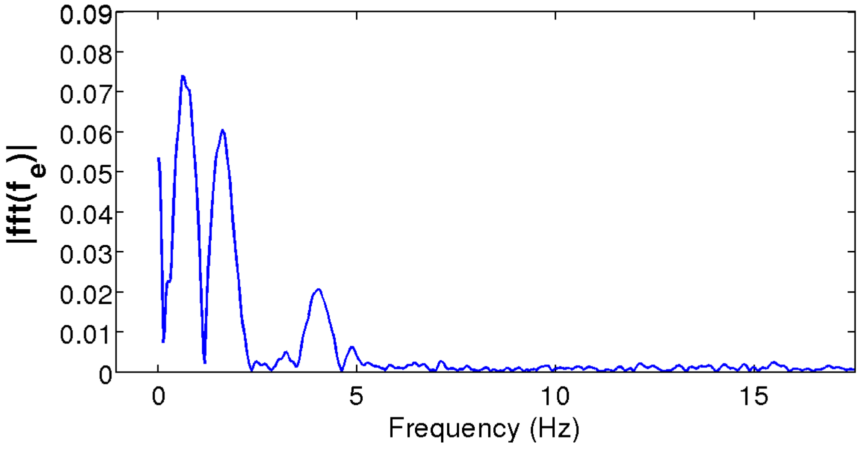


Figure 4.6: fft of $\mathbf{f}_e$ to identify the signal frequency component

	Frequency Component of signal			
	min vel (10%)	max vel(100%)		
linear movement	0.055	0.641	1.617	4.059
angular movement	0.061	0.762	1.617	4.028

Table 4.2: Values of signal frequency component for linear and angular movements at minimum and maximum cartesian velocity

The frequency analysis of both force and torque measured in static and dynamic condition ((*a*) in Figure 4.2, 4.3, 4.4 and 4.5) on the three cartesian axes of $\{fs\}$ shows three main peacks of noise at 13 Hz, 18 Hz, 22 hz. Hence, the notch frequency is set to 18 Hz with a bandwidth of 20 Hz (Figure 3.15, 3.16), considering that the main frequency component of the signal range from 0.055Hz to 4.059 Hz depending on the kind and the velocity of the movements, see Figure 4.6 and Table 4.2. The frequency analysis of the same signals after filtering ((*b*) in Figure 4.2, 4.3, 4.4 and 4.5) shows that the noise component at 18 Hz is completely eliminated, while the other two main noise components at 13Hz and 22Hz are partially attenuated.

4.2.3 Comparison between $\mathbf{f}_m$ and $\mathbf{f}_f$

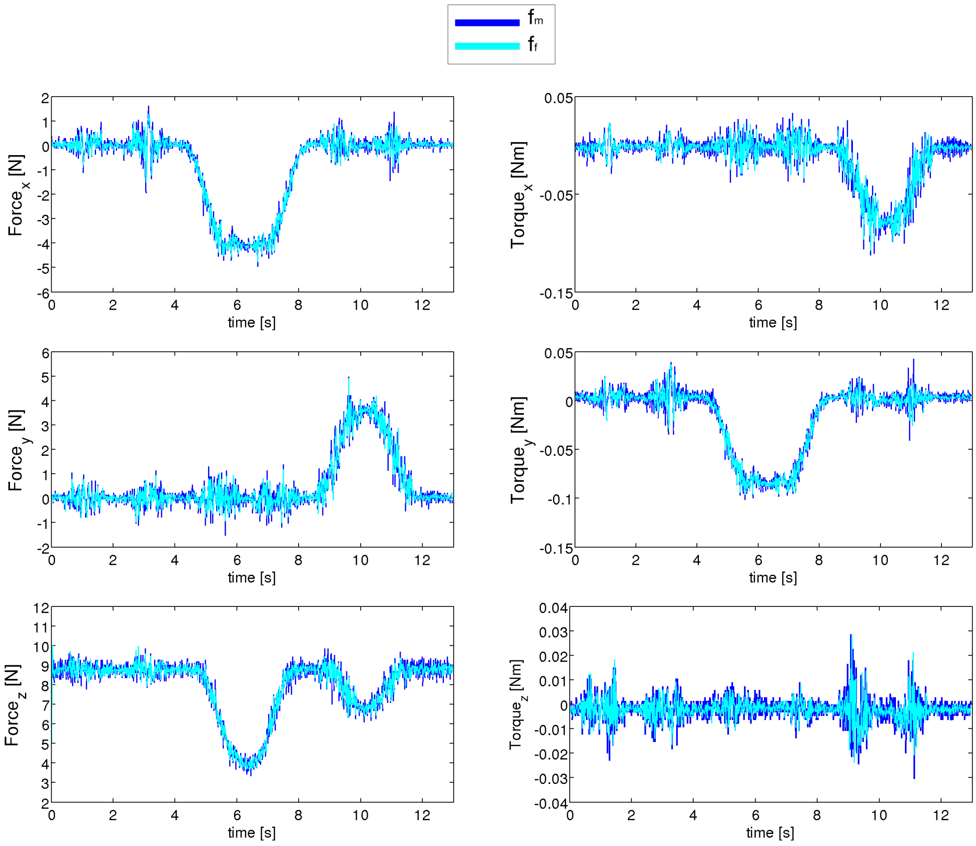
Figure 4.7shows the force and torque signals, recorded during angular movements around $\{fs\}$ cartesian axes, before and after the notch filtering.

The comparison between $\mathbf{f}_m$ and $\mathbf{f}_f$ in $\{fs\}$ cartesian axes is presented in column (a), while the comparison between $\mathbf{t}_m$ and $\mathbf{t}_f$ in $\{fs\}$ cartesian axes is presented in column (b)

The signal to noise ratio *snr* between $\mathbf{f}_m$ and $\mathbf{f}_f$ is computed as the ratio between the variance of $\mathbf{f}_m$ and the variance of $\mathbf{f}_f$, when the force sensor is biased, with mean equal to zero:

$$snr = var(f_m)/var(f_f) \tag{4.1}$$

The *snr* is equal to 0.2900 N.



(*a*) (*b*)

Figure 4.7: (a) comparison between $\mathbf{f}_m$ and $\mathbf{f}_f$ in x, y, z axes from the upper to the lower plot- (b) comparison between $\mathbf{t}_m$ and $\mathbf{t}_f$ in x, y, z axes from the upper to the lower plot

4.3 Force and torque estimation results

4.3.1 Identification of mass and center of mass and ee-FS transformation

In Table 4.3 are presented the values of *mass* and *cdm*, estimated in Section 3.4.2.2, of the tool shown in Figure 3.14.

Figure 4.8 shown the $\{fs\}$ reference frame estimated in Section 3.4.2.3 with respect to the $\{s_b\}$.

	real value			estimated value		
mass [Kg]	0.273Kg			0.1629Kg		
center of mass [m] no tool	-	-	-	-0.0006	0.0	0.0155
$_{\{s_{ee}\}}T^{\{fs\}}(\theta)$ [rad]	-	-	-	0.1192 (6.8279 deg)		

Table 4.3: Values of mass, center of mass and $_{\{s_{ee}\}}T^{\{fs\}}(\theta)$

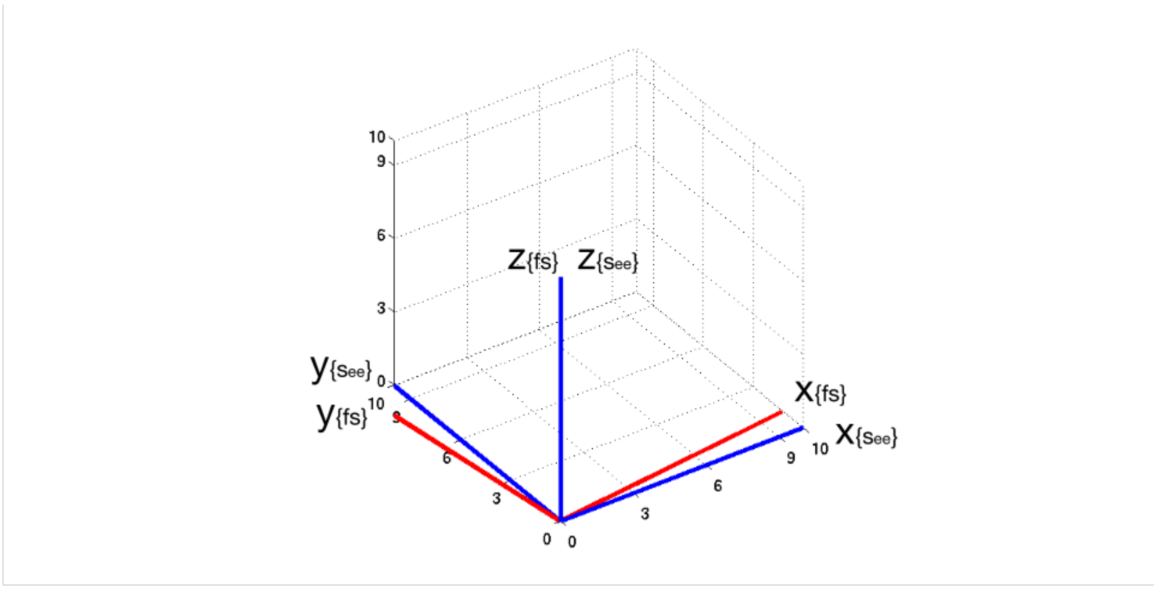


Figure 4.8: $\{s_{ee}\}$ reference frame in blue with respect to$\{fs\}$ in red

4.3.2 Force and torque estimation

This section presents $\mathbf{f}_e$ and $\mathbf{t}_e$ with respect to $\mathbf{f}_f$ and $\mathbf{t}_f$. The estimation is computed with the algorithm described in Section 3.4.2.

Figure 4.9 shows in column (a) and column (b) respectively $\mathbf{f}_f$, $\mathbf{f}_e$ and $\mathbf{t}_f$, $\mathbf{t}_e$ in $\{fs\}$ x, y, z axes from the upper to the lower plot.

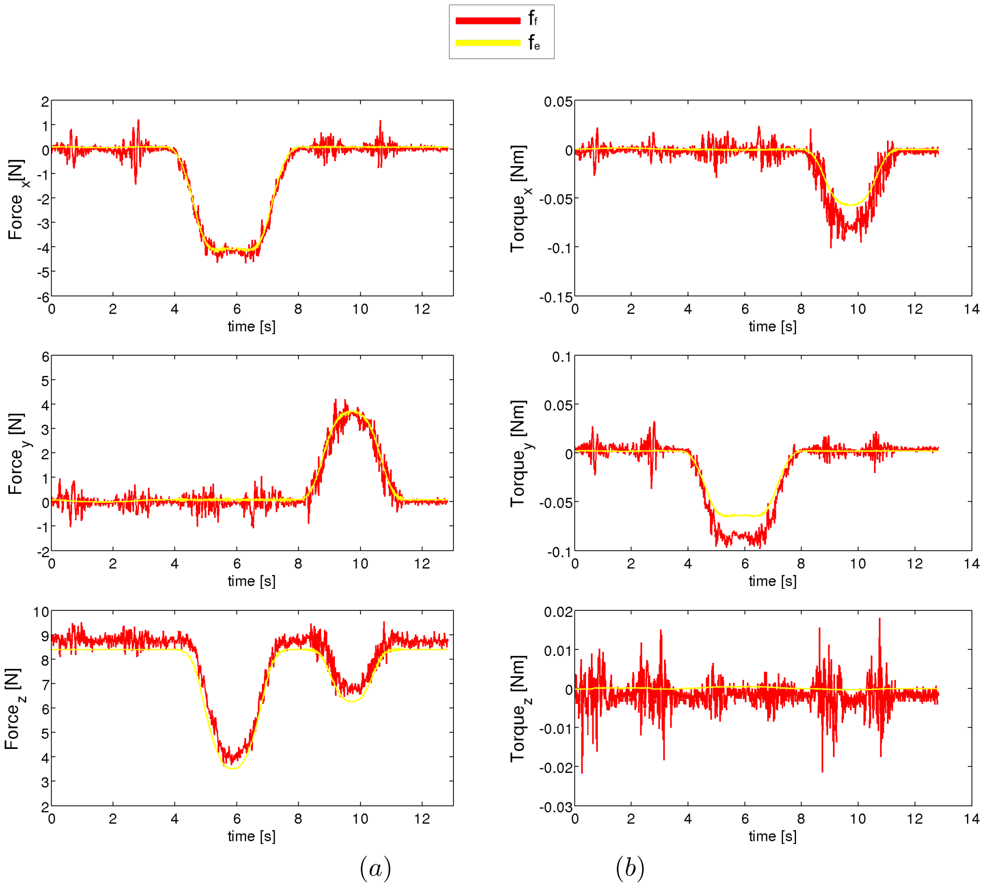


Figure 4.9: (a) comparison between $\mathbf{f}_m$ and $\mathbf{f}_e$ in x, y, z axes from the upper to the lower plot - (b) comparison between $\mathbf{t}_m$ and $\mathbf{t}_e$ - (example of angular movement around $\{fs\}$ cartesian axes x, y, z from the upper to the lower plot)

4.3.2.1 Static condition

Figure 4.10 and 4.11 show the population of the $\mathbf{rmse}_f$ and $\mathbf{rmse}_t$ committed by the $\mathbf{f}_e$, $\mathbf{t}_e$ estimation algorithm at 40 different cartesian pose. The points in the 3D space represent 40 different static poses of LWR with FS attached to the ee, expressed in roll, pith, yaw convetion. The color scale shows the $\mathbf{rmse}_f$ and $\mathbf{rmse}_t$ associated at each cartesian pose, starting from the minimum value (shows in blue point) until the max value (shows in red point).

In Table 4.4 are presented the $\mathbf{rmse}_f$ and $\mathbf{rmse}_t$ of the overall data recorded in 40 trials in x, y, z axes.

Table 4.5 shows the correlation coefficients and the corresponding p-values of the Pearson Correlation between the orientation of the force sensor and the $rmse_{f_{x,y,x}}$ and $rmse_{t_{x,y,x}}$ of the estimation error computed on x, y, x axes.

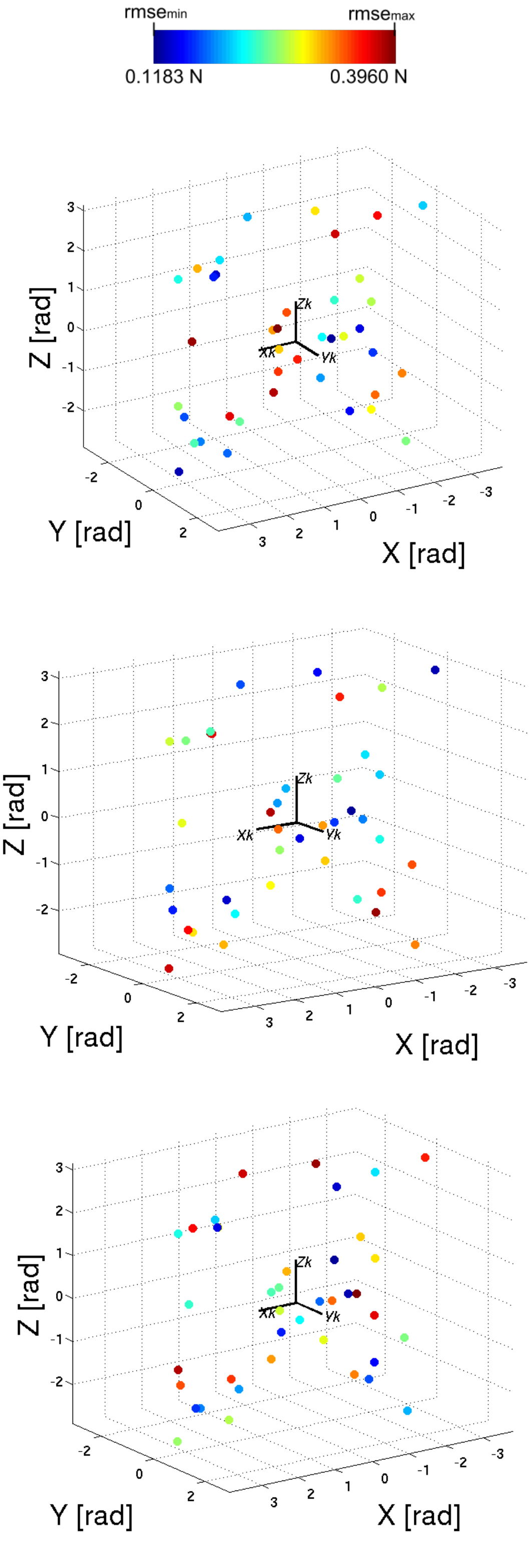


Figure 4.10: Population of $\mathbf{rmse}_f$ of the estimation error in x, y, z (from top to bottom) depending on the orientation of the FS in 40 static pose with respect to $\{s_b\}$

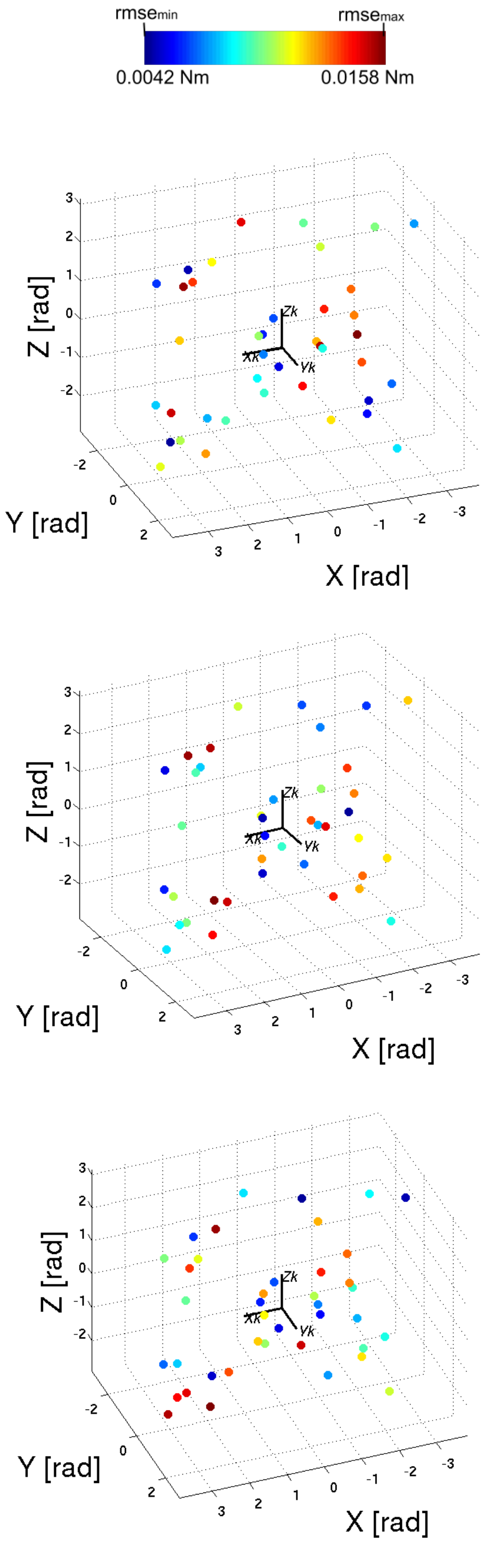


Figure 4.11: Population of **rmse**$_t$ of the estimation error in x, y, z (from top to bottom) depending on the orientation of the FS in 40 static pose with respect to $\{s_b\}$

$\mathbf{rmse}_f$ [N]	
x	0.1071
y	0.2948
z	0.3590

$\mathbf{rmse}_t$ [Nm]	
x	0.0142
y	0.0129
z	0.0061

Table 4.4: Value of $\mathbf{rmse}_f$ and $\mathbf{rmse}_t$ of the estimation error of 40 static robot pose on x, y, x axes in $\{fs\}$

	correlation		
force	roll	pitch	yaw
$rmse_{f_{xyz}}$[N]	-0.0483	-0.4292	0.1044
torque			
$rmse_{f_{xyz}}$[Nm]	-0.0529	-0.0602	0.0155

p-value		
roll	pitch	yaw
-0.7671	0.0057	0.5216
0.7460	0.7123	0.9243

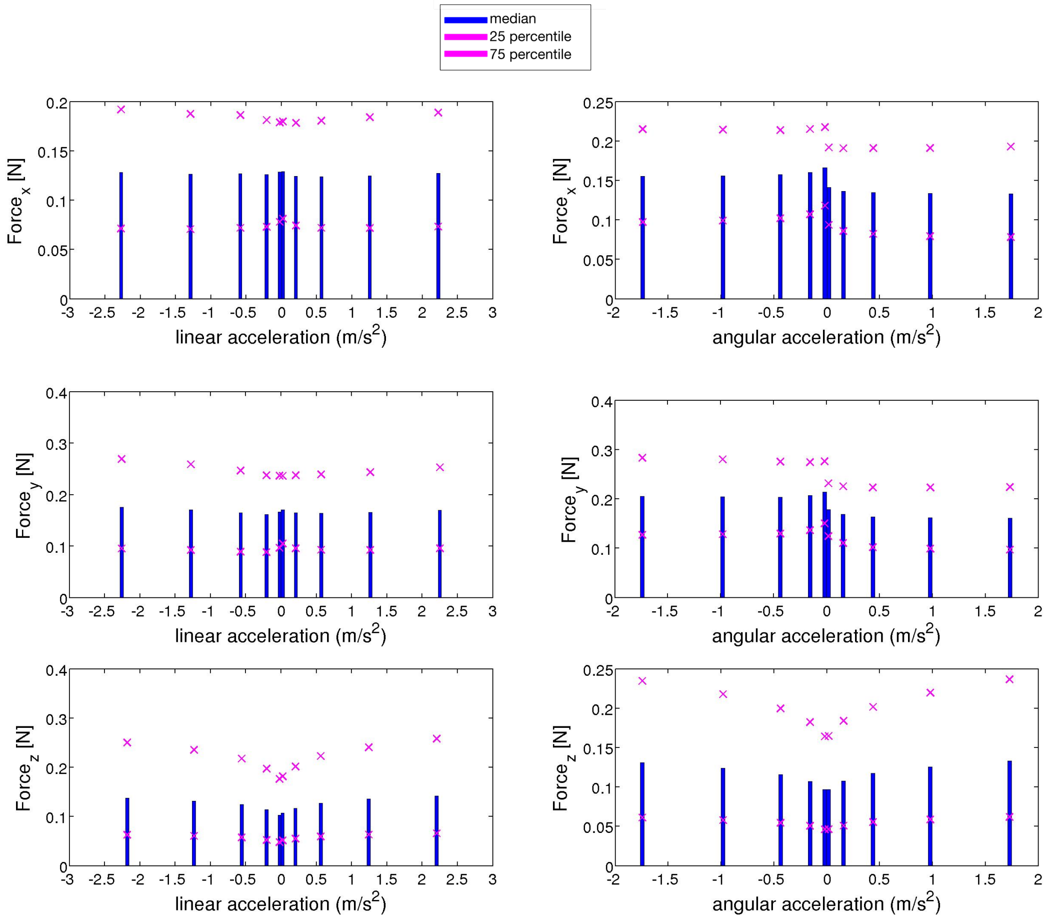
Table 4.5: Correlation between the orientation of the force sensor $\{fs\}$ expressed in roll pitch yaw convention with respect to $\{s_b\}$ and the $rmse_{f_{x,y,z}}$ and $rmse_{t_{x,y,z}}$ of the overall xyz estimation error

4.3.2.2 Dynamic condition

Figure 4.12 and 4.12 show analysis of $\mathbf{err}_f$ and $\mathbf{err}_t$ committed in the estimation. Column (a) and column(b) represent respectively analysis of linear and angular movements.

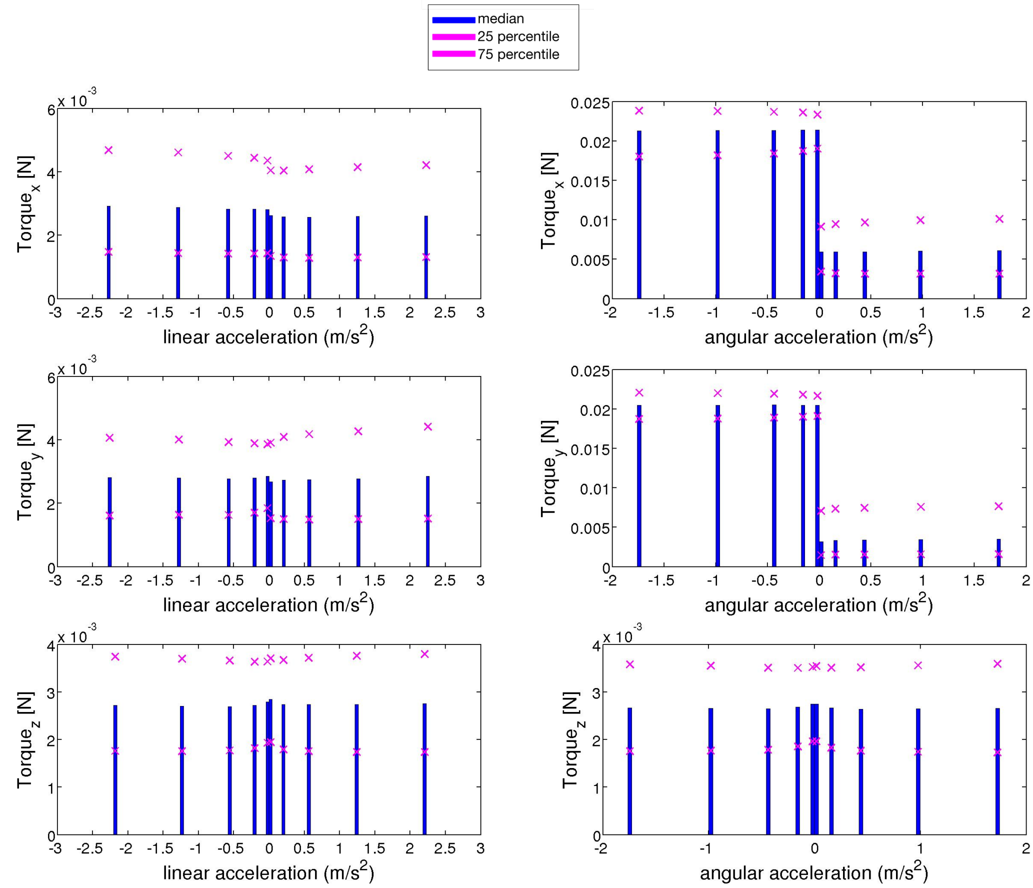
The bars show the *median* of the $\mathbf{err}_f$ and $\mathbf{err}_t$, the crosses are the 25° and 75°*percentile* of $\mathbf{err}_f$ and $\mathbf{err}_t$ at different acceleration.

Table 4.6 shows the correlation values computed between $\mathbf{rmse}_f$ and $\mathbf{rmse}_t$ at each value of deceleration and acceleration and the vector of acceleration and deceleration.



(*a*) (*b*)

Figure 4.12: (*a*) *rmse*, *median*, 25° and 75°*percentile* of $err_{\mathbf{f}}$ at different acceleration during linear movement along x,y,z cartesian axis - (*b*) *rmse*, *median*, 25° and 75°*percentile* of $err_{\mathbf{f}}$ at different acceleration during angular movement around x,y,z cartesian axis



(*a*) (*b*)

Figure 4.13: (a) $rmse$, $median$, 25° and 75°$percentile$ of $err_{\mathbf{t}}$ at different acceleration during linear movement along x,y,z cartesian axis - (b) $rmse$, $median$, 25° and 75°$percentile$ of $err_{\mathbf{t}}$ at different acceleration during angular movement around x,y,z cartesian axis

	correlation: linear motion	
force	dec	acc
x	0.9699	0.9526
y	0.9906	0.9480
z	0.9950	0.9950
torque		
x	0.9726	0.9989
y	0.9726	0.9474
z	0.9832	0.9828

	correlation: angular motion	
force	dec	acc
x	0.6285	0.9751
y	0.9790	0.4585
z	0.9748	0.9754
torque		
x	0.8542	0.9028
y	0.8465	0.9232
z	0.9640	0.9643

	p-value (linear motion)	
force	dec	acc
x	0.0062	0.0123
y	0.0011	0.0141
z	0.0004	0.0004
torque		
x	0.0054	0.0004
y	0.0054	0.0144
z	0.0026	0.0027

(*a*)

	p-value (angular motion)	
force	dec	acc
x	0.2562	0.0047
y	0.0036	0.4374
z	0.0048	0.0046
torque		
x	0.0653	0.0359
y	0.0705	0.0252
z	0.0081	0.0081

(*b*)

Table 4.6: Correlation and p-values between error and velocity: (*a*) linear movement (*b*) angular movement

Chapter 5

Discussion and Conclusion

5.1 Discussion and Conclusion

The aim of this work was to dynamically compensate the force and torque exerted by the weight of the tool on the force sensor attached to the slave robot, in order to reproduce on the master side to the surgeon only the interaction force of the remote surgical tool with the tissues.

The dynamic compensation algorithm described estimates the forces and torques exerted on the force sensor by a known geometry tool during its motion and remove them form the forces and torques measured by the force sensor.

First of all, the force and torque measured are filtered with a notch filter in order to eliminate the noise components.

Considering that the signal to noise ratio between the signals measured and filtered is equal to 0.2900 N, the filtering results not optimal, but it is the better choice to deal with the group sample delay of the filtered signals, which is 1 sample at 9 Hz. Moreover, the noise filtering may be not optimal due to the sampling frequency of the force and torque signals. The serial port communication with the force sensor limits the frequency acquisition at 85Hz. This can lead to aliasing in the frequency analysis causing the incorrect identification of the real noise components.

The second step is the estimation of force and torque acting on the force sensor due to the weight of the tool. The error of estimation is evaluated in static and dynamic condition.

In static condition the $rmse_f$ of the estimation error is below 0.3590 N on the three axes, and considering that the accuracy of the force sensor Nano25 is 1.0 N and the resolution is 0.083 N, it is acceptable because is $< \frac{1}{2}$accuracy. The same results are shown about the torque estimation, the $rmse_t$ of the estimation error is below 0.0142 Nm on the three axes, and considering the accuracy of the torque sensor Nano25 (0.03 Nm) and the resolution (0.0015 Nm) it is acceptable because is $< \frac{1}{2}$accuracy.

In dynamic condition, for both linear and angular movements, the median accuracy of the force estimation error is below 0.2 N, with the same considerations made previously on the force sensor. While the median accuracy of the torque estimation error is below 0.003 Nm. These results are satisfactory, considering the force sensor characteristics, as said before. However, this errors in the estimation are not acceptable if the system is proposed to be used in neurosurgery. The force of contact between the instruments and the brain are near to 0.2N [45], and the force sensor used in this project is not so accurate in discriminating this type of force. The solution could be to use a force sensor with higer accuracy, that can lead also to better estimation.

The correlation between the $rmse_{f,t}$ of the estimation error for each x, y, z axis and the acceleration shows that the more the absolute value of the acceleration is high the more will be the estimation error. Higher errors are caused by the estimation of velocity and acceleration of the tool with FOAW Filter.

In the estimation of force and torque, during angular movements, there is a step in the behaviour of the error estimation. This is due to an error in the identification of the inertial matrix of the tool. If the tool is not exactly symmetrical, the geometric computation of the inertial matrix of a solid cylinder is not correct and can lead to error in the estimation.

Once the force measured by the force sensor is filtered and compensated, it has to be sent to the haptic interface, in order to be reproduced at the hand of the surgeon. Although reflecting the encountered forces back to the human operator enables the human to rely on his tactile sense along with visual sense, it may cause instability in the system if delays are present in the communication of data between HI and LWR.

The analysis on the communication delay shows that the median of the delay between master and slave position is equal to 5 samples. This means that, recording the position of master and slave at 100Hz, the median delay in second is 50ms. The interquartile range is 7.5 ; where value equal to 0 means that the delay is below 10ms. These delays are tested only between the position of master and slave, and considering that the median computational delay δ_c is 7.0881e-06 and is negligible, the delay due to the communication between the devices in the local network is 50ms. This delay is also involved in the exchange of forces data between the force sensor and the haptic interface. Assuming that the computational delay of the force filtering and compensation, and the delay introduce by the force filtering is negligible with respect to the local network delay, the surgeon receive the force data delayed by 50ms. If he want to perform a task and manage the surgical tool on the slave to go deep in the brain of 1 μm (a safe condition), considering the 50ms delay, tha velocity of the task has not to overcame 0.125m/s.

5.2 Future works

The bilateral telemanipulation system presented allows to perform task from a remote site, tracking the movement of the haptic interface, and reproducing to the surgeon the only interaction force with the remote tool with the environment. Several possible future developments might be considered.

- A force sensor with higher accuracy, better resolution and faster data communication. With the actual force sensor the identification of frequency components of noise is not reliable due to the low sampling data. Hence, increase in the communication frequency can improve the quality of the force compensated in terms of noise reduction. The higher is the accuracy and resolution of the force sensor, the lower the estimation error, and the small forces generated in the brain contact can be detected.

- An haptic interface able to reproduce not only force in x,y,z axes, but also torques. The PHANToM Omni is actuated only on three DoFs and is capable to reproduce only force, but adding to the surgeon the information of torques could improve the transparency of the system and give better information about the task performed and the tissue in contact.

- Automatic estimation of internal matrix of the tool. In this project the estimation is computed using the geometric information of the tool. However, in case of surgical instruments with complex geometry there are algorithm able to identify the inertial matrix, starting from the Newton-Euler models and using least square estimation methods [32].

- Development of a novel bilateral control approach with a decoupled local force control at the slave site, and an online environment parameters estimation and adaptation, in order to improve the transparency in force feedback of the system [46]. The error in position between master and slave and caused by the communication delay is corrected introducing a virtual spring between master and slave. The desired force is produced on master by the virtual spring based upon the position difference between master and slave ee. A force controller on the slave enforces the contact force to track this desired force, while the desired force is fed back to the user at the master.

Appendix

The Best fit FOAW [44] computes the derivative of the signals as the linear regression of the sample in a changing window. It is an adaptive filter, so the window size should be short when the velocity is high and should be large when the velocity is low producing more precise estimates.

Provided that the signal is well represented by its samples, increasing the window size is equivalent to decreasing the sampling rate. Thus, a large window introduces time delays and also reduces the estimation reliability.

The main characteristic of the FOAW filter is that, in order to trade precision against reliability, the window size should be selected adaptively depending on the signal itself. Noise reduction and precision put a lower bound on the window size, while reliability provides an upper limit for the window lenght.

This filter find the longest window which satisfies the accuracy requirement, solving a min-max problem. A criterion is established to determine whether the slope of the linear regression of the samples in the window approximates reliably the derivative of a signal between two samples x_k, x_{k-n}. The solution can be stated as finding a window of lenght n where $n = max\{1, 2, 3, ...\}$such that

$$\left|y_{k-1} - y_{k-1}^{L}\right| \leq d, \;\; \veebar i \epsilon \{1, 2, ..., n\} \tag{5.1}$$

where $y_{k-1}^{L} = a_n + b_n(k-1)T$, given that

$$a_n = \tfrac{ky_{k-n}+(n-k)y_k}{n} \quad and \quad b = \frac{\sum_{i=0}^{n} y_{k-i} - 2\sum_{i=0}^{n} y_{k-i}}{Tn(n+1)(n+2)/6} \tag{5.2}$$

Bibliography

[1] K. Hongo, T. Goto, T. Miyahara, Y. Kakizawa, J. Koyama, and Y. Tanaka, "Telecontrolled micromanipulator system (neurobot) for minimally invasive neurosurgery," in *Medical Technologies in Neurosurgery*, pp. 63–66, Springer, 2006.

[2] 11/09/2013, "http://www.neuroarm.org/,"

[3] G. R. Sutherland, S. Wolfsberger, S. Lama, and K. Zarei-nia, "The evolution of neuroarm," *Neurosurgery*, vol. 72, no. Supplement 1, pp. A27–A32, 2013.

[4] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, *et al.*, "The kuka-dlr lightweight robot arm-a new reference platform for robotics research and manufacturing," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pp. 1–8, VDE, 2010.

[5] "http://www.orocos.org/," 19/08/2013.

[6] T. Ortmaier, B. Deml, B. Kubler, G. Passig, D. Reintsema, and U. Seibold, "Robot assisted force feedback surgery," pp. 361–379, 2007.

[7] G. Tholey, J. P. Desai, and A. E. Castellanos, "Force feedback plays a significant role in minimally invasive surgery: results and analysis," *Annals of surgery*, vol. 241, no. 1, p. 102, 2005.

[8] H. G. Stassen and G. Smets, "Telemanipulation and telepresence," *Control Engineering Practice*, vol. 5, no. 3, pp. 363–374, 1997.

[9] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[10] R. L'Orsa, C. J. Macnab, and M. Tavakoli, "Introduction to haptics for neurosurgeons," *Neurosurgery*, vol. 72, no. Supplement 1, pp. A139–A153, 2013.

[11] F. Tendick, S. S. Sastry, R. S. Fearing, and M. Cohn, "Applications of micromechatronics in minimally invasive surgery," *Mechatronics, IEEE/ASME Transactions on*, vol. 3, no. 1, pp. 34–42, 1998.

[12] K. Kaltenborn and O. Rienhoff, "Virtual reality in medicine.," *Methods of information in medicine*, vol. 32, no. 5, pp. 407–417, 1993.

[13] Y. S. Kwoh, J. Hou, E. A. Jonckheere, and S. Hayati, "A robot with improved absolute positioning accuracy for ct guided stereotactic brain surgery," *Biomedical Engineering, IEEE Transactions on*, vol. 35, no. 2, pp. 153–160, 1988.

[14] J. M. Drake, M. Joy, A. Goldenberg, and D. Kreindler, "Computer-and robot-assisted resection of thalamic astrocytomas in children," *Neurosurgery*, vol. 29, no. 1, pp. 27–33, 1991.

[15] A. Benabid, P. Cinquin, S. Lavalle, J. Le Bas, J. Demongeot, and J. De Rougemont, "Computer-driven robot for stereotactic surgery connected to ct scan and magnetic resonance imaging," *Stereotactic and Functional Neurosurgery*, vol. 50, no. 1-6, pp. 153–154, 1987.

[16] C. Burckhardt, P. Flury, and D. Glauser, "Stereotactic brain surgery," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 14, no. 3, pp. 314–317, 1995.

[17] E. De Momi and G. Ferrigno, "Robotic and artificial intelligence for keyhole neurosurgery: the robocast project, a multi-modal autonomous path planner," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 224, no. 5, pp. 715–727, 2010.

[18] M. J. Lang, A. D. Greer, and G. R. Sutherland, "Intra-operative robotics: Neuroarm," in *Intraoperative Imaging*, pp. 231–236, Springer, 2011.

[19] K. Hongo, S. Kobayashi, Y. Kakizawa, J.-i. Koyama, T. Goto, H. Okudera, K. Kan, M. G. Fujie, H. Iseki, and K. Takakura, "Neurobot: telecontrolled micromanipulator system for minimally invasive microneurosurgery-preliminary results," *Neurosurgery*, vol. 51, no. 4, pp. 985–988, 2002.

[20] T. Goto, K. Hongo, Y. Kakizawa, H. Muraoka, Y. Miyairi, Y. Tanaka, and S. Kobayashi, "Clinical application of robotic telemanipulation system in neurosurgery: case report," *Journal of neurosurgery*, vol. 99, no. 6, pp. 1082–1084, 2003.

[21] H. I. Son, T. Bhattacharjee, and D. Y. Lee, "Estimation of environmental force for the haptic interface of robotic surgery," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 6, no. 2, pp. 221–230, 2010.

[22] K. Hashtrudi-Zaad and S. E. Salcudean, "Analysis of control architectures for teleoperation systems with impedance/admittance master and slave manipulators," *The International Journal of Robotics Research*, vol. 20, no. 6, pp. 419–445, 2001.

[23] G. De Gersem, "Kinaesthetic feedback and enhanced sensitivity in robotic endoscopic telesurgery," *Catholic University of Leuven*, 2005.

[24] T. B. Sheridan and W. R. Ferrell, "Remote manipulative control with transmission delay," *Human Factors in Electronics, IEEE Transactions on*, no. 1, pp. 25–29, 1963.

[25] S. M. Biggs SJ. Lawrence Erlbaum, 2002.

[26] B. Siciliano and O. Khatib, *Springer handbook of robotics.* Springer, 2008.

[27] M. Tavakoli, A. Aziminejad, R. V. Patel, and M. Moallem, "High-fidelity bilateral teleoperation systems and the effect of multimodal haptics," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 6, pp. 1512–1528, 2007.

[28] J. Arata, Y. Tada, H. Kozuka, T. Wada, Y. Saito, N. Ikedo, Y. Hayashi, M. Fujii, Y. Kajita, M. Mizuno, *et al.*, "Neurosurgical robotic system for brain tumor removal," *International journal of computer assisted radiology and surgery*, vol. 6, no. 3, pp. 375–385, 2011.

[29] R. C. Harwell and R. L. Ferguson, "Physiologic tremor and microsurgery," *Microsurgery*, vol. 4, no. 3, pp. 187–192, 1983.

[30] A. Rossi, A. Trevisani, and V. Zanotto, "A telerobotic haptic system for minimally invasive stereotactic neurosurgery," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 1, no. 2, pp. 64–75, 2005.

[31] S. Katsura, W. Iida, and K. Ohnishi, "Medical mechatronics an application to haptic forceps," *Annual Reviews in Control*, vol. 29, no. 2, pp. 237–245, 2005.

[32] A. Haddadi and K. Hashtrudi-Zaad, "A new method for online parameter estimation of hunt-crossley environment dynamic models," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 981–986, IEEE, 2008.

[33] T. Yamamoto, B. Vagvolgyi, K. Balaji, L. L. Whitcomb, and A. M. Okamura, "Tissue property estimation and graphical display for teleoperated robot-assisted surgery," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 4239–4245, IEEE, 2009.

[34] N. Zemiti, T. Ortmaier, M.-A. Vitrani, and G. Morel, "A force controlled laparoscopic surgical robot without distal force sensing," in *Experimental Robotics IX*, pp. 153–164, Springer, 2006.

[35] K. Nishimura and K. Ohnishi, "Gravity estimation and compensation of grasped object for bilateral teleoperation," in *Advanced Motion Control, 2006. 9th IEEE International Workshop on*, pp. 72–77, IEEE, 2007.

[36] A. J. Silva, O. A. D. Ramirez, V. P. Vega, and J. P. O. Oliver, "Phantom omni haptic device: Kinematic and manipulability," in *Electronics, Robotics and Automotive Mechanics Conference, 2009. CERMA'09.*, pp. 193–198, IEEE, 2009.

[37] "http://www.geomagic.com/en/products/phantom-omni/specifications/," 10/09/2013.

[38] "http://www.ati-ia.com/," 19/08/2013.

[39] "http://www.ros.org," 29/08/2013.

[40] R. Smits and H. Bruyninckx, "Composition of complex robot applications via data flow integration," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 5576–5580, IEEE, 2011.

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

[42] 7/09/2013, "http://geomagic.com/en/products/phantom-omni/overview,"

[43] D. Kubus, T. Kroger, and F. M. Wahl, "On-line rigid object recognition and pose estimation based on inertial parameters," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 1402–1408, IEEE, 2007.

[44] F. Janabi-Sharifi, V. Hayward, and C.-S. Chen, "Discrete-time adaptive windowing for velocity estimation," *Control Systems Technology, IEEE Transactions on*, vol. 8, no. 6, pp. 1003–1009, 2000.

[45] M. A. Howard III, B. A. Abkes, M. C. Ollendieck, M. D. Noh, C. Ritter, and G. T. Gillies, “Measurement of the force required to move a neurosurgical probe through in vivo human brain tissue,” *Biomedical Engineering, IEEE Transactions on*, vol. 46, no. 7, pp. 891–894, 1999.

[46] J. Park and O. Khatib, “A haptic teleoperation approach based on contact force control,” *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 575–591, 2006.