POLITECNICO DI MILANO

Master of Science in Biomedical Engineering

School of Industrial and Information Engineering

THE BIOROBOTICS INSTITUTE

Sant'Anna
School of Advanced Studies – Pisa

Dynamic Modeling of the Da Vinci Research Kit Arm for the Estimation of Interaction Wrench

Advisor: Prof. Elena De Momi

Co-advisor: Prof. Arianna Menciassi

Supervisors: Mohamed Nassim Boushaki,
Margherita Brancadoro

MSc Thesis student:

Francesco Piqué, ID 873785

Academic Year 2017-2018

Contents

List of Figures

List of Tables

Sommario

Nel corso degli ultimi decenni, è stato rivolto sempre maggiore interesse verso la chirurgia minimamente invasiva (*Minimally Invasive Surgery*, MIS). Essa, in contrapposizione alla chirurgia aperta, mira a intervenire limitando il danno per il paziente, riducendo la dimensione delle incisioni, aumentando la sicurezza dell'intervento riducendo i tempi di recupero per il paziente e i rischi di complicazioni. La laparoscopia tradizionale tuttavia limita la destrezza del chirurgo e la coordinazione tra mano e occhio, rendendo la procedura più difficile per il chirurgo e più faticosa. Per questo motivo, sono stati introdotti sistemi robotici per la chirurgia mini-invasiva.

Il robot *da Vinci* è attualmente il sistema più usato per la laparoscopia roboticamente assistita. Sviluppato da Intuitive Surgical, esso ambisce ad integrare l'approccio minimamente invasivo della laparoscopia tradizionale con la tecnologia innovativa della robotica teleoperata e della realtà virtuale. Il robot *da Vinci* consiste di una console la quale consente al chirurgo di teleoperare, tramite due bracci *master*, due bracci *slave* dotabili di diversi strumenti. Il chirurgo riceve feedback visivo da un visore stereo che invia in tempo reale le immagini provenienti da una telecamera endoscopica. Il sistema consente di filtrare il tremore della mano, può scalare i movimenti del chirurgo e consente di raggiugere una coordinazione inedita tra occhio e mano, aumentando così la precisione dell'operazione e l'abilità del chirurgo.

Tuttavia, la versione del robot da Vinci attualmente sul mercato non è dotata di feedback tattile o di forza al braccio *master*. Pertanto, il chirurgo non ha la sensazione delle forze da egli stesso applicate e può contare solo sulla propria esperienza per stimare le forze di interazione tra lo strumento e il paziente. È stato dimostrato che un eventuale superamento di questa lacuna può migliorare la curva di apprendimento per i chirurghi principianti e può ridurre il rischio di danneggiare i tessuti o di rompere le suture. Per questo motivo, le recenti attività di ricerca che utilizzano la piattaforma da Vinci Research Kit (dVRK), ambiscono sempre più alla realizzazione di un feedback di forza che rifletta le forze di interazione tra il braccio *slave* e il paziente verso il braccio *master*.

Pertanto, l'obiettivo di questa tesi è presentare e convalidare un metodo per la stima delle forze e delle coppie di interazione tra la punta del braccio slave e il paziente. Il metodo proposto è *sensorless*, in quanto la stima non si basa su un sensore di forza/coppia, ma viene eseguita indirettamente tramite le coppie misurate a livello dei giunti. Parte cospicua di questo lavoro si concentra sulla modellazione della dinamica del braccio *slave*. Il modello dinamico è stato calcolato utilizzando l'algoritmo di Newton-Euler (NEA) e include l'attrito viscoso e statico. I coefficienti dinamici del modello sono stati identificati utilizzando un approccio ai minimi quadrati, a partire da dati provienienti dal robot ottenuti durante l'esecuzione traiettorie ottimizzate di eccitazione dei giunti. Tali dati includono posizioni e velocita di giunto e coppie di giunto.

Il modello dinamico del braccio è necessario perché la coppia totale di giunto, calcolabile a partire dalle correnti dei motori, include sia la coppia dovuta alla dinamica del robot che la coppia risultante dall'interazione esterna. Infatti, ogni volta che viene applicata una forza esterna alla punta del braccio, i motori devono esercitare una coppia aggiuntiva in modo da resistere a questa interazione e mantenere la posizione dei giunti. Questa coppia deve essere isolata per poter calcolare le forze e le coppie di interazione. Utilizzando le coppie di giunto ottenute dalle correnti degli attuatori e sottraendo da queste le coppie risultanti dalla dinamica del braccio del robot, cioè stimate con il modello dinamico, è possibile ricavare le coppie di giunto dovute solo alle forze e coppie esterne. Queste forze e coppie che agiscono sull'utensile possono essere infine ottenute attraverso l'inversa della matrice Jacobiana trasposta. L'accuratezza di questo metodo è stata valutata confrontando le forze e le coppie stimate con quelle misurate da un sensore forza/coppia (ATI mini 45). È stato dimostrato che le forze e le coppie esterne sono ben stimate rispetto a quelle misurate, con una deviazione quadratica media normalizzata tra il 6 e il 15 percento.

Abstract

During the last decades, surgical procedures have evolved towards a minimally invasive approach. Minimally Invasive Surgery (MIS), as opposed to open surgery, aims at lowering the damage to the patient's body by intervening on the patient with smaller incisions, so as to increase safety, to lower the wound recovery time and in general the risk associated to a medical procedure. The traditional laparoscopic approach however limits the dexterity of the surgeon and the hand to eye coordination, making the procedure more difficult for the surgeon and more fatiguing. For this reason, robotic systems for minimally invasive surgery were brought about.

The *da Vinci* system is currently the leading system for Robot-assisted Minimally Invasive Surgery (RMIS). Developed by Intuitive Surgical, it merges the minimally invasive approach of laparoscopy with the cutting-edge advances made in teleoperated robotics and virtual reality. It consists of a console that allows the surgeon to teleoperate, by means of two master arms, two slave arms that can be equipped with several tools such as grippers, graspers etc. The surgeon receives visual feedback from a stereo viewer which displays in real time the visual information filmed by an endoscopic camera. The system allows to filter the surgeon's hand tremor, it provides scaling of movements and an augmented hand to eye coordination thus increasing the precision and capability of the surgeon. However, The commercialized ver-

sion of the da Vinci robot currently lacks of the haptic feedback to the master arm. Thus, the surgeon has no haptic sense and can rely only on experience to be aware of the forces of interaction between the tip of the tool and the patient tissue. For this reason, in the recent research activities using the da Vinci Research Kit (dVRK) platform, the reflection of the interaction force between the slave tool tip and the environment to the master manipulator is a topic of high interest. It has in fact been proven that the implementation of force feedback benefits the learning curve of novice surgeon and reduces the risk of damaging tissues or breaking suture knots.

Therefore, the goal of this thesis is to present and validate a sensorless model-based method for the estimation of the forces and torques of interaction at the tip of the slave arm of the da Vinci Research Kit. Sensorless, because the estimation of the forces and torques of interaction does not rely on any force/torque sensor, but it is carried out indirectly by means of the torques measured at the actuators level. It is model-based, because the dynamic model of the slave arm is computed and explicitly taken into account. The dynamic model of the dVRK slave arm, including viscous and static joint frictions, was computed using the Newton-Euler algorithm. The base parameters of the model were identified using a least squares approach, from data obtained while exciting the robot with optimal trajectories. These data include joint position and velocities and actuator torques. The dynamic model is needed because the total joint torque, estimated by means of the motor current, includes both the torque due to the dynamics of the robot and the torque resulting form external interactions. In fact, whenever an external force is applied to the tip of the arm, the motors must exert an additional torque so as to resist to this interaction and to reach the desired pose of the end effector. This torque must be separated from the total one

in order to compute the forces and torques of interaction. The idea is to use the actuator torques obtained from the measured motor currents and to subtract the torques resulting from the dynamics of the robot arm, to be estimated with the dynamic model. The resulting torques are therefore only due to the external forces and torques acting on the tool. Finally, the forces and torques of interaction are obtained through the inverse transpose of the Jacobian matrix. The accuracy of this method is assessed by comparing the estimated wrench to the one measured by a force/torque sensor (ATI mini 45). It was shown that the external wrench is well estimated compared to the measured one, with a normalized root mean squared deviation (NRMSD) between 6 and 15 percent.

Chapter 1

Introduction & State of the Art

Minimally Invasive Surgery (MIS) is an umbrella term which defines the medical procedures and devices which aim at intervening on the patient with the lowest damage possible so as to reduce the trauma for the patient and the time of recovery. Laparoscopic surgery for example, or endoscopic surgery are all solutions which allow to operate the patient without performing open surgery, avoiding the trauma that the latter approach entails. On the other hand, in the passage from open surgery to MIS, some limitations are introduced such as the loss of the haptic sensation for the surgeon, which will be discussed later in this thesis. Several methods for MIS have been implemented relying on surgical robots, of which an overview is given.

1.1 Surgical Robotics

In the last decades, robot manipulators have found application in the medical field improving the quality of patient care. Surgical robots have been increasingly adopted to enhance the surgeon's ability and performance, to improve the accuracy, the precision and the speed of the operation, and to reduce trauma and complications for patients [1]. This was made possible by the rapid technological advancements in the field of industrial automation which have eventually found application in the operating rooms. The improvements in design, kinematics, and control algorithms have made feasible the realization of robotic manipulators with suitable accuracy and safety requirements suitable for use in a clinical environment. The use of robotic manipulators, coupled with the computing abilities of modern computers and to increasingly accurate pre-operative and intra-operative imaging techniques, has paved the way for Computer-Aided-Surgery / Computer-Integrated-Surgery. This approach comprises the acquisition and processing of pre-operative images and the envisioning of a surgical plan to be executed and monitored in real time during the procedure. The use of robotic manipulators allows to exploit the numeric nature of its kinematic parameters so as to obtain a quantitative, rather than qualitative, information (e.g. the position of a biopsy needle) and compare it to the data from a surgical plan.

The first robotic manipulators used in clinics were actually industrial robots, exploited for their geometric accuracy and stability. In 1985, The Unimation Puma 200 (Programmable Universal Machine for Assembly) robot (Fig. 1.2) was used in the operating room to perform the first robotized stereotactic brain biopsy [2]. In this context, the Puma 200 is used as a passive needle holder capable of positioning and orienting a guide in such a way that a biopsy needle can reach the surgical target inside the brain following accurately a predetermined trajectory.

The trajectory is planned in advance using data acquired through CT imaging. The target is localized on the CT image and its coordinates are computed with respect to the base of a stereotactic frame surgically fixed on the patient skull. After having bolted the robot to the patient's couch,

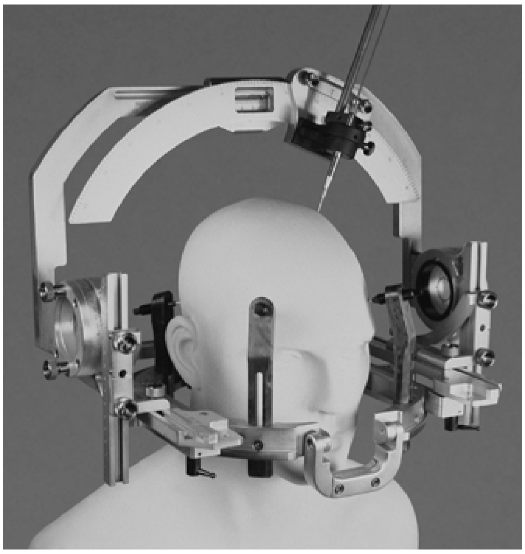


Figure 1.1: The Leskell Frame.

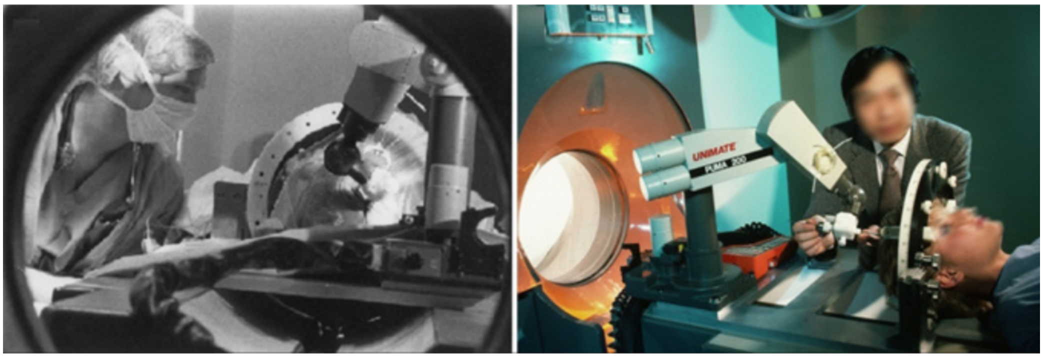


Figure 1.2: The Unimation PUMA 200, used to perform the first stereotactic brain biopsy in 1985 [2].

the coordinates of the target are computed with respect to the robots reference frame. The robot can then move smoothly its links to provide a virtual fixture near the chosen entry point on the skull, so as to ensure a stable trajectory inside the skull for a biopsy needle or an electrode for deep stimulation. This procedure, in addition to ensuring a reliable trajectory planned to avoid sensible structures in the brain, allows to avoid the tedious and time consuming operations required to orient the needle guide that were previously done manually with the Leskell frame. Moreover, if multiple entry points are expected, the robot can re-orient rapidly the probe guide from one entry point to another resulting in a faster and more flexible procedure. State of the art systems stemming from this pioneering work include the Renishaw Neuromate, FDA-cleared for deep brain stimulation (DBS) and stereoelectroencephalograpy (SEEG), and the Pathfinder surgical robot which provides also an embedded optical localizer system [3]. Other robotic systems for neurosurgery in research include the NeuroArm system [4], a telerobotic system for neurosurgery compatible with intra-operative MRI image guidance. Six DOF robotic arms have been exploited with a similar approach also for spine surgery (Mazor X, ROSA spine). Similarly to neurosurgical devices, these manipulators orient their end effector providing a virtual fixture to suitably assist the placement of a screw pedicle [5] (Fig. 1.3).

Along with neurosurgical applications, orthopedics was one of the first areas in which robotic applications were developed; this development was mainly motivated by the rigidity of the surgical target [6]. The amount of deformation undergone by bones during a surgical procedure is in fact negligible with respect to the one affecting soft tissues, which must be estimated using complex biomechanical models. This results in simpler techniques for planning and robotic navigation, because the plan outline on pre-operative images is considered reliable also intraoperatively.

The TSolution One® system by THINK Surgical (formerly ROBODOC), for example, is a system for hip and knee surgery which performs bone preparation prior to the placement of a prosthetic implant [7]. It consists of two modules: the TPLAN® workstation for 3D pre-operative planning and TCAT®, a computer-assisted tool for the execution of the procedure. A 3D patient specific model of the bone anatomy and bone density is constructed

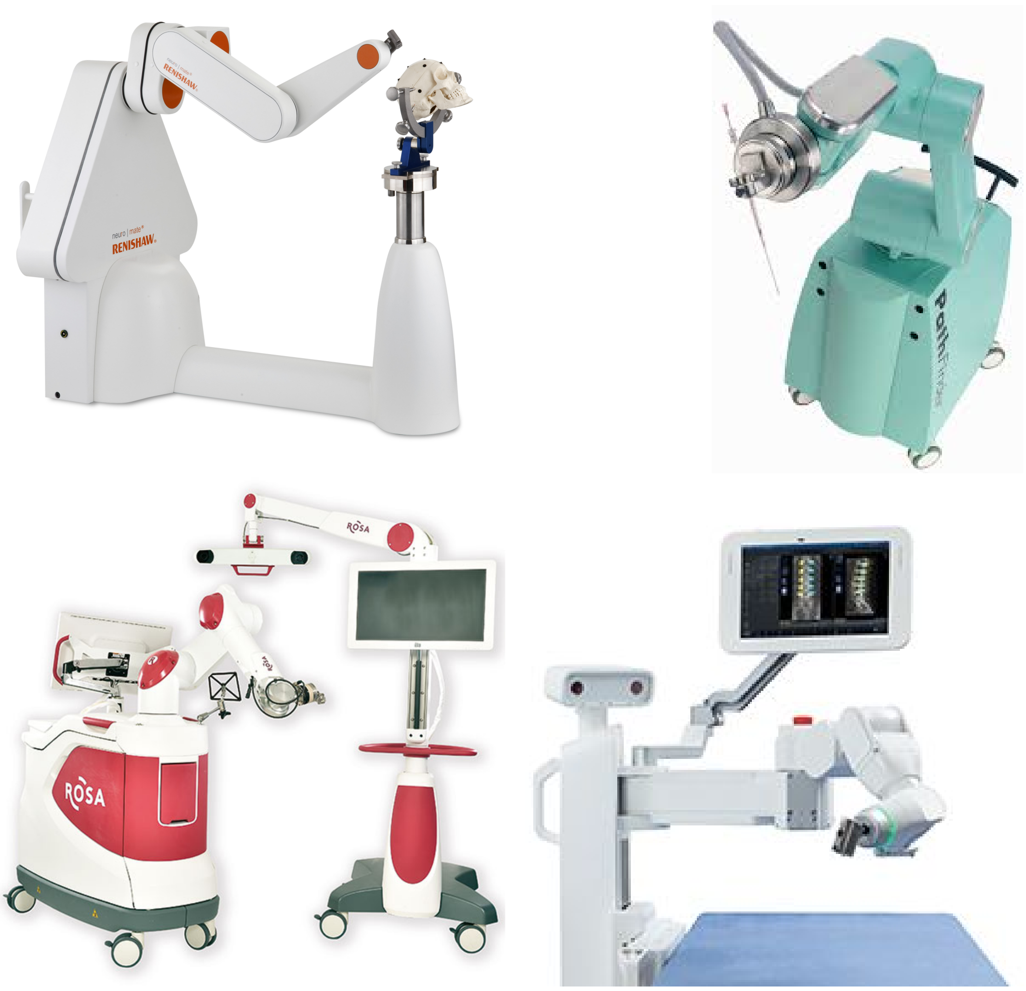


Figure 1.3: Starting from the top left in anticlockwise direction: the Reinshaw Neuromate, the Pathfinder, the ROSA spine, Mazor X.

from pre-operative CT images and it is used to plan the bone preparation procedure specifying the type of implant and the desired fit and alignment. Subsequently, after bone registration is performed, the system actively mills and cuts the bone under surgeon supervision. The rigidity of the bone tissue ensures a more stable correspondence between the patient specific model and the actual tissue during the procedure.

Surgical robotic devices can be classified according to their degree of autonomy [6], [1]. The autonomy of a surgical robot indicates to which amount its functionalities are driven by the commands of a surgeon or operator during the surgical procedure (i.e. no autonomy) or are automated via a surgical plan. A fully automated surgical procedure would therefore be defined as totally autonomous. The TSolution One® system belongs to this category since the robot autonomously performs the bone preparation technique according to the plan, with only the supervision of the surgeon. The MAKO system, commercialized by Stryker, is another popular FDA-cleared system for hip and knee replacement. This system can be regarded instead as a semi-autonomous device because, unlike TSolution One®, it does not autonomously perform the operation but assists the surgeon with virtual fixtures, which restrain the tool from exiting the area specified by the pre-operative plan [8]. The two robotic platforms are shown in (Fig. 1.4). Fully passive robots are usually teleoperated manipulators, as described in the following section.

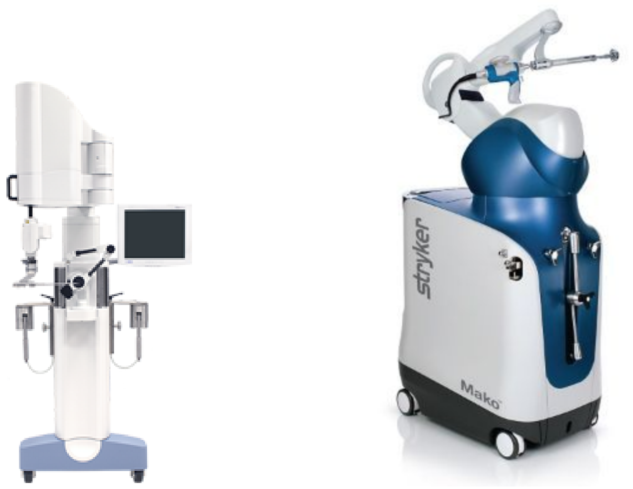


Figure 1.4: From left to right: the TSolution One® system by THINK Surgical and the MAKO system by Stryker for total hip replacement.

1.2 Teleoperation

Teleoperation is generally the control strategy of choice for completely passive manipulators. It involves the control of a master manipulator by the operator which causes the displacement of the robot manipulator, which in this context is known as “slave”. In the first teleoperated robots mechanical cables and linkages were used as communication links between the master and the slave side. These systems were intended for manipulating hazardous radioactive materials from a safe distance, typically from the other side of a thick glass (Fig. 1.5). It was not long before the mechanical connections between the operator and the robotic slave were replaced by electrical signals exchanged by sensors and actuators, allowing for remote control through greater distances [9].

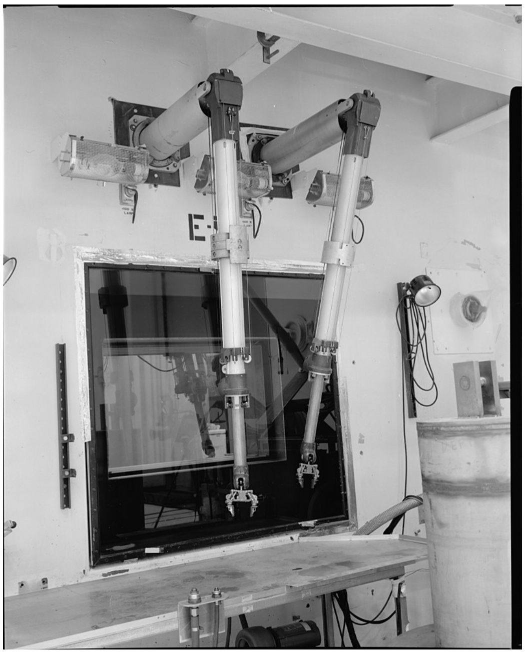


Figure 1.5: Remote manipulator arms designed for the manipulation of radioactive material (Engine Maintenance Assembly & Disassembly Facility, Area 25 of the Nevada Test Site).

Today, teleoperated robots are widely used in nuclear plant decommissioning, environment exploration and of course surgery [10], [11]. The idea of teleoperated robots able to perform surgery was first brought about in the 1970's, when researchers from the National Aeronautics and Space Administration (NASA) imagined to operate astronauts in space by means of surgical robots commanded from earth by expert surgeons [12]. However this idea was soon abandoned because of the time delays introduced by the large distances [8]. Later the same idea was pushed forward by the Defense Advanced Research Projects Agency (DARPA) which wished to perform remote surgical operation on soldiers directly on the battlefield [13]. The impulse given by the DARPA to this kind of projects ultimately led to the development of the first teleoperated surgical devices. The most popular of these devices is the Da Vinci Surgical System which will be dealt with in detail in the next sections.

Currently, several devices available on the market are controlled by means of teleoperation. Such devices include the Flex Robotic System by Medrobotics which provides a flexible endoscope steerable via a remote controller to perform transoral and transanal procedures [14]. Research devices which involve a teleoperated control include the Preceyes system for eye surgery, and the aforementioned Neuroarm system for neurosurgery (Fig. 1.6).

1.2.1 The Da Vinci Surgical System

The Da Vinci Surgical System is nowadays the most popular surgical robot for Robot-assisted Minimally Invasive Surgery (RMIS). Installed in more than 4400 hospital facilities around the world, it allows to perform minimally invasive laparoscopic surgery with the benefits of robotic teleoperation. The scope of RMIS is to push forward the possibilities opened up by traditional laparoscopic surgery [15]. In the traditional scenario, hand

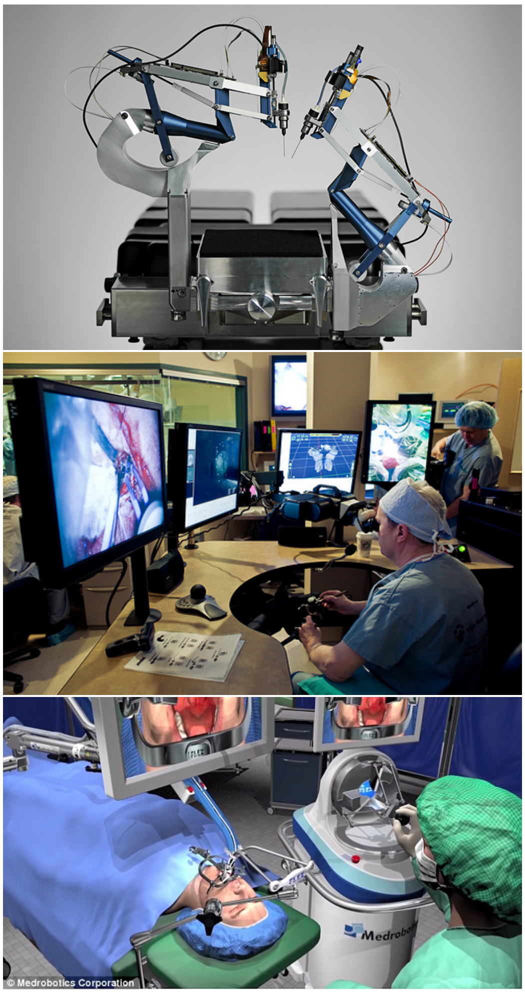


Figure 1.6: From top to bottom: The Preceys system for teleoperated eye surgery, the Neuroarm for teleoperated MRI-compatible neurosergory and the FLEX endoscopic system.

held tools are inserted in the operating space by means of a trocar and a small incision while the laparoscope provides visual feedback through a fiber optic cable. The surgeon can than see the images on a side monitor screen, while operating by means of th ehand held tools. Traditional laparoscopy works well for simpler operations such as laparoscopic cholecystectomy (removal of the gall bladder), oophorectomy (removal of an ovary or ovaries), and hysterectomy (removal of the uterus) [13]. More complex operations involving more than just tissue excision and simple sutures were not carried out in a significant manner. These shortcomings have pushed researchers to develop robotic systems specifically tailored for robot-assisted minimally invasive surgery, such as the Da Vinci Surgical System.

The first developments of the Da Vinci Surgical System date back to 1995, when Dr. Frederick Moll founded Intuitive surgical. In 1997 the first human intervention was performed with a Da Vinci prototype. In the same year Computer Motion, the main competitor of Intuitive, launched its platform for RMIS, the Zeus Robotic Surgical System (Fig. 1.7) [16]. Computer Motion was one of the leading companies in surgical robotics, having specialized in robotic tools for laparoscopy. The Zeus Robotic Surgical System was an evolution of their earlier product AESOP (Automated Endoscopic System for Optimal Positioning). AESOP was an endoscopic voice controlled camera holder to be used in laparoscopic surgery and it was the first surgical robot to receive FDA clearance. In the year 2000, Computer Motion sued Intuitive Surgical for patent infringement, starting a legal litigation which finally ended in 2003 with the merger agreement between Intuitive Surgical and Computer Motion, and with the phasing out of the Zeus system in favor of the Da Vinci [13].

The Da Vinci Surgical System is composed of a patient cart, a surgeon console and a vision cart (Fig. 1.8). The patient cart consists of four or more arms called patient side manipulators, which can be equipped with different tools conceived to operate inside the patient body through a small incision. One of the arms is equipped with an endoscopic camera which provides visual information to the surgeon console. The arms are teleoperated by the surgeon from the surgeon console by means of two master arms. The foot-switch at the surgeon console side allows the surgeon to switch between

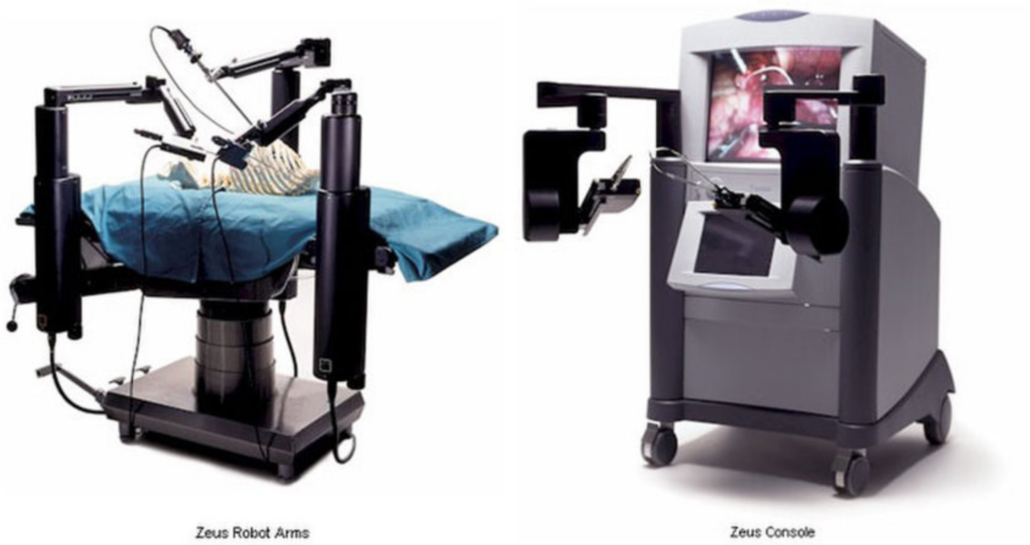


Figure 1.7: The Zeus Robotic Surgical System by Computer Motion.

operating patient side arms. Simultaneously, the surgeon visualizes the operating space from the high resolution stereo viewer present in the surgeon console (Fig. 1.9). The vision cart provides communication across the da Vinci system components, and integrates image and signal processing. A display also shows a live feed of the procedure, available to all the personnel in the operating room. The immersivity guaranteed by the stereo viewer is an appealing feature of the system because the surgeon feels as if he or she were performing open surgery, while instead minimally invasive surgery is being performed ensuring all the benefits for the patient in terms of post-operative pain, blood loss and in duration of hospital stay. The robotic teleoperation system is able to cancel hand tremor, to augment the natural range of motion of the human hand, to scale the movements of the surgeon and to eliminate the fulcrum effect, all disadvantages that were part of conventional laparoscopic surgery. Moreover the surgeon operates while seated and not standing up, resulting in less fatigue.

One drawback of the Da Vinci Surgical System is the lack of the feeling of touch for the surgeon [17]. Even the residual kinesthetic feeling which is transmitted by the traditional hand held laparoscopic tools is lost in the case of RMIS. The reconstruction of such a feeling will be dealt with in the next section.

1.3 Haptics

Haptics, from the greek word $\alpha\pi\tau\kappa\iota\kappa o\varsigma$ (haptikos) meaning "pertaining to the sense of touch”, indicates the research field which aims at recreating the sense of touch by means of forces or vibrations to the user. Along with vision and hearing, touch is one of the main senses which allow a subject to experience the external world. Therefore, research in haptics is of crucial importance for the implementation of virtual environments as well as for the remote reconstruction of real environments. This type of sensory translation

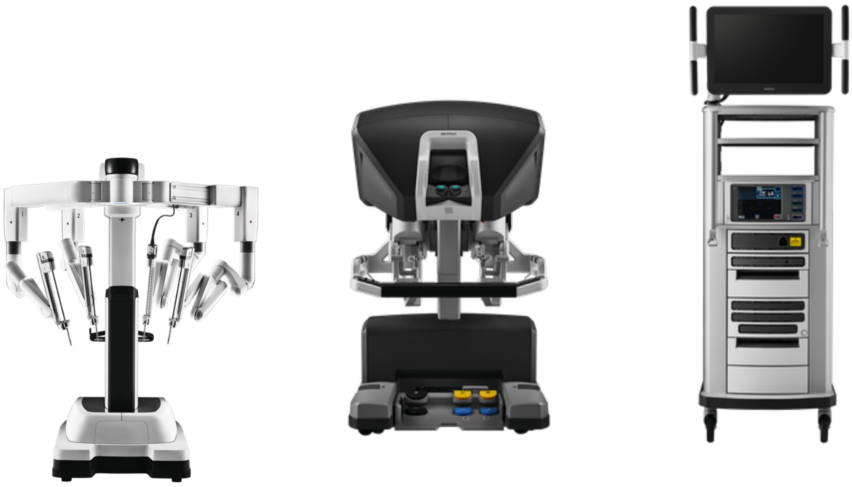


Figure 1.8: The Da Vinci Surgical Systems Components. From left to right: the patient cart, the surgeon console and the vision cart.

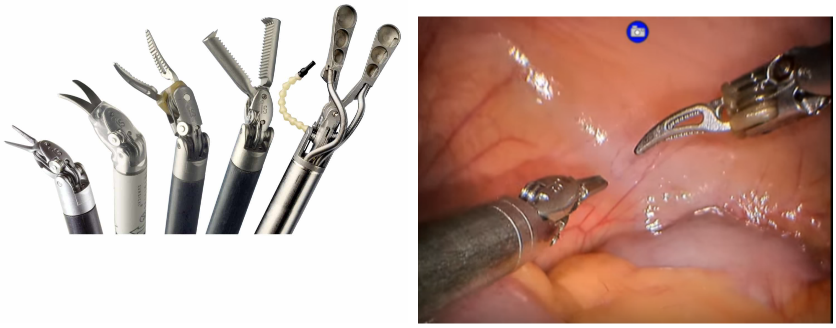


Figure 1.9: Left: different tools with which the patient side manipulators can be equipped. Right: an example of the scene viewed by the surgeon through the high resolution stereo viewer.

has found application in several fields such as gaming, flight simulators, but also in the development of artificial hands, prosthetic limbs and in surgical robotics [18]. In the field of surgical robotics, haptic interfaces have found application surgical simulators for training and in the recreation of the haptic sense in teleoperated surgical systems, such as the Da Vinci. It is useful to point out that the tactile sensation of an object is not reducible to a single sensation, but it is constituted by the perception of a range of properties such as temperature, texture, skin stretch, vibrations and forces. While information regarding texture, temperature and skin stretch is detected by receptors placed on the surface of the skin, force and torque information is felt through receptors present in muscles and tendons. Consequently it is possible to classify tactile information, and the respective devices to reconstruct that information, into two categories: cutaneous and kinestethic [19].

In open surgery both the cutaneous and the kinesthetic information deriving from the interaction between the surgeon and the tissue are preserved, since the surgeon can paplate directly the tissue with gloved hands. Traditional laparoscopic tools and hand held robotic devices such as the MAKO by Stryker or the TSolution One naturally transmit part of the kinestethic sense to the surgeon, while discarding all cutaneous information. In teleoperated devices, such as the Da Vinci, all haptic sensation is lost if a proper feedback is not developed. The surgeon relies only on experience and on visual feedback to estimate the exerted force [20]. It has been shown, in the context of RMIS, that the implementation of a force feedback using the master device as a kinesthetic interface could provide several potential benefits. In suturing tasks, the force feedback improves the learning curve of novice surgeons and reduces the risk of breaking the suture knot [21]. In palpation tasks, the force feedback allows for an easier detection of hard lumps under the tissue, such as tumors [22]. In general the use of force feedback reduces the risk of tissue damaging [23].

Force feedback implies the recreation of the forces of interaction from the slave side, to the master side by means of actuators. Therefore, the interaction forces must first be sensed or estimated through a suitable approach, then the stability of the master-slave loop must be ensured - even if it is not addressed in this work - and finally the forces must be correctly displayed to the surgeon. The reflection of forces to the master arm requires implementing complex control architectures which all entail stringent requirements for robustness and stability [24] [25]. These issues can be avoided by exploiting different methods of displaying the force to the user such as auditory cues [17], visual displays [21], or asymmetric force feedback [26].

1.3.1 Force Estimation Methods

In order to implement a force feedback from the slave arm to the master arm, the force of interaction between the slave arm and the patient's tissue must be sensed or estimated. Many research works have been conducted in order to develop novel force sensors either to be integrated on the shaft or on the tip of the instrument of the da Vinci (Fig. 1.10). In [27] an optical sensor is placed on the terminal part of the trocar. In [28] strain gauges are placed on the shaft of the tool, while in [29] they are placed on the shaft of the cables which drive the wrist of the tool. Piezoresistive sensors were also used, and were placed at the tip of the tool so as to estimate the interaction [30]. In some cases, the design of completely new tools equipped with a force sensor is required [31]. However, these solutions entail sterilization and biocompatibility issues, as well as a high costs. In this thesis a sensorless approach is studied, which could potentially solve these issues.

For what concerns products present in the market, the company TransEn-

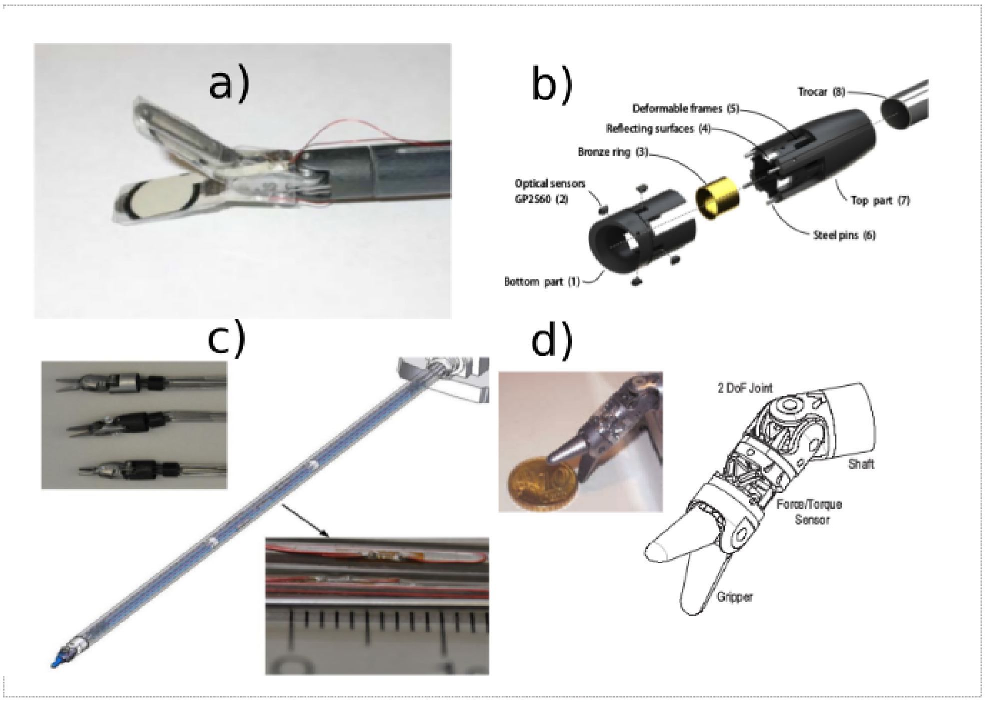


Figure 1.10: Force sensor integration in the da Vinci tools. a) Piezoresistive sensors placed at the tip [30]. b) Optical sensor to be integrated on the terminal part of the trocar [27]. c) Strain gauges placed on the cable shafts [29]. d) Completely re-designed tool equipped with force sensor, developed by the German Aerospace Center (DLR) [31].

terix, with their product Senhance, have recently proposed a teleoperated system for RMIS, similar to the da Vinci, including also haptic feedback making it the only device for RMIS today having such a feature [32]. It has obtained FDA clearance and CE marking but is today present in few facilities and it is struggling to compete with the more affirmed Da Vinci System.

The scope of this work is to exploit the dynamic model of a robotic arm to gain an estimation of the forces of interaction at its tip, without having to integrate an additional sensor. A review of the methods for dynamic modeling of a robotic arm is presented in Appendix B.

1.4 Research Problem

The goal of this thesis is to present and validate a sensorless, model-based method for the estimation of the forces and torques of interaction at the tip of the slave arm of the Da Vinci Research Kit. It is sensorless because the estimation of the forces and torques of interaction does not rely on any force/torque sensor, but it is carried out indirectly by means of the torques measured at the actuators. In fact, it is possible, by means of the control software already available for the dVRK [33], to retrieve the total torques exerted by the actuator to every joint of the robot. These torques are estimated from the currents in the actuators by means of a proper calibration already implemented in the dVRK control software [34]. These torques, indicated as τ_{tot} in Fig. 1.11, include both the joint torques due to the dynamics of the robot, which cause the robot's motion, and the joint torque resulting form external interaction (τ_{ext}). In fact, whenever an external force is applied to the tip of the arm, the motors must exert an additional torque so as to resist to this interaction and to reach the desired pose of the end effector.

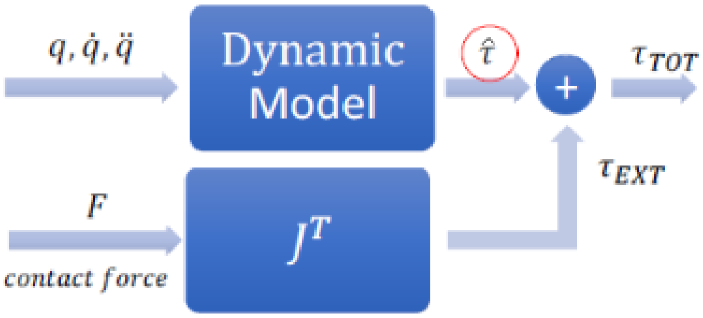


Figure 1.11: Scheme of the composition of torques acting on a robotic joint.

This torque must be separated from the total one in order to compute the forces and torques of interaction. By using the joint torques obtained from the measured motor currents (τ_{tot}) and by subtracting from them the torques resulting from the dynamics of the robot arm ($\hat{\tau}$), it is possible to obtain the joint torques only due to the external forces and torques acting on the tool.

Finally, the forces and torques of interaction are obtained through the inverse transpose of the Jacobian matrix. The accuracy of this method is assessed by comparing the estimated wrench to the one measured by a force/torque sensor (ATI mini 45). It is shown that the external wrench is well estimated compared to the measured one.

This work is organized as follows. First the Da Vinci Research Kit (DVRK), an open source robotic platform for research similar to the commercially available system, is presented. Then the kinematic and dynamic modeling of the DVRK is carried out. The parameters of the dynamic model are then presented and identified through an optimal excitation experiment. A section regarding the generation of the optimal excitation trajectory is also provided. The model, along with its identified parameters, is validated by assessing its capability of estimating joint torques during a task in free space. The model is finally exploited to estimated external forces acting on the end effector of the arm.

Chapter 2

Materials and Methods

In this section, the Da Vinci Research Kit robotic platform is presented and the kinematic and dynamic modeling of the slave arm is carried out. Then, the problem of the identification of the model's base parameters is explained in detail and finally the method to choose the optimal joint trajectory for the identification of the dynamic parameters is shown.

2.1 The Da Vinci Research Kit Platform

The Da Vinci Research Kit[1] is a telerobotic platform, consisting of parts from earlier generations of Da Vinci systems, provided by Intuitive Surgical. It is an open-source version of the commercially available Da Vinci Surgical System described in the previous sections, conceived to provide to researchers full access to the robot's hardware and software [34]. Since the commercialized Da Vinci Surgical System is a proprietary product, it provides limited

[1] Developed in partnership by: Johns Hopkins University, SMARTS Lab; Worcester Polytechnic Institute, AIM lab; Medical Motion Corporation; Intuitive Surgical, Inc. and Neuron Robotics, LLC

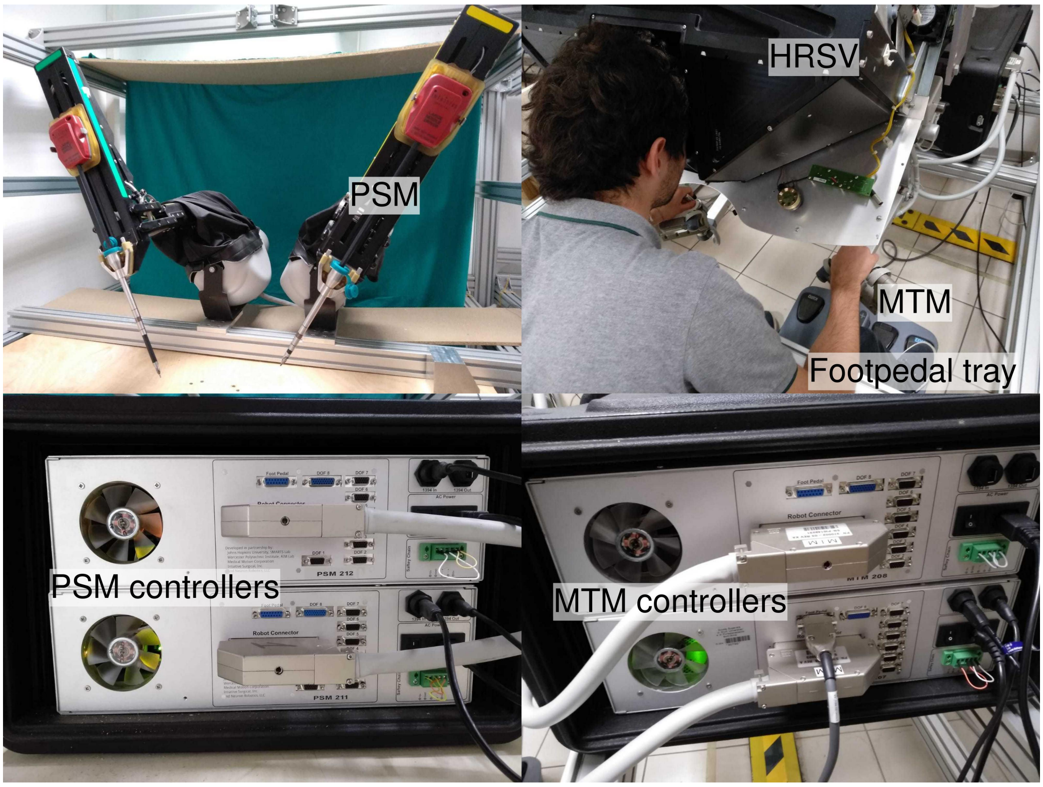


Figure 2.1: The da Vinci Research Kit platform available at The BioRobotics Institute comprising two patient side manipulators (PSM) and two master tool manipulators (MTM) with their respective controllers, a high resolution stereo viewer (HRSV) and a footpedal tray.

access to its software: as a consequence, the development of an open source telerobotic platform was of utmost importance for the flourishing of research activities on RMIS. Currently, the dVRK is used in 35 research groups around the world [35].

This research was carried out on the dVRK available at *The BioRobotics Institute* of *Scuola Superiore Sant'Anna*. The system used in this work comprises two patient side manipulators (PSM), two master tool manipulators (MTM), a high resolution stereo viewer (HRSV) with 640X480 resolution to be coupled to an endoscopic camera manipulator (ECM) which is currently to be integrated, and a footpedal tray (Fig. 2.1). Open source electronics for the PSM and MTM controllers were built and consist of an FPGA (with its firmware) coupled to a quad linear amplifier, which communicate to a Linux PC through a IEEE-1396 (firewire) bus [34], (Fig. 2.2). Open source software was also implemented based on *cisst* [36] and SAW (Surgical Assistant Workstation) packages interfacing with the Robot Operating System (ROS) (Fig. 2.3). This software was also integrated with different higher level programming environments, such as Matlab and Python [37]. Because the Python environment provides a higher frequency communication with the robot, with respect to Matlab scripts, in this thesis Python scripts were used to control the robot. Examples of such scripts will be shown in the Appendices.

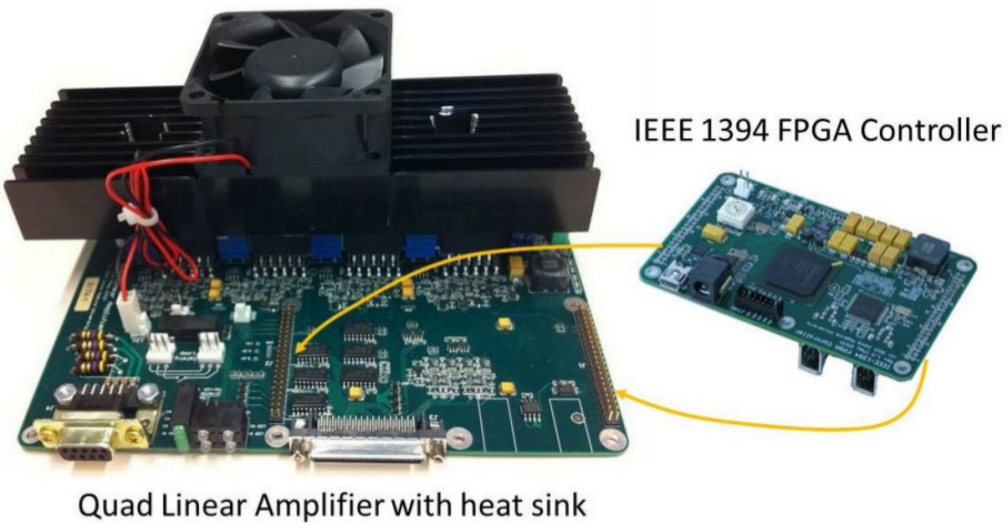
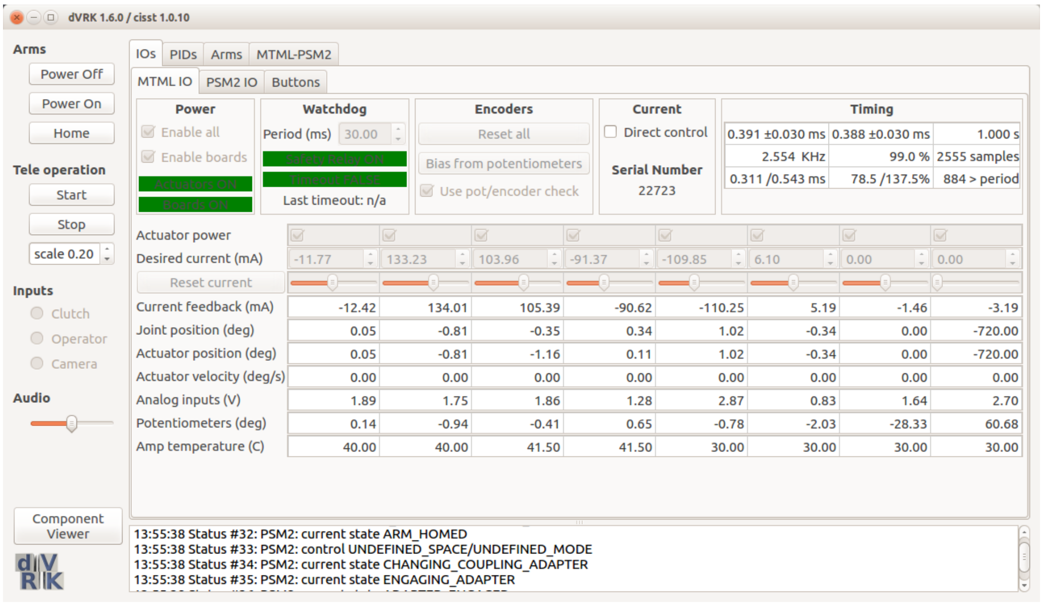

IEEE 1394 FPGA Controller
Quad Linear Amplifier with heat sink

Figure 2.2: The open source electronics built for the dVRK: an FPGA controller with IEEE-1394 communication bus coupled to a Quad Linear Amplifier with heat sink.



scale 0.20

☑ Enable all

☑ Enable boards

Actuators ON

Boards ON

Period (ms) 30.00

Safety Relay ON

Timeout FALSE

Last timeout: n/a

☑ Use pot/encoder check

□ Direct control

Serial Number
22723

0.391 ±0.030 ms	0.388 ±0.030 ms	1.000 s
2.554 KHz	99.0 %	2555 samples
0.311 /0.543 ms	78.5 /137.5%	884 > period

Actuator power	☑	☑	☑	☑	☑	☑	☑	☑
Desired current (mA)	-11.77	133.23	103.96	-91.37	-109.85	6.10	0.00	0.00
Current feedback (mA)	-12.42	134.01	105.39	-90.62	-110.25	5.19	-1.46	-3.19
Joint position (deg)	0.05	-0.81	-0.35	0.34	1.02	-0.34	0.00	-720.00
Actuator position (deg)	0.05	-0.81	-1.16	0.11	1.02	-0.34	0.00	-720.00
Actuator velocity (deg/s)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Analog inputs (V)	1.89	1.75	1.86	1.28	2.87	0.83	1.64	2.70
Potentiometers (deg)	0.14	-0.94	-0.41	0.65	-0.78	-2.03	-28.33	60.68
Amp temperature (C)	40.00	40.00	41.50	41.50	30.00	30.00	30.00	30.00

```
13:55:38 Status #32: PSM2: current state ARM_HOMED
13:55:38 Status #33: PSM2: control UNDEFINED_SPACE/UNDEFINED_MODE
13:55:38 Status #34: PSM2: current state CHANGING_COUPLING_ADAPTER
13:55:38 Status #35: PSM2: current state ENGAGING_ADAPTER
```

Figure 2.3: A screenshot of the ROS based control software for the DVRK.

2.2 Kinematic Modeling of the PSM Arm

The PSM is a 7 DOFs serial robot with five revolute (R) joints and one prismatic (P) joint in a RRPRRR configuration, which manipulates a tool around a remote center of motion (Fig. 2.4). The kinematic model is described by the modified Denavit-Hartenberg parameters given in Table 2.1, which are available in the dVRK User Guide [38]. The last degree of freedom, corresponding to the opening and closing of the gripper, was neglected in the modeling. These parameters will be needed afterwards to initialize the Newton Euler Algorithm so as to obtain the dynamic model of the arm. A schematic describing the PSM joints is shown in Fig. 2.5. It is worthwhile to note that the movements of the wrist joints are actuated by tendons which are in turn actuated by disks on the tool coupled to the sterile adapter 2.6. In particular the yaw motion of the wrist and also the gripping are produced by the coordinated movement of disks a and b, which move the two grippers.

Table 2.1: PSM modified DH parameters

#	Joint Name	Type	a	α	d	θ
1	Outer Yaw	R	0	$\pi/2$	0	$q_1 + \pi/2$
2	Outer Pitch	R	0	-$\pi/2$	0	$q_2 - \pi/2$
3	In/Out Ins.	P	0	$\pi/2$	$q_3 - 0.43$	0
4	Outer Roll	R	0	0	0.41	q_4
5	Wrist Pitch	R	0	-$\pi/2$	0	$q_5 - \pi/2$
6	Wrist Yaw	R	0.0091	-$\pi/2$	0	$q_6 - \pi/2$
7	End Effector	/	0	-$\pi/2$	0.01	0

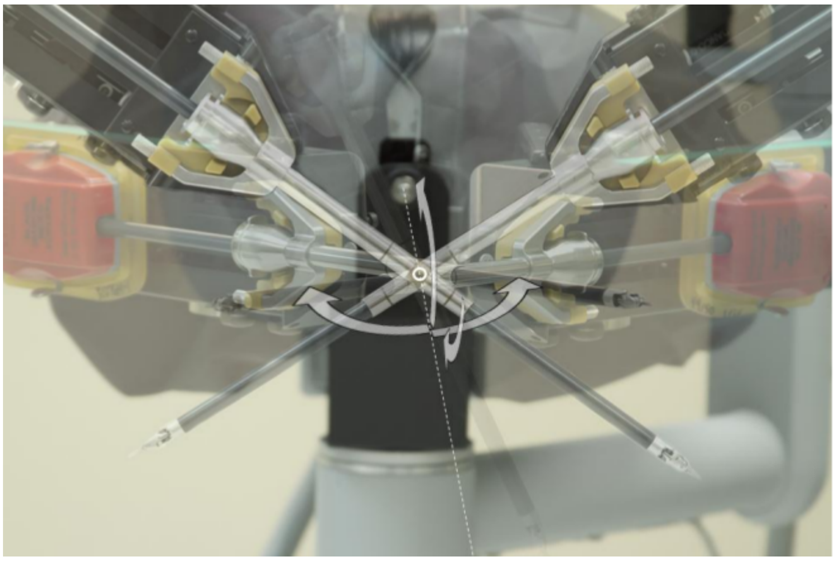


Figure 2.4: The remote center of motion (RCM) of the PSM arm.

2.3 Dynamic Modeling of The PSM Arm

The dynamic model of a robot provides the relationship between the actuator torques acting on the robot joints, and the resulting joint accelerations and joint motion [39]. The canonical expression of the PSM dynamic model expressed in joint space formulation is given by:

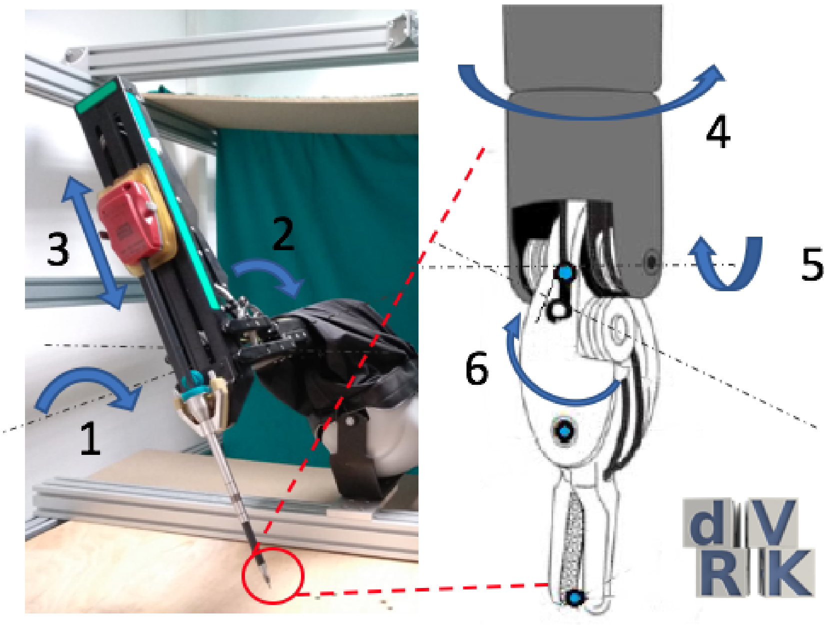


Figure 2.5: The Patient Side Manipulator arm and its joints. Joints 4, 1 and 2 allow respectively the outer roll, pitch and yaw of the instrument; joint 3 permits the in/out insertion of the tool, while joints 5 and 6 provide the pitch and yaw of the wrist.

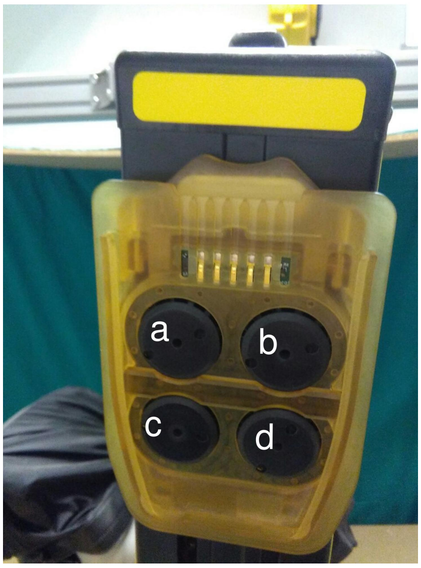


Figure 2.6: The disks on the sterile adapter that couple to the tool. The movement of disks a and b cause the wrist yaw motion and the gripping, disk c causes the wrist pitch, and disk d is responsible for the roll of the shaft.

$$\boldsymbol{\tau} = \boldsymbol{M(q)\ddot{q}} + \boldsymbol{C(q,\dot{q})} + \boldsymbol{G(q)} \tag{2.1}$$

where $\boldsymbol{\tau}$ is a (6×1) vector containing the joint actuator torques, $\boldsymbol{M}$ is the (6×6) inertia matrix, $\boldsymbol{C}$ is the (6×1) vector accounting for Coriolis and centrifugal effects, $\boldsymbol{G}$ is the (6×1) vector of gravitational forces and $\boldsymbol{q, \dot{q}, \ddot{q}}$ are respectively the (6×1) vectors of the joint angles, velocities and accelerations.

In the dynamic modeling of the PSM, several additional torque contributions can be considered, as in [40]. Firstly, an elastic contribution $\boldsymbol{\tau_e}$ was taken into account to model the elastic effects present in joints 1 and 2 due to power cables, and in joint 4 due to a torsional spring:

$$\boldsymbol{\tau_e} = \boldsymbol{K_e q} \tag{2.2}$$

with $\boldsymbol{K_e} = diag\{K_{e1}, K_{e2}, 0, K_{e4}, 0, 0\}$. No elastic effect was modeled on joints 3, 5 and 6. The static and viscous friction contributions were modeled as:

$$\boldsymbol{\tau_f} = \boldsymbol{F_v \dot{q}} + \mathbf{F_s} sgn(\dot{\mathbf{q}}) \tag{2.3}$$

where the matrices $\boldsymbol{F_v} = diag\{F_{v1}, ..., F_{v4}, \boldsymbol{F_{vl}}\}$, and $\boldsymbol{F_s} = diag\{F_{s1}, ..., F_{s6}\}$. The matrix $\boldsymbol{F_{vl}}$ is a 2 x 2 matrix modeling the viscous friction of the last 2 joints, that are coupled by a tendon driving mechanism. A constant additive torque $\boldsymbol{\tau_0}$ has been added to all joints in order to model the residual elastic force of the cables and any motor current offset.

A simplified drive inertia term $\boldsymbol{I_a}$, taking into account the effect of the transmission system without considering gyroscopic effects, was modeled as:

$$\boldsymbol{\tau_I} = \boldsymbol{I_a \ddot{q}} \tag{2.4}$$

where $\boldsymbol{I_a} = diag\{I_{a1}, ..., I_{a6}\}$.

Finally, since the mass of the last three links is negligible, they were neglected in the dynamic modeling and their dynamic parameters were set to zero. By summing all these contributions, the dynamic equations of the PSM arm are expressed by:

$$\begin{aligned} \boldsymbol{\tau} = \boldsymbol{M(q)\ddot{q}} + \boldsymbol{C(q,\dot{q})} + \boldsymbol{G(q)} \\ + \boldsymbol{K_e q} + \boldsymbol{F_v \dot{q}} + \mathbf{F_s} sgn(\dot{\mathrm{q}}) + \boldsymbol{\tau_0} + \boldsymbol{I_a \ddot{q}} \end{aligned} \tag{2.5}$$

A well known property of the dynamic model of a serial manipulator is its linearity with respect to a set of dynamic parameters, known as barycentric parameters [39]. By making explicit this linear relationship, it is possible to rewrite the equation of motion in (2.5) as follows:

$$\boldsymbol{\tau} = \boldsymbol{Y(q, \dot{q}, \ddot{q})\delta} \tag{2.6}$$

where $\boldsymbol{\delta} = \{\boldsymbol{\delta_1}^T, ..., \boldsymbol{\delta_6}^T\}^T$ is the vector of all the robots dynamic parameters. Matrix $\mathbf{Y}$ summarizes the whole dynamic model, and it depends only on the robot motion. By taking into account the previously modeled dynamics, we can write the dynamic parameters of each link as follows:

$$\begin{aligned}\boldsymbol{\delta_i} &= \{Lxx_i, Lyy_i, Lzz_i, Lxy_i, Lxz_i, Lyz_i, lx_i, ly_i, lz_i,\\ &\qquad m_i, K_{ei}, F_{vi}, F_{si}, I_{ai}, \tau_{0i}\}^T \text{ for links 1 and 2}\\ \boldsymbol{\delta_3} &= \{Lxx_3, Lyy_3, Lzz_3, Lxy_3, Lxz_3, Lyz_3, lx_3, ly_3, lz_3,\\ &\qquad m_3, F_{v3}, F_{s3}, I_{a3}, \tau_{03}\}^T \text{ for link 3}\\ \boldsymbol{\delta_4} &= \{lx_4, ly_4, lz_4, K_{e4}, F_{v4}, F_{s4}, I_{a4}, \tau_{04}\}^T \text{ for link 4}\\ \boldsymbol{\delta_5} &= \{lx_5, ly_5, lz_5, F_{v55}, F_{v56}, F_{s5}, I_{a5}, \tau_{05}\}^T \text{ for link 5}\\ \boldsymbol{\delta_6} &= \{lx_6, ly_6, lz_6, F_{v65}, F_{v66}, F_{s6}, I_{a6}, \tau_{06}\}^T \text{ for link 6}\end{aligned}$$

Therefore $\boldsymbol{\delta}$ groups 68 parameters in a column vector. L_i is the inertia tensor of the i-th link with respect to the center of mass, l_i is the first moment of inertia of the i-th link and m_i is the mass of the i-th link. However, in (2.6), some of the columns of matrix $\boldsymbol{Y}(6 \times 68)$ are always null or linearly dependent. Therefore, it is possible to transform (2.6) in a set of linearly independent equations [41], [42]:

$$\boldsymbol{\tau} = \boldsymbol{Y_b(q, \dot{q}, \ddot{q})\beta} \tag{2.7}$$

where $\boldsymbol{\beta}(48 \times 1)$ is the vector of the base parameters which are equal to a linear combination of the barycentric parameters and $\boldsymbol{Y_b}$ is a (6×48) matrix whose values depend only on the robot's motion, i. e. joint positions, velocities and accelerations.

The expressions of each element of $\boldsymbol{Y_b}$ and of all the base parameters were computed using *SymPyBotics*, a symbolic computing software that uses as input the robot DH parameters previously presented. The software is based on the Newton-Euler algorithm and was developed at University of Coimbra by Sousa and Cortesao [43], [44].

2.4 Parameters Identification

The dynamic identification is a problem of estimating the model base parameters $\boldsymbol{\beta}$ from recorded data during an excitation trajectory. These data include joint torques, joint positions, velocities and accelerations. The joint torques are computed by reading the currents in the joint actuators. Joint positions and velocities are given by encoders and acceleration is computed by derivation.

The data were collected online at a rate of 100 Hz during an optimal excitation trajectory. The computation of the optimal excitation trajectory is done offline and will be described in the next section. The robot joints are excited with the optimal trajectory by means of a Python script interfaced with ROS, as described in Appendix A. The collected data were filtered offline twice with a third order Butterworth low pass filter with 0.1 normalized cut-off frequency, inverting the time in the second filtering so as to eliminate the non linear phase shift.
Then, recalling (2.7), the data were organized in the following matrix equation:

$$\boldsymbol{\tau_M} = \boldsymbol{Y_M} \times \boldsymbol{\beta} \tag{2.8}$$

where $\boldsymbol{\tau_M}$ is equal to:

$$\boldsymbol{\tau_M} = \begin{bmatrix} \boldsymbol{\tau}(t_1) \\ \vdots \\ \boldsymbol{\tau}(t_M) \end{bmatrix} \tag{2.9}$$

Vector $\boldsymbol{\tau}(t_i)$ (6×1) is the vector of the joint torques estimated by means of the motor currents, for each time instant t_i $(i = 1 \ldots M$, where M is the number of acquired samples). The observation matrix $\boldsymbol{Y_M}$ was computed likewise by substituting the measured values of $\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)$, $\ddot{\boldsymbol{q}}(t)$ at each time iteration in the expressions of $\boldsymbol{Y_b}$ yielded in *SymPyBotics*, as:

$$\boldsymbol{Y_M} = \begin{bmatrix} \boldsymbol{Y_b}(\boldsymbol{q}(t_1), \dot{\boldsymbol{q}}(t_1), \ddot{\boldsymbol{q}}(t_1)) \\ \vdots \\ \boldsymbol{Y_b}(\boldsymbol{q}(t_M), \dot{\boldsymbol{q}}(t_M), \ddot{\boldsymbol{q}}(t_M)) \end{bmatrix} \tag{2.10}$$

Since the torque values are not at the same order of magnitude on each joint, a weighing matrix containing the inverse of the maximum torques recorded at each joint was introduced:

$$\boldsymbol{W} = diag\{1/max(\boldsymbol{\tau})_i, ..., 1/max(\boldsymbol{\tau})_n\} \tag{2.11}$$

Consequently, both sides of (2.8) were premultiplied offline by $\boldsymbol{W}$ as follows:

$$\begin{bmatrix} \boldsymbol{W} \times \boldsymbol{\tau}(t_1) \\ \vdots \\ \boldsymbol{W} \times \boldsymbol{\tau}(t_M) \end{bmatrix} = \begin{bmatrix} \boldsymbol{W} \times \boldsymbol{Y_b}(\boldsymbol{q}(t_1), \dot{\boldsymbol{q}}(t_1), \ddot{\boldsymbol{q}}(t_1)) \\ \vdots \\ \boldsymbol{W} \times \boldsymbol{Y_b}(\boldsymbol{q}(t_M), \dot{\boldsymbol{q}}(t_M), \ddot{\boldsymbol{q}}(t_M)) \end{bmatrix} \boldsymbol{\beta} \tag{2.12}$$

The approach chosen to identify the optimal base parameters vector $\boldsymbol{\beta^*}$ is the general least squares approach based on the left pseudo inverse of the observation matrix $\mathbf{Y_M}$:

$$\boldsymbol{\beta^*} = (\boldsymbol{Y_M}^T \boldsymbol{Y_M})^{-1} \boldsymbol{Y_M}^T \times \boldsymbol{\tau_M} \tag{2.13}$$

The identified parameters were computed offline and are given in Table 2.2. Note that not all identified parameters are physically consistent: for instance the positive definiteness of the inertia matrix, which is a property of physically existing inertia matrices, was not respected. This is due to the non-consideration of the parameters feasibility, since the minimization of the

Table 2.2: Identified PSM dynamic base parameters

fv_{56}	0.0087
fv_{65}	0.015
K_{e1}	2.5958
K_{e2}	0.6219
K_{e4}	0.0022
m_3	0.0115
lx_1	0.2298
$l1_y + l2_z$	-0.0069
f_{v1}	0.1090
f_{c1}	0.1363
τ_{o1}	-0.3259
lx_2	-0.0013
ly_2	-0.2103
f_{v2}	0.2940

f_{c2}	0.1896
τ_{o2}	-0.0107
lx_3	-0.0093
ly_3	-0.0247
$lz_3 + lz_4$	-0.2299
I_{a3}	0.0132
f_{v3}	-13.5792
f_{c3}	1.7079
τ_{o3}	0.4847
lx_4	7.74e-4
$ly_4 + ly_5$	-2.77e-4
I_{a4}	-9.86e-5
f_{v4}	0.0015
f_{c4}	0.0029

τ_{o4}	-0.0052
lx_5	2.9e-5
ly_5	5.83e-4
I_{a5}	-0.0025
f_{v5}	0.0271
f_{c5}	0.0126
lx_6	2.59e-4
ly_6	1.52e-4
lz_6	-0.0014
I_{a6}	-0.0028
f_{v6}	0.014
f_{c6}	0.004
τ_{o6}	0.0079

$I_{a1} + Lzz_1 + Lyy_2 + Lzz_3$	0.0201
$Lxx_2 - Lyy_2 + Lxx_3 - Lzz_3 + 0.8324 * lz_4$	0.0385
$Lxy_2 - Lxz_3$	-0.0352
$Lxz_2 - Lxy_3$	-0.0072
$Lyz_2 - Lyz_3$	0.0012
$I_{a2} + Lzz_2 + Lyy_3 + 0.8324 * lz_4$	-6.32e-4

torque prediction error was considered as the only optimization criterion. It has been shown that the physical inconsistency of the dynamic parameters gives rise to problems in control [43]. However, works in literature have proved that torque prediction without accounting for the consistency of the dynamic parameters is possible [45].

2.5 Generation of The Optimal Excitation Trajectory

In order to identify the dynamic parameters, the excitation of the robot in its joint space is required. The excitation trajectory should be sufficiently rich in harmonics so as to obtain an accurate estimation of the dynamic parameters. However unmodeled dynamics, such as link elasticity, should not be excited. The optimal excitation trajectory is chosen among a class of trajectories which are described by the following equation [46], [47]:

$$q_i(t) = \sum_{l=1}^{L} \frac{a_l^i}{\omega_f l} \sin(\omega_f l t) - \frac{b_l^i}{\omega_f l} \cos(\omega_f l t) + q_0^i. \tag{2.14}$$

The chosen trajectory is thus a finite harmonic series with L harmonics of the fundamental frequency $\omega_f/2\pi$. The number of harmonics L limits the frequency content of the excitation trajectory, so as to not excite unmodeled dynamics. In this study $L = 5$ and $\omega_f/2\pi = 0.1$ Hz were imposed, as in [40]. The quantities a_l^i, b_l^i and q_0^i were tuned so as to minimize the conditioning number of $\boldsymbol{Y_M} \times \boldsymbol{P}$, where $\boldsymbol{P}$ is a (68×68) weighting matrix of the form:

$$\boldsymbol{P} = diag(\frac{1}{||Y_{M,1}||}, \cdots, \frac{1}{||Y_{M,p}||}) \tag{2.15}$$

In (2.15), $||Y_{M,i}||$ is the norm of the i-th column of Y_M [40]. This minimization strategy was chosen because the conditioning number of $\boldsymbol{Y_M} \times \boldsymbol{P}$ is a measure of the sensitivity of the least squares solution $\boldsymbol{\beta^*}$ to perturbations in $\boldsymbol{Y_M}$ and in the measured torques $\boldsymbol{\tau_M}$. The minimization was computed using the Active-Set algorithm [48] with constraints on maximum and minimum joint positions ($\boldsymbol{q_{max}}$, $\boldsymbol{q_{min}}$) and on maximum and minimum joint velocities ($\dot{\boldsymbol{q}}_{\boldsymbol{max}}$, $\dot{\boldsymbol{q}}_{\boldsymbol{max}}$):

Table 2.3: PSM joint limits

	J1	J2	J3	J4	J5	J6
q_{min} [deg - m]	-60	-45	0.05	-180	-90	-90
q_{max} [deg - m]	60	45	0.18	180	90	90
$\dot{q}_{min}$ [rad/s - m/s]	-2	-2	-0.4	-6	-5	-5
$\dot{q}_{max}$ [rad/s - m/s]	2	2	0.4	6	5	5

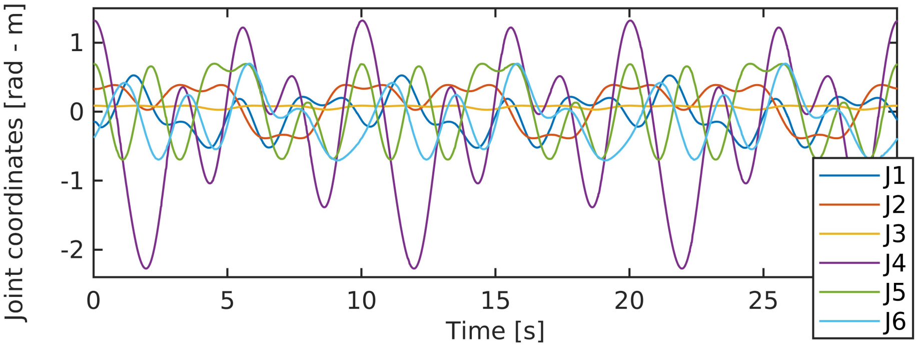


Figure 2.7: Optimal Excitation Trajectories of the PSM arm joints.

$$\begin{aligned} \boldsymbol{q_{min}} < q(t) < \boldsymbol{q_{max}} \\ \boldsymbol{\dot{q}_{min}} < q(t) < \boldsymbol{\dot{q}_{max}} \end{aligned} \tag{2.16}$$

The imposed constraints on joint position and velocity are summarized in Table 2.3. The resulting PSM optimal joint trajectories are shown in Fig. 2.7.

Chapter 3

Results and Discussion

In this section, the validation of the proposed dynamic model with the identified parameters is shown and an application of the model is proposed. The validation is performed by assessing the capability of the model to estimate the torques exerted by the joint actuators during a motion of the robot in free space. Conversely, the proposed model application is to exploit the dynamic model to estimated forces and torques exerted at the tip of the slave arm (interaction wrench). This wrench, in a surgical scenario, could represent the forces and torques exchanged between the slave arm and the patient tissue.

3.1 Free-space Manipulation - Model Validation

Ideally, in free-space manipulation, the torques estimated with the dynamic model are identical to the ones estimated by means of the motor currents, which are given by the ROS-Python interface and are used for validation. Consequently, the validation of the proposed model is performed by evaluating the ability of the model to estimate joint torques during free space motion.

The validation data were acquired using two different protocols. In the first protocol, a test trajectory in joint space was imposed automatically using the DVRK control software. This trajectory was chosen among the same class of trajectories described by (2.14), but different from the trajectory used for identification (reported in Fig. 2.7). Conversely, during the second validation protocol an operator tele-manipulated the PSM from the master console. During both tests the data for validation were recorded. These data include, joint torques from motor currents, joint positions and velocities from encoders. All these signals were acquired using the open-source controller described in [34], and were filtered twice with a third order Butterworth LPF, inverting the time in the second filtering so as to obtain zero-lag (normalized cutoff frequency 0.1). Accelerations were computed by derivation of velocities.

The torques estimated by means of the dynamic model are given by:

$$\hat{\boldsymbol{\tau}} = \boldsymbol{Y_b}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\beta}^* \tag{3.1}$$

Where $\boldsymbol{\beta}^*$ is the vector of the identified base parameters and $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ are the values of the robot motion which are substituted in $\boldsymbol{Y_b}$. The estimated torques in both tests were compared with the actual joint torques, estimate by means of the the motor currents. Results are shown in Fig. 3.1 for the automatic trajectory and in Fig. 3.2 for the teleoperation.

A numerical and undimensional evaluation metric of the model performance is provided by exploiting the normalized root mean square deviation for each joint:

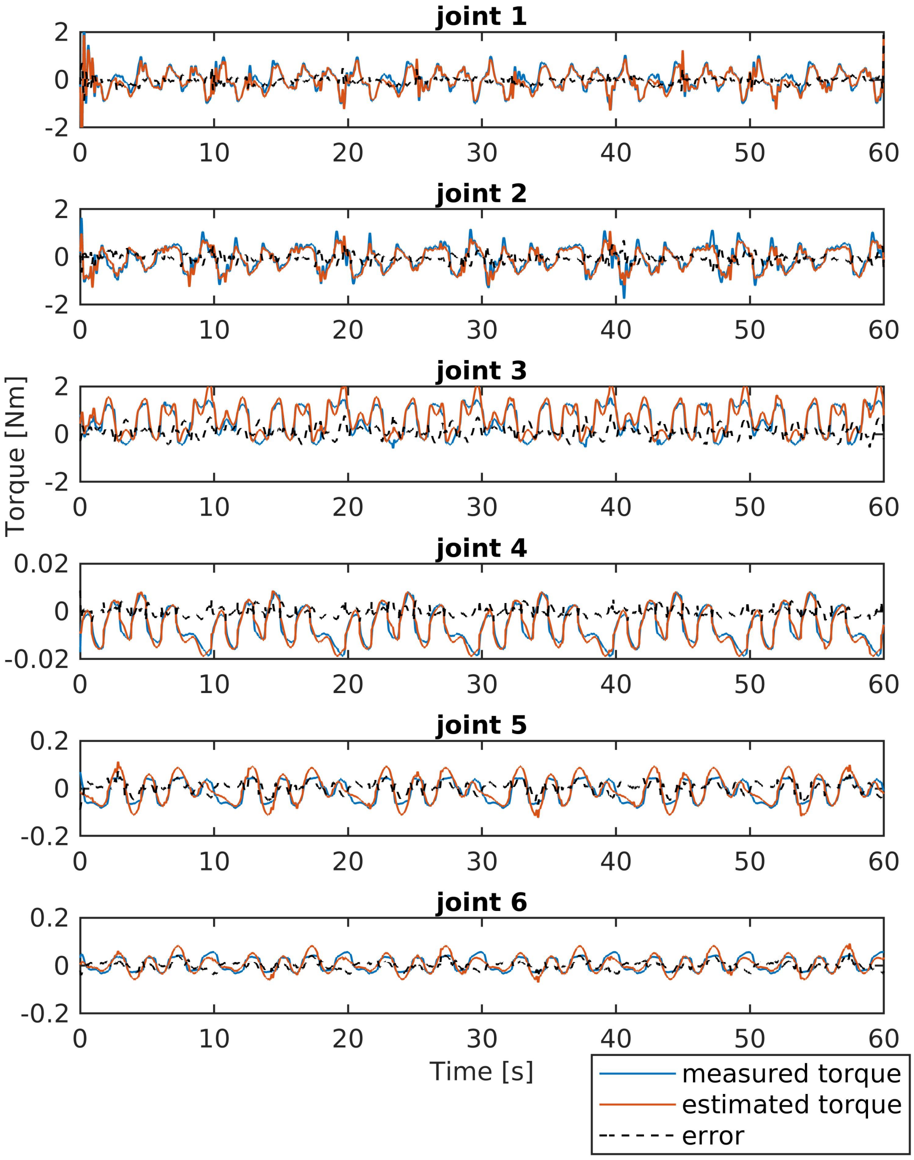


Figure 3.1: Estimated and measured joint torques in an automatic trajectory tracking.

$$NRMSD_i = \frac{\sqrt{\frac{1}{N}\sum_{n=1}^{N}[\hat{\tau}(t_n) - \tau(t_n)]_i^2}}{(\tau_{max} - \tau_{min})_i} \tag{3.2}$$

where $\hat{\tau}(t_n)_i$ is the torque computed with the model at time t_n for the

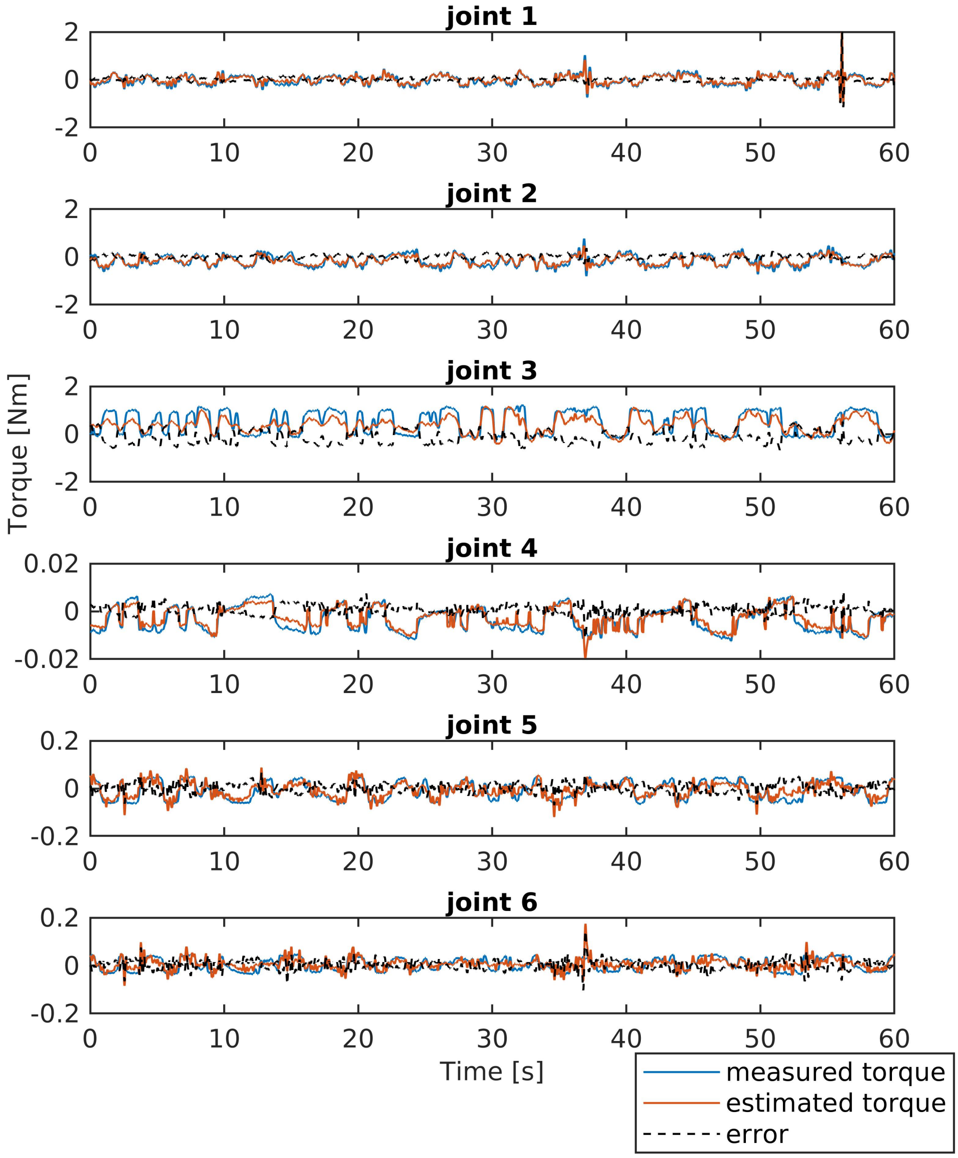


Figure 3.2: Estimated and measured joint torques in a teleoperation scenario.

i^{th} joint. Likewise, $\tau(t_n)_i$ is the torque estimated from the motor currents at time t_n for the i^{th} joint, and N is the number of samples. The performance of the model in terms of dynamic joint torque estimation in free-space is shown in Tables 3.1 and 3.2. A better precision is achieved on every joint during

Table 3.1: PSM torque NRMSD in a teleoperation scenario

joints	1	2	3	4	5	6
NRMSD(%)	5.92	5.78	18.84	10.41	16.84	22.96

Table 3.2: PSM torque NRMSD in an automatic trajectory tracking scenario

joints	1	2	3	4	5	6
NRMSD(%)	2.38	4.80	12.83	6.49	15.29	18.11

a test trajectory (i. e. with no operator) rather than in teleoperation. This is because the test trajectory is similar to the excitation trajectory used for identification, meaning that it follows the same parametric equation but with different parameters. For the test trajectory, the best performance is achieved on joint 1 (2.38 % NRMSD) while the worst performance is achieved on the last joint (18.11 % NRMSD). Instead, the teleoperation trajectory shows best performance on joint 2 (5.78 % NRMSD) while the worst performance is again on the last joint (22.96 % NRMSD). The evaluation of these estimated torques using the same criterion as [40] gives similar accuracy.

3.2 External Wrench Estimation - Model Application

In the presence of an external interaction wrench at the tip of the PSM, the total torques at each joint are given by:

$$\begin{aligned} \boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{G}\boldsymbol{q} \\ +\boldsymbol{K_e}\boldsymbol{q} + \boldsymbol{F_v}\dot{\boldsymbol{q}} + \mathbf{F_s}sgn(\dot{\mathbf{q}}) + \boldsymbol{\tau_0} + \boldsymbol{I_a}\ddot{\boldsymbol{q}} + \boldsymbol{\tau_{ext}} \end{aligned} \tag{3.3}$$

Where $\boldsymbol{\tau_{ext}}$ is the effect at the joint level of the external forces and torques. These external forces and torques in the surgical scenario could represent the interaction between the surgical tool and the patient. Equation 3.3 can be rewritten as:

$$\boldsymbol{\tau} = \boldsymbol{Y_b(q, \dot{q}, \ddot{q})\beta^*} + \boldsymbol{\tau_{ext}} \tag{3.4}$$

therefore, recalling Eq. 3.1, it is possible to obtain the torques due to external interaction by:

$$\boldsymbol{\tau_{ext}} = \boldsymbol{\tau} - \boldsymbol{\hat{\tau}} \tag{3.5}$$

The wrench $\boldsymbol{w}$ at the end effector was estimated, as in [39], by:

$$\boldsymbol{w} = (J^T)^+ * \boldsymbol{\tau_{ext}} \tag{3.6}$$

where $(J^T)^+$ is the Moore-Penrose inverse of the transpose of the robot Jacobian. The wrench at the end effector estimated with (3.6) was compared with the forces and torques sensed by an ATI mini 45 force/torque sensor, previously calibrated by using a look-up table in LabVIEW. The experimental setup is shown in Fig. 3.3. An operator teleoperates the PSM so as to make its end effector interact with a pin attached to the force sensor fixed to ground. At the same time, the data from the ATI mini 45 is acquired through LabVIEW (Fig. 3.4), and $\boldsymbol{q}, \boldsymbol{\dot{q}}, \boldsymbol{\ddot{q}}$ and $\boldsymbol{\tau}$ are acquired via ROS-Python. In the experiment aiming at measuring torques, they were exerted at an equal distance from the sensor reference frame and the robot's reference frame, which is on the system's RCM (the z axes of both frames are coaxial). This allows obtaining consistent torque measures. The wrench estimated with (3.6) was computed with respect to the robot reference frame by using the spatial Ja-

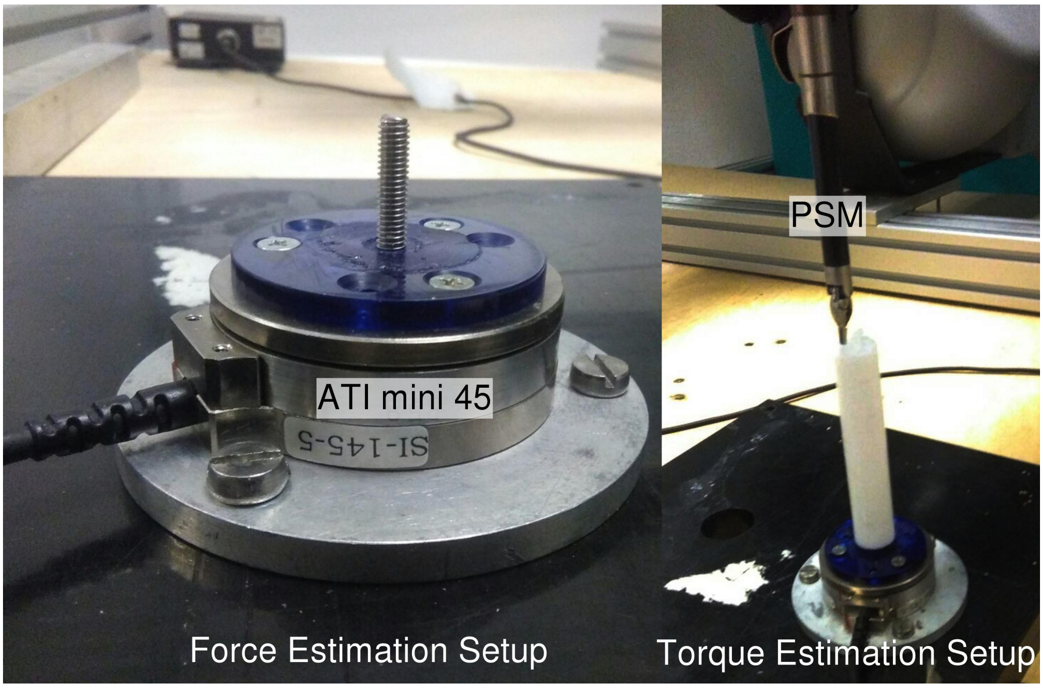


Figure 3.3: Interaction force and torque sensing using an ATI mini 45 force/torque sensor.

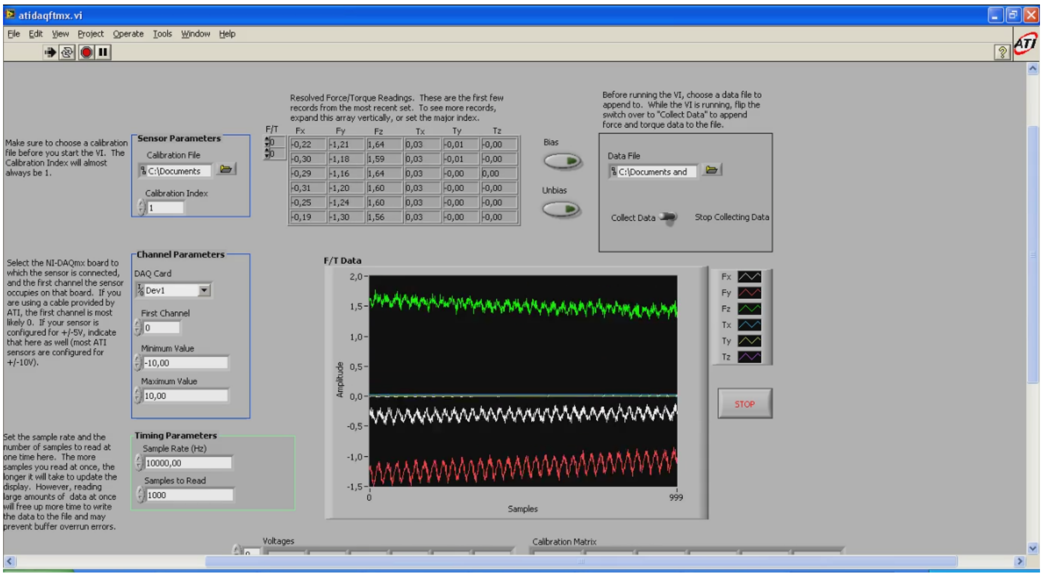
cobian matrix. In Fig. 3.5 a comparison between the estimated wrench and the wrench measured with the sensor is shown.

The normalized root mean square deviation of the forces and torques for each axis can be calculated and evaluated as an index of the prediction accuracy, and therefore as a performance measure of the method:

$$NRMSD = \frac{\sqrt{\frac{1}{N}\sum_{n=1}^{N}[\hat{F}(t_n) - F(t_n)]_i^2}}{(F_{max} - F_{min})_i} \tag{3.7}$$

$$NRMSD = \frac{\sqrt{\frac{1}{N}\sum_{n=1}^{N}[\hat{T}(t_n) - T(t_n)]_i^2}}{(T_{max} - T_{min})_i} \tag{3.8}$$

Where $\hat{F}$ and $\hat{T}$ are the external forces/torque estimated with the model, while F and T are the external forces/torques measured with the ATI sensor. The NRMSD is presented in Table 3.3. External torques and forces acting at



Make sure to choose a calibration file before you start the VI. The Calibration Index will almost always be 1.

Sensor Parameters

Calibration File: C:\Documents

Calibration Index: 1

Resolved Force/Torque Readings. These are the first few records from the most recent set. To see more records, expand this array vertically, or set the major index.

F/T: 0, 0

Fx	Fy	Fz	Tx	Ty	Tz
-0,22	-1,21	1,64	0,03	-0,01	-0,00
-0,30	-1,18	1,59	0,03	-0,01	-0,00
-0,29	-1,16	1,64	0,03	-0,00	0,00
-0,31	-1,20	1,60	0,03	-0,00	-0,00
-0,25	-1,24	1,60	0,03	-0,00	-0,00
-0,19	-1,30	1,56	0,03	-0,00	-0,00

Before running the VI, choose a data file to append to. While the VI is running, flip the switch over to "Collect Data" to append force and torque data to the file.

Data File: C:\Documents and

Collect Data / Stop Collecting Data

Select the NI-DAQmx board to which the sensor is connected, and the first channel the sensor occupies on that board. If you are using a cable provided by ATI, the first channel is most likely 0. If your sensor is configured for +/-5V, indicate that here as well (most ATI sensors are configured for +/-10V).

Channel Parameters

DAQ Card: Dev1

First Channel: 0

Minimum Value: -10,00

Maximum Value: 10,00

Set the sample rate and the number of samples to read at one time here. The more samples you read at once, the longer it will take to update the display. However, reading large amounts of data at once will free up more time to write the data to the file and may prevent buffer overrun errors.

Timing Parameters

Sample Rate (Hz): 10000,00

Samples to Read: 1000

Figure 3.4: A screenshot of the acquisition of forces and torques from the ATI mini 45 sensor through LabVIEW.

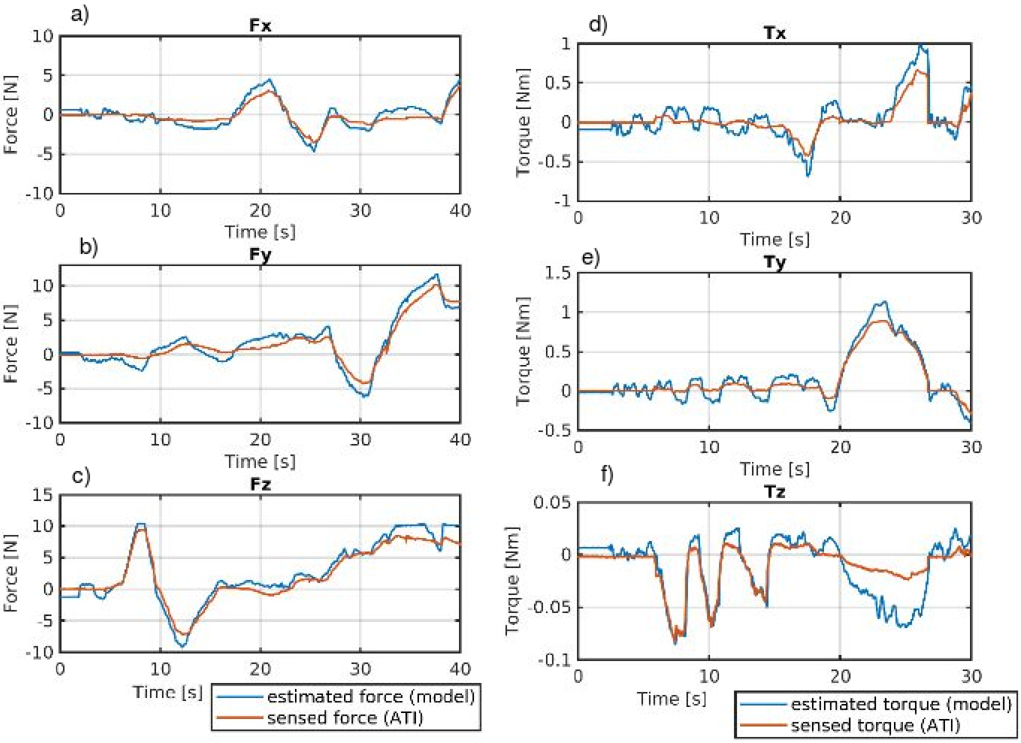


Figure 3.5: Comparison of measured and estimated external forces/torques.

Table 3.3: NRMSD of the force and torque estimation

	Fx	Fy	Fz	Tx	Ty	Tz
NRMSD (%)	8.26	5.96	6.10	8.13	5.71	15.50

the end effector are well estimated except for some jumps due to the sudden loss of contact between the gripper and the pin. In some cases the torque around the Z axis is not transmitted faithfully during the torsion of the pin since it was hard to grip well the pin with the tool. This may have given rise to less accurate sensed torque (Fig. 3.5f).

The only work found in the literature which can bring a comparison to this results is [49]. Here, the performance in the estimation of the forces achieved is higher (2.29% for Fx, 3.19% for Fy, 3.35 % for Fz). However, only a small range of external forces have been exerted compared to the ones typically used in surgical applications, which can be as high as 5 N for palpation and 7 N for suturing [50] [20]). Also, in [49], the external torques of interactions were not considered in the model evaluation.

In this thesis instead, the exerted forces were in the range used in typical surgical applications, proving that this method could be used in such scenarios. Note that this comparison between the measured forces/torques and the estimated ones is highly dependent on the used filter. In fact, all measured data - wrench obtained from force sensor, position obtained from encoders and used for the estimation - are noisy.

Chapter 4

Conclusion

This thesis was developed in collaboration with the Biorobotics Institute of Scuola Superiore Sant'anna in Pontedera. The aim of this thesis is the dynamic modeling of the patient side manipulator (PSM) of the da Vinci Research Kit (dVRK) in order to better estimate the external interaction wrench exerted on the tip of the robotic tool. The modeling was done by exploiting a Python package available online which implements the Newton Euler Algorithm and computes the dynamic model of the robot arm starting from its DH parameters [43]. The identification of the model's parameters was done by means of optimal excitation trajectories. The excitation of the robot is achieved via Python scripts which interface with the Robot Operating System. The data from the robot such as joint positions, joint velocities and joint torques was acquired with the same method. The dynamic model was validated by evaluating its torque estimation ability at the joint level and its contact wrench estimation ability at the end effector. The model is able to estimate the joint dynamic torques during a free space task with an accuracy that is comparable to the one present in the literature [40]. It is also shown that the model can be used to estimate the interaction wrench, in the range of forces of typical robotic procedures [50], [20]. Limitations of the model include the fact that it doesn't consider backlash of the tendons that drive the actuation of the joints, which is difficult to model. Also, the links are considered as infinitely rigid whereas some link elasticity is always present. Future work may focus on the implementation of these additional features in the proposed dynamic model. The python package in [43] used in this work to compute the dynamic model of the PSM could be easily used to model any other serial robot. Moreover, the method of robot control and data acquisition using python scripts interfacing with ROS can be used for any other project concerning the dVRK. Finally, this work provides only an estimation of the interaction wrench at the tip of the end effector. The obtained wrench estimation could be used as input for the reflection of the wrench to the MTM arm in a force reflection teleoperation scenario.

Bibliography

[1] R. H. Taylor, A. Menciassi, G. Fichtinger, P. Fiorini, and P. Dario, "Medical robotics and computer-integrated surgery," in *Springer handbook of robotics*, pp. 1657–1684, Springer, 2016.

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

[3] G. Deacon, A. Harwood, J. Holdback, D. Maiwand, M. Pearce, I. Reid, M. Street, and J. Taylor, "The pathfinder image-guided surgical robot," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 224, no. 5, pp. 691–713, 2010.

[4] G. R. Sutherland, S. Wolfsberger, S. Lama, and K. Zarei-nia, "The evolution of neuroarm," *Neurosurgery*, vol. 72, no. suppl_1, pp. A27–A32, 2013.

[5] J. R. Joseph, B. W. Smith, X. Liu, and P. Park, "Current applications of robotics in spine surgery: a systematic review of the literature," *Neurosurgical focus*, vol. 42, no. 5, p. E2, 2017.

[6] R. D. Howe and Y. Matsuoka, “Robotics for surgery,” *Annual review of biomedical engineering*, vol. 1, no. 1, pp. 211–240, 1999.

[7] W. L. Bargar, A. Bauer, and M. Börner, “Primary and revision total hip replacement using the robodoc (r) system.,” *Clinical Orthopaedics and Related Research (1976-2007)*, vol. 354, pp. 82–91, 1998.

[8] A. Takács, D. Á. Nagy, I. Rudas, and T. Haidegger, “Origins of surgical robotics: From space to the operating room,” *Acta Polytechnica Hungarica*, vol. 13, no. 1, pp. 13–30, 2016.

[9] R. Goertz, J. Burnett, and F. Bevilacqua, “Servos for remote manipulation,” tech. rep., Argonne National Lab. Argonne, IL (US), 1953.

[10] L. Cragg and H. Hu, “Application of mobile agents to robust teleoperation of internet robots in nuclear decommissioning,” in *Industrial Technology, 2003 IEEE International Conference on*, vol. 2, pp. 1214–1219, IEEE, 2003.

[11] N. C. Mitsou, S. V. Velanas, and C. S. Tzafestas, “Visuo-haptic interface for teleoperation of mobile robot exploration tasks,” in *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, pp. 157–163, IEEE, 2006.

[12] A. D. Alexander, “Impacts of telemation on modern society,” in *On Theory and Practice of Robots and Manipulators*, pp. 121–136, Springer, 1972.

[13] S. DiMaio, M. Hanuschik, and U. Kreaden, “The da vinci surgical system,” in *Surgical Robotics*, pp. 199–217, Springer, 2011.

[14] S. Lang, S. Mattheis, P. Hasskamp, G. Lawson, C. Güldner, M. Mandapathil, P. Schuler, T. Hoffmann, M. Scheithauer, and M. Remacle, "A european multicenter study evaluating the flex robotic system in transoral robotic surgery," *The Laryngoscope*, vol. 127, no. 2, pp. 391–395, 2017.

[15] G. S. Guthart and J. K. Salisbury, "The intuitive/sup tm/telesurgery system: overview and application," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1, pp. 618–621, IEEE, 2000.

[16] J. Marescaux and F. Rubino, "The zeus robotic system: experimental and clinical applications," *Surgical Clinics*, vol. 83, no. 6, pp. 1305–1315, 2003.

[17] A. M. Okamura, "Methods for haptic feedback in teleoperated robot-assisted surgery," *Industrial Robot: An International Journal*, vol. 31, no. 6, pp. 499–508, 2004.

[18] M. A. Srinivasan and C. Basdogan, "Haptics in virtual environments: Taxonomy, research status, and challenges," *Computers & Graphics*, vol. 21, no. 4, pp. 393–404, 1997.

[19] N. Enayati, E. De Momi, and G. Ferrigno, "Haptics in robot-assisted surgery: Challenges and benefits," *IEEE reviews in biomedical engineering*, vol. 9, pp. 49–65, 2016.

[20] M. Kitagawa, A. M. Okamura, B. T. Bethea, V. L. Gott, and W. A. Baumgartner, "Analysis of suture manipulation forces for teleoperation with force feedback," in *International Conference on Medical Image*

Computing and Computer-Assisted Intervention, pp. 155–162, Springer, 2002.

[21] C. E. Reiley, T. Akinbiyi, D. Burschka, D. C. Chang, A. M. Okamura, and D. D. Yuh, "Effects of visual force feedback on robot-assisted surgical task performance," *The Journal of thoracic and cardiovascular surgery*, vol. 135, no. 1, pp. 196–202, 2008.

[22] M. Mahvash, J. Gwilliam, R. Agarwal, B. Vagvolgyi, L.-M. Su, D. D. Yuh, and A. M. Okamura, "Force-feedback surgical teleoperator: Controller design and palpation experiments," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, International Symposium on (HAPTICS)*, (Reno, NE), pp. 465–471, IEEE, Mar. 2008.

[23] C. R. Wagner and R. D. Howe, "Force feedback benefit depends on experience in multiple degree of freedom robotic surgery task," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1235–1240, 2007.

[24] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE transactions on robotics and automation*, vol. 9, no. 5, pp. 624–637, 1993.

[25] A. M. Okamura, L. N. Verner, T. Yamamoto, J. C. Gwilliam, and P. G. Griffiths, "Force feedback and sensory substitution for robot-assisted surgery," in *Surgical Robotics*, pp. 419–448, Springer, 2011.

[26] O. Mohareri, C. Schneider, and S. Salcudean, "Bimanual telerobotic surgery with asymmetric force feedback: a davinci® surgical system implementation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4272–4277, 2014.

[27] G. A. Fontanelli, L. R. Buonocore, F. Ficuciello, L. Villani, and B. Siciliano, “A novel force sensing integrated into the trocar for minimally invasive robotic surgery,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 131–136, 2017.

[28] M. Tavakoli, R. V. Patel, and M. Moallem, “A force reflective master-slave system for minimally invasive surgery,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 4, pp. 3077–3082, IEEE, 2003.

[29] A. Talasaz, A. L. Trejos, and R. V. Patel, “Effect of force feedback on performance of robotics-assisted suturing,” in *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pp. 823–828, IEEE, 2012.

[30] M. O. Culjat, C.-H. King, M. L. Franco, C. E. Lewis, J. W. Bisley, E. P. Dutson, and W. S. Grundfest, “A tactile feedback system for robotic surgery,” in *Engineering in Medicine and Biology Society (EMBS)*, pp. 1930–1934, IEEE, 2008.

[31] U. Seibold, B. Kubler, and G. Hirzinger, “Prototype of instrument for minimally invasive surgery with 6-axis force sensing capability,” in *International Conference on Robotics and Automation (ICRA)*, pp. 496–501, IEEE, 2005.

[32] S. G. Alletti, C. Rossitto, S. Cianci, E. Perrone, S. Pizzacalla, G. Monterossi, G. Vizzielli, S. Gidaro, and G. Scambia, “The senhance™ surgical robotic system (“senhance”) for total hysterectomy in obese patients: a pilot study,” *Journal of robotic surgery*, vol. 12, no. 2, pp. 229–234, 2018.

[33] "davinci research kit ros stack." Available at: `"https://github.com/jhu-dvrk/dvrk-ros"`, 2018.

[34] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da vinci surgical system," in *International Conference on Robotics and Automation (ICRA)*, (Hong Kong, China), pp. 6434–6439, 2014.

[35] "The da vinci research kit wiki." Available: `"https://rsearch.intusurg.com/index.php/Main_Page"`, 2018.

[36] A. Kapoor, A. Deguet, and P. Kazanzides, "Software components and frameworks for medical robot control," in *IEEE International Conference on Robotics and Automation,ICRA*, pp. 3813–3818, 2006.

[37] Z. Chen, A. Deguet, R. H. Taylor, and P. Kazanzides, "Software architecture of the da vinci research kit," in *Robotic Computing (IRC), IEEE International Conference on*, pp. 180–187, IEEE, 2017.

[38] "User guide: The da vinci reaserch kit," Intuitive Surgical, Inc., 2014.

[39] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, "Robotics: modelling, planning and control," Springer, 2009.

[40] G. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano, "Modelling and identification of the da vinci research kit robotic arms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

[41] M. Gautier, "Numerical calculation of the base inertial parameters of robots," *Journal of Field Robotics*, vol. 8, no. 4, pp. 485–506, 1991.

[42] M. Gautier and W. Khalil, "Direct calculation of minimum set of inertial parameters of serial robots," *IEEE Transactions on robotics and Automation*, vol. 6, no. 3, pp. 368–373, 1990.

[43] C. D. Sousa and R. Cortesão, "Physical feasibility of robot base inertial parameter identification: A linear matrix inequality approach," *The International Journal of Robotics Research*, vol. 33, no. 6, pp. 931–944, 2014.

[44] C. D. Sousa and R. Cortesao, "Sagerobotics: open source framework for symbolic computation of robot models," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pp. 262–267, ACM, 2012.

[45] G. Calafiore, M. Indri, and B. Bona, "Robot dynamic calibration: Optimal excitation trajectories and experimental parameter estimation," *Journal of robotic systems*, vol. 18, no. 2, pp. 55–68, 2001.

[46] J. Swevers, C. Ganseman, J. De Schutter, and H. Van Brussel, "Experimental robot identification using optimised periodic trajectories," *Mechanical Systems and Signal Processing*, vol. 10, no. 5, pp. 561–577, 1996.

[47] J. Swevers, C. Ganseman, D. B. Tukel, J. De Schutter, and H. Van Brussel, "Optimal robot excitation and identification," *IEEE transactions on robotics and automation*, vol. 13, no. 5, pp. 730–740, 1997.

[48] P. E. Gill, W. Murray, M. H. Wright, *et al.*, "Numerical linear algebra and optimization," vol. 1, Addison-Wesley Redwood City, CA, 1991.

[49] H. Sang, J. Yun, R. Monfaredi, E. Wilson, H. Fooladi, and K. Cleary, "External force estimation and implementation in robotically assisted

minimally invasive surgery," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 13, no. 2, p. e1824, 2017.

[50] A. L. Trejos, J. Jayender, M. Perri, M. D. Naish, R. V. Patel, and R. Malthaner, "Robot-assisted tactile sensing for minimally invasive tumor localization," *The International Journal of Robotics Research*, vol. 28, no. 9, pp. 1118–1133, 2009.

Appendix A

Python and Matlab Scripts

A.1 Robot Excitation and Data Collection

Here follows the Python script used in the robot excitation experiment. While the robot is excited in its joints according to the previously computed optimal excitation trajectory, the joint positions, velocities and torques are acquired. The data are used later offline to perform the identification of the dynamic parameters.

```
# import the necessary libraries:

import rospy         # provides Python to ROS interface
import dvrk          # dVRK software based on cisst/SAW packages
import numpy          # numerical calculus package
import scipy.io  # to save and open .mat files for offline analysis

# load the previously computed optimal trajectory:

data = scipy.io.loadmat('Excitation_trajectory.mat')
```

```
q = data['traj_opt']

# define the slave arm to control:

slave1 = dvrk.psm('PSM1')

# initialise position velocity and torque matrices:

current_joint_position = numpy.zeros(q.shape)
current_joint_velocity = numpy.zeros(q.shape)
current_joint_effort = numpy.zeros(q.shape)

# define frequency of reading and writing at 100 Hz:

rate = 100
r = rospy.Rate(rate)

# move arm to initial position before acquiring data:

slave1.move_joint(q[0])

# excite the robot and simultaneously collect data at 100 Hz

for i in range(len(q)):
    slave1.move_joint(q[i], interpolate = False)
    current_joint_position[i] = slave1.get_current_joint_position()
    current_joint_velocity[i] = slave1.get_current_joint_velocity()
    current_joint_effort[i] = slave1.get_current_joint_effort()
    r.sleep() # pause so as to have the selected frequency

# save data in a .mat file

Output = {'q' : joint_pos , 'dq' : joint_vel, 'tau' : joint_eff}
scipy.io.savemat('Output_ident.mat',Output)
```

A.2 Force Comparison Experiment

This code was executed during the experiments with the force sensor. While the PSM was manipulating, the data relative to joint positions, velocities, torques and the robot Jacobian were acquired, so as to compute offline the estimation of the external wrench.

```
    # import the necessary libraries:

import rospy
import dvrk
import numpy
import scipy.io

# define the robot arm

slave1 = dvrk.psm('PSM1')

# define the frequency and the duration of the experiment
```

```
rate = 100
secs = 90
samples = rate*secs
r = rospy.Rate(rate)

# define position, velocity, torque and Jacobian matrices

joint_position = numpy.zeros([samples, 7])
joint_velocity = numpy.zeros([samples, 7])
joint_effort = numpy.zeros([samples, 7])
Jacob_body = numpy.zeros([samples, 6, 6 ])
Jacob_space = numpy.zeros([samples, 6, 6 ])

# acquire data (while manipulating)

for i in range(samples):
    print('OK!')
    joint_position[i] = slave1.get_current_joint_position()
    joint_velocity[i] = slave1.get_current_joint_velocity()
    joint_effort[i] = slave1.get_current_joint_effort()
    Jacob_body[i] = slave1.get_jacobian_body()
    Jacob_space[i] = slave1.get_jacobian_spatial()
    r.sleep()
print('done!')

# save data in a .mat file
```

```
OutData1 = {'Q' : joint_pos , 'DQ' : joint_vel, 'Tau' : joint_effort}
OutData2 = {'Jbody' : Jacob_body, 'J_space' : Jacob_spatial}

scipy.io.savemat('F_Joints.mat', OutData1)
scipy.io.savemat('F_Jacob.mat', OutData2)
```

Appendix B

Dynamic Modeling

The development of the dynamic model of a robot manipulator plays an important role in the design of control algorithms, in the simulation of the robot motion, and in the design of robot prototypes [39]. The objective of dynamic modeling is to provide the relationship between forces and torques ($\boldsymbol{\tau}$) acting on the robot joints and the motion of the joints ($\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$).

One method of computing the Dynamic model is through the Lagrange formulation which comprises the differentiation of the system Lagrangian. The Lagrangian (L) of a system is the difference between the system's kinetic (E_k) and potential (E_p) energy:

$$L = E_k - E_p \tag{B.1}$$

The Lagrange formulation of the dynamic model relates the Lagrangian to the joint torques ($\boldsymbol{\tau}$):

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial L}{\partial \mathbf{q}} = \boldsymbol{\tau} \tag{B.2}$$

where q and $\dot{q}$ denote the robot joint positions and velocities. The computation of the system kinetic and potential energy for different types of robotic systems can be found in more detailed robotics textbooks [39]. Another well known method to compute a robot dynamic model is the Newton Euler Algorithm (NEA). Differently from the Lagrange formulation of the dynamic model, the NEA is a recursive method, meaning that it entails a forward recursion and a backward recursion. The forward recursion is relative to the propagation of velocities and accelerations from base of the manipulator to tip, while the backward recursion regards the propagation of forces and torques in the structure, from the end effector to the base. As the name suggests, it is based on the dynamic equilibrium equations, that is the Newton equation for linear equilibrium, and the Euler equations for rotational equilibrium. Considering a generic link i let $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ be the joint positions velocities and accelerations, let $\boldsymbol{\mu}$ and $\boldsymbol{f}$ be the torques and forces exchanged at the boundaries of each link. $\boldsymbol{R}_i^{i-1}$ is the rotation matrix from frame i-1 to frame i and finally $\boldsymbol{\tau}_i$ is the torque acting on joint i. The following equations describe the Newton Euler algorithm:

$$\boldsymbol{\omega}_i = (\boldsymbol{R}_i^{i-1})^T(\boldsymbol{\omega}_{i-1} + \dot{q}_i \boldsymbol{z}_0) \tag{B.3}$$

$$\dot{\boldsymbol{\omega}}_i = (\boldsymbol{R}_i^{i-1})^T(\dot{\boldsymbol{\omega}}_{i-1} + \ddot{q}_i \boldsymbol{z}_0 + \dot{q}_i \boldsymbol{\omega}_{i-1} \times \boldsymbol{z_0}) \tag{B.4}$$

$$\boldsymbol{a}_i = (\boldsymbol{R}_i^{i-1})^T \boldsymbol{a}_{i-1} + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{r}_{i-1,i} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i}) \tag{B.5}$$

$$\boldsymbol{a}_{Ci} = \boldsymbol{a}_i + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{r}_{i,Ci} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{r}_{i,Ci}) \tag{B.6}$$

$$\boldsymbol{f}_i = \boldsymbol{R}_{i+1}^i \boldsymbol{f}_{i+1} + m_i \boldsymbol{a}_{Ci} \tag{B.7}$$

$$\boldsymbol{\mu}_i = -\boldsymbol{f}_i \times (\boldsymbol{r}_{i-1,i} + \boldsymbol{r}_{i,Ci}) + \boldsymbol{R}^i_{i+1}\boldsymbol{\mu}_i + 1 + \boldsymbol{R}^i_{i+1}\boldsymbol{f}_{i+1} \times \boldsymbol{r}_{i,Ci} + \boldsymbol{I}_i\dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\boldsymbol{I}_i\boldsymbol{\omega}_i) \tag{B.8}$$

$$\tau_i = \boldsymbol{\mu}_i^T(\boldsymbol{R}_i^{i-1})^T\boldsymbol{z}_0 + F_{vi}\dot{\boldsymbol{q}}_i + F_{si}sgn(\dot{\boldsymbol{q}}_i) \tag{B.9}$$

where: $\boldsymbol{\omega}_i$ and $\dot{\boldsymbol{\omega}}_i$ are the angular velocities and accelerations of the links, $\boldsymbol{z}_0$ is the axis of rotation of the base frame, $\boldsymbol{r}_{i-1,i}$ is the distance between the reference frame of link i-1 and link i (according to the Denavit-Hartenberg convention), $\boldsymbol{r}_{i,Ci}$ is the distance between the center of mass of the link i and the reference frame of link i, m_i and $\boldsymbol{I}_i$ are respectively the mass and the inertia tensor of the i-th link, $\boldsymbol{a}_i$ is the acceleration of the origin of reference frame i and $\boldsymbol{a}_{C_i}$ is the acceleration of the center of mass of link i. The workflow of the algorithm is schematized in schematized in Fig. (B.1). For given joint positions, velocities and acceleration the recursive algorithm is carried out in the following two phases: 1) The forward recursion phase, where, given the initial conditions $\boldsymbol{\omega}_0$, $\boldsymbol{a}_0 - \boldsymbol{g}$ and $\dot{\boldsymbol{\omega}}_0$ (which are respectively the angular velocity, the linear acceleration and the angulare acceleration of the base link) equations (B.3), (B.4), (B.5), and (B.6) are used for $i = 1 \dots n$ to compute $\boldsymbol{\omega}_i$, $\dot{\boldsymbol{\omega}}_i$, $\boldsymbol{a}_i$ and $\boldsymbol{a}_{C_i}$. 2) The backward recursion phase, where, with known terminal conditions $\boldsymbol{f}_{n+1}$ and $\boldsymbol{\mu}_{n+1}$, equations (B.7) and (B.8) are used for $i = 1 \dots n$ to compute $\boldsymbol{f}_i$, $\boldsymbol{\mu}_i$ and equation (B.9) to compute τ_i.

Since the NEA provides a computationally efficient algorithm, it has been used throughout this work.

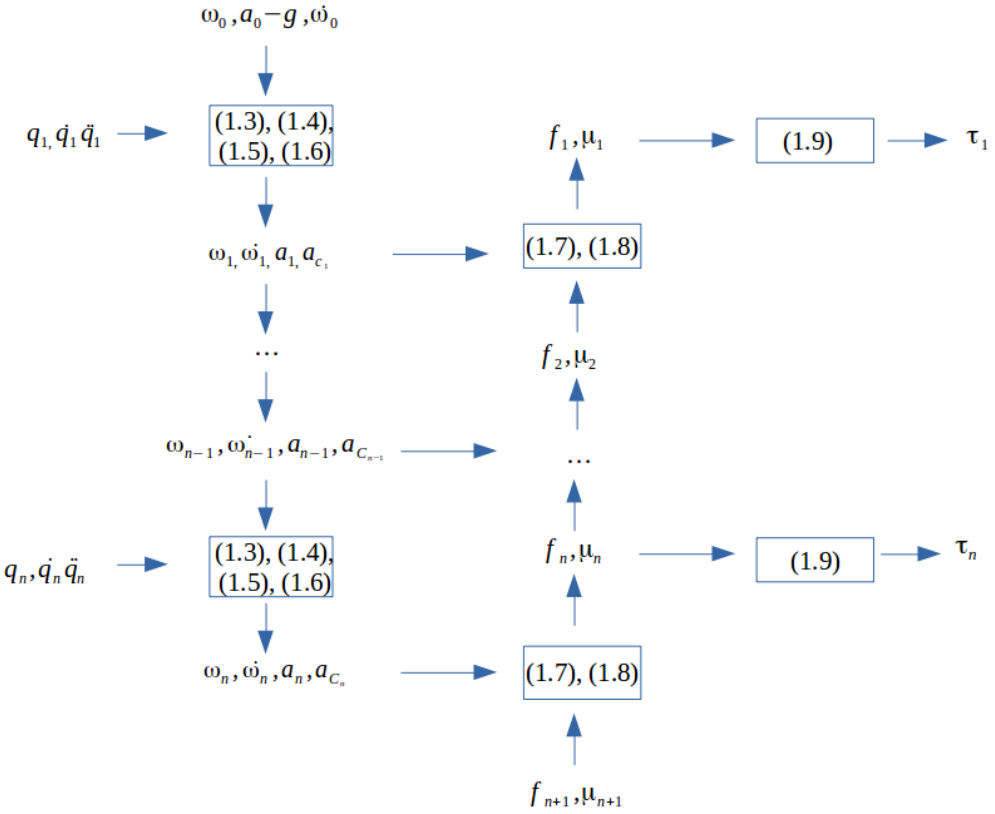

$\omega_0, a_0 - g, \dot{\omega}_0$
$q_1, \dot{q}_1 \ddot{q}_1$
(1.3), (1.4), (1.5), (1.6)
$\omega_1, \dot{\omega}_1, a_1, a_{c_1}$
...
$\omega_{n-1}, \dot{\omega}_{n-1}, a_{n-1}, a_{C_{n-1}}$
$q_n, \dot{q}_n \ddot{q}_n$
(1.3), (1.4), (1.5), (1.6)
$\omega_n, \dot{\omega}_n, a_n, a_{C_n}$
f_1, μ_1
(1.9)
τ_1
(1.7), (1.8)
f_2, μ_2
...
f_n, μ_n
(1.9)
τ_n
(1.7), (1.8)
f_{n+1}, μ_{n+1}

Figure B.1: Schematic of the Newton Euler Alogrithm.