DEPARTMENT OF
SYSTEMS ENGINEERING AND AUTOMATION
UNIVERSIDAD DE MÁLAGA

PhD Thesis

SMART NAVIGATION IN SURGICAL ROBOTICS

Carlos Jesús Pérez del Pulgar Mancebo

DEPARTMENT OF
SYSTEMS ENGINEERING AND AUTOMATION
UNIVERSIDAD DE MÁLAGA

PhD THESIS

SMART NAVIGATION IN SURGICAL ROBOTICS

Author: D. Carlos Jesús Pérez del Pulgar Mancebo
Supervisor: Prof. Mr. Víctor F. Muñoz Martínez
Co-supervisor: PhD. Ms. Isabel García Morales

MÁLAGA, November 9th, 2015

D. Víctor Fernando Muñoz Martínez, Catedrático de Universidad, y Dña. Isabel García Morales, Profesora Contratada Doctora, ambos adscritos al Departamento de Ingeniería de Sistemas y Automática de la Universidad de Málaga, hacen constar

QUE

Ambos son directores de la tesis titulada "SMART NAVIGATION IN SURGICAL ROBOTICS", realizada por D. Carlos Jesús Pérez del Pulgar Mancebo, y que dicho autor ha llevado a cabo los objetivos de investigación propuestos para la completa elaboración de la Tesis, y que se encuentra en disposición de la defensa de la misma ante el Tribunal Examinador.

Málaga, 9 de Noviembre de 2015

Fdo. Víctor F. Muñoz Martínez   Fdo. Isabel García Morales

Dedicado a mi esposa Eva y mi abuelo José

Dedicated to my wife Eva and my grandfather José

Agradecimientos - Acknowledgements

Este trabajo de investigación no ha sido un camino en solitario, por esto, quiero agradecer el apoyo que he recibido por parte de tantos amigos y compañeros.

En primer lugar me gustaría agradecer el apoyo de mi familia y amigos, principalmente el de mi esposa Eva y mis padres Fernando e Isabel. A mi esposa por su apoyo incondicional y a mis padres por los sabios consejos que siempre me han dado.

A mis compañeros del Departamento de Ingeniería de Sistemas y Automática por haber sido pacientes conmigo durante estos últimos años y apoyarme. Además, me gustaría agradecer a los estudiantes de la Escuela de Ingenierías que han colaborado conmigo. En especial a Rubén Gómez, Juan José Velasco, Pedro Cervera e Ignacio Alzugaray.

Additionally, I would like to thank the Telerobotics and Haptic Lab (ESA-ESTEC) team for accepting me as a team member. In particular, I would like to thank André for supervising my research stay and Jan for supporting my research work. Furthermore, I would like to thank the European Space Agency staff that I met during my stay in the Netherlands. I will never forget my time in ESTEC.

Esta estancia de investigación no la podría haber realizado sin la ayuda de viejos y nuevos amigos a los que estoy especialmente agradecido. En concreto, muchas gracias Carolina por animarme a realizar la estancia en la ESA, sin tus ánimos estoy seguro de que no la hubiera realizado. Álvaro, no sabes cómo te agradezco todo lo que me has ayudado el tiempo que estuve allí. Y como no, gracias a David por enseñarme esos sitios tan maravillosos que hay en Leiden y preparar esas tortillas españolas que tan bien nos saben. Nunca me hubiera imaginado encontrar allí a personas tan maravillosas como vosotros.

Por último, me gustaría agradecer especialmente a mis directores de tesis. A Isabel por ser una gran compañera y amiga, y por ayudarme en la elaboración de esta memoria y la preparación de los artículos de investigación. Víctor, muchas gracias por inculcarme el pensamiento crítico y científico, tan importante para que mi carrera investigadora no se quede aquí, sino que continúe.

Resumen

La cirugía mínimamente invasiva, y concretamente la cirugía laparoscópica, ha supuesto un gran cambio en la forma de realizar intervenciones quirúrgicas en el abdomen. Actualmente, la cirugía laparoscópica ha evolucionado hacia otras técnicas aún menos invasivas, como es la cirugía de un solo puerto, en inglés *Single Port Access Surgery*. Esta técnica consiste en realizar una única incisión, por la que son introducidos los instrumentos y la cámara laparoscópica a través de un único trocar multipuerto. La principal ventaja de esta técnica es una reducción de la estancia hospitalaria por parte del paciente, y los resultados estéticos, ya que el trocar se suele introducir por el ombligo, quedando la cicatriz oculta en él. Sin embargo, el hecho de que los instrumentos estén introducidos a través del mismo trocar hace la intervención más complicada para el cirujano, que necesita unas habilidades específicas para este tipo de intervenciones.

Esta tesis trata el problema de la navegación de instrumentos quirúrgicos mediante plataformas robóticas teleoperadas en cirugía de un solo puerto. En concreto, se propone un método de navegación que dispone de un centro de rotación remoto virtual, el cuál coincide con el punto de inserción de los instrumentos (punto de fulcro). Para estimar este punto se han empleado las fuerzas ejercidas por el abdomen en los instrumentos quirúrgicos, las cuales han sido medidas por sensores de esfuerzos colocados en la base de los instrumentos. Debido a que estos instrumentos también interaccionan con tejido blando dentro del abdomen, lo cual distorsionaría la estimación del punto de inserción, es necesario un método que permita detectar esta circunstancia. Para solucionar esto, se ha empleado un detector de interacción con tejido basado en modelos ocultos de Markov el cuál se ha entrenado para detectar cuatro gestos genéricos. Por otro lado, en esta tesis se plantea el uso de guiado háptico para mejorar la experiencia del cirujano cuando utiliza plataformas robóticas teleoperadas. En concreto, se propone la técnica de aprendizaje por demostración (*Learning from Demonstration*) para generar fuerzas que puedan guiar al cirujano durante la resolución de tareas específicas.

El método de navegación propuesto se ha implantado en la plataforma quirúrgica CISOBOT, desarrollada por la Universidad de Málaga. Los resultados experimentales obtenidos validan tanto el método de navegación propuesto, como el detector de interacción con tejido blando. Por otro lado, se ha realizado un estudio preliminar del sistema de guiado háptico. En concreto, se ha empleado una tarea genérica, la inserción de una clavija, para realizar los experimentos necesarios que permitan demostrar que el método propuesto es válido para resolver esta tarea y otras similares.

Abstract

Laparoscopic surgery has been considered a breakthrough for performing abdominal surgeries. Now, surgery has evolved towards less invasive techniques, such as Single Port Access Surgery, which is on the cutting edge of this type of technique. This procedure involves only one incision, through which all surgical instruments as well as a laparoscopic camera are introduced using a multiport trocar. The primary advantage of this technique is a reduction in the hospital stay and better cosmetic results, i.e., only one scar is needed, which is hidden in the umbilicus. However, because the instruments are introduced through a single trocar, the surgery becomes more difficult for surgeons, who need specific skills to perform this type of surgery. Thus, teleoperated surgical robotic platforms have emerged as a solution.

This thesis is focused on the navigation of surgical instruments by teleoperated surgical robotic platforms in Single Port Access Surgery. In particular, the proposed navigation method is based on a virtual Remote Centre of Motion, which coincides with the insertion point (fulcrum point) that is estimated using abdominal interaction forces along the surgical instruments. Because these instruments also interact with the soft tissue inside the abdomen, which affects the fulcrum point estimation, a method is needed to determine whether the instrument tip interacts with the soft tissue inside the abdomen. To this end, we have used a soft tissue interaction detector based on a Hidden Markov Model. Furthermore, this thesis proposes the use of haptic guidance to improve the surgeon's experience when using teleoperated robotic platforms. Thus, Learning from Demonstration is proposed to generate guidance force references that guide the surgeon during the reproduction of a task.

The proposed navigation method has been implemented in the CISOBOT surgical platform, which was developed by the Universidad de Málaga. The obtained experimental results validate the proposed navigation method and the soft tissue interaction detector. Additionally, a preliminary study of the haptic guidance system has been performed. Specifically, a generic and complex task, i.e., peg-in-hole insertion, has been used to perform the necessary experiments that demonstrate that the proposed methodology can solve this task, among others.

Content

List of Figures

List of Tables

1 Introduction

1.1 Smart Navigation in Surgical Robotics

Over the last two decades, surgical procedures have evolved from laparotomies (open surgery) to minimally invasive surgeries following the hypothesis that reducing the number and size of incisions leads to a reduction in both patient recovery time and the likelihood of postoperative complications. Laparoscopic surgery involves the use of specialized instruments and a camera introduced through the abdominal wall via, at least, three small incisions that allow the surgeon to manipulate the inner organs. However, current trends are leading towards the development of different approaches. One new method is known as single port access surgery (SPAS) (Gomes, 2011), a laparoscopic surgical procedure in which a single incision is performed, and all surgical instruments and a camera are inserted by a multiport trocar. In this way, reducing the number of incisions in the abdominal wall provides some benefits, such as better cosmetic results, reduced postoperative pain and shorter hospital stay (Halim, 2008) (Kahnamoui et al., 2007). Despite these advantages, this technique has some drawbacks for surgeons because the instruments are inserted through the same trocar and can collide inside and outside of the abdomen, i.e., “sword fighting”. Moreover, the close proximity of the instruments to the endoscope entails a loss of triangulation and a reduction in the field of view (Shussman et al., 2011). This fact implies constrained movements of the instruments, which requires more skill of the surgeons. Thus, teleoperated robotic platforms are a useful tool for surgeons. In these platforms, a smart navigation system can be integrated within the robotic platform to improve surgeon skill using a guidance system that assists surgeons during the teleoperation. Furthermore, the smart navigation system may be able to recognize the gesture that is performed, move the instruments while considering the SPAS constraints, and correctly guide the surgeon during the reproduction of the recognized gesture. These features require a previous knowledge of the surgical procedure. So, a method that is able to learn this knowledge would be needed.

Therefore, different technological fields can be extended and integrated to achieve the proposed smart navigation system for SPAS as follows:

Navigation

As stated previously, the inherent constraints of SPAS may be considered when manipulators are handling the instruments. The most important constraint is the fulcrum point, which is the point at which the surgical instruments are inserted. It is commonly determined using a Remote Centre of Motion (RCM), which coincides with the fulcrum point. The RCM reduces the number of degrees of freedom to four: three rotations and one longitudinal displacement along the instrument. The da Vinci, as well as other surgical robotic systems use a mechanical RCM based on a dedicated kinematic design that is adjusted before the surgery, and it remains fixed during the operation (Chin-Hsing et al., 2012). This solution is primarily used in clinical applications owing to its robustness and safety, i.e., a fault in the robot controller would not damage the patient. However, this solution requires specific, large and complex kinematic structures that are not useful for SPAS because of their volume. For example, the da Vinci system uses four arms that are attached to the main structure by a parallelogram RCM kinematic structure. Although it has been used for SPAS, it has several limitations due to its volume, i.e., a reduction in the range of motion inside the abdomen and collision of the arms during the movements (Kroh et al., 2011). Thus, lightweight robotic surgical systems based on generic kinematic structures use a virtual or software RCM, where the fulcrum point is estimated online and can be changed during surgery. This solution is useful when the entry point is moving, as the abdominal wall does during breathing. Nevertheless, when a virtual RCM is used, the fulcrum point location has to be known. Thus, the use of an online estimator based on multi-axial Force-Torque (F/T) sensors placed between the end effector and the surgical instrument appears to be a solution. The primary advantage of this method is that the same sensor would be used for different purposes, i.e., bilateral teleoperation, fulcrum point estimation and/or force control. In fact, different authors ((Krupa et al., 2004), (Cortesao et al., 2006) and (Ruiz Morales and Correcher, 2012)) performed an extensive study of the use of F/T sensors to estimate the fulcrum point using different control strategies. However, none of these contributions considers the interaction of the instrument with the soft tissue inside the abdomen during the motion of the surgical tools. This is an important fact to be considered because it leads to errors in the fulcrum point estimation and, consequently, to undesirable forces on the patient's abdomen. Furthermore, none of these studies has been adapted for use in SPAS, where several instruments are inserted by the same trocar, and redundant information would be used to improve the estimation of the fulcrum point.

Human-Robot Interaction

In teleoperation, a surgeon performs a task by remotely controlling a robotic platform. For efficient operation, surgeons need to receive rich sensory information from the robots. Thus, the platforms can be classified as follows:

- Direct teleoperation: this is the basic teleoperation system. The surgeon provides movements by handling a master device during the surgery, and the robot (slave) follows its movements in real time. Generally, the surgeon operates the robot using a console that contains the master devices that are used to move the manipulators and screens to provide visual information. This type of teleoperation has evolved towards telesurgery, where a surgeon can perform surgery even though he is not physically in the same location. This technique involves not only a master console and slave robots but also communication and information technologies.

- Bilateral teleoperation: this is an evolution of direct teleoperation. In this case, the surgeon feels the forces that are exerted on the instruments during the surgery. Thus, force sensors are placed on the slave robots, and haptic devices are used to teleoperate them and provide force feedback to the surgeon. The force sensors can be classified depending on their proximity to the instrument tip and sensing capabilities. The first type is a multi-axial force-torque sensor placed on a robot's wrist (Krupa et al., 2004), i.e., between the end effector and the instrument. The sensor can feel the exerted forces and torques throughout the instrument and do not need to be sterilized because it is not introduced into the abdomen. However, they do not provide enough sensitivity to differentiate between different types of tissue and different force interactions through the instrument, i.e., forces exerted on the abdominal wall during movement could affect the measured forces. Conversely, the second type of sensor has been designed to be installed on the instrument tip in order to feel the soft tissue inside the abdomen and, thus, to enable the surgeon to feel the tissue in the same way as in a laparotomy (Berkelman, 2003). However, there are two primary disadvantages to this type of sensor: sterilization is needed because the sensor is inserted into the abdomen, and temperature changes affect the force measurements when transducers are used (Puangmali et al., 2008).

As described, this kind of teleoperation allows reproduction of only the surgeon's hand movements using the instruments. However, a method that provides additional support during the teleoperation would be advantageous. Haptic guidance is an evolution of bilateral teleoperation. It is used to provide guidance forces to limit the workspace or to assist the surgeon during the execution of a predefined trajectory to solve a task. This technique is commonly known as *virtual fixtures* (Abbott et al., 2007). It improves the accuracy and safety of surgery and reduces the time needed for surgical tasks because it combines the precision of a robotic system and the intelligence of humans (Casals, Basomba et al., 2011). The use of virtual fixtures is based on the premise that the reference trajectories and forbidden regions are previously known, i.e., relies on an accurate reference position that is typically obtained from images or marks (Park et al., 2001; Abbott et al., 2007). However, there are tasks in which this position can be affected by estimation errors or cannot be estimated. Thus, other methods to generate haptic guidance are needed.

Machine Learning

Machine learning has been widely used in robotics. It primarily consists of using sensory information to retrieve movements from multiple demonstrations, which are encoded into probabilistic models. Then, these models are used to recognize these movements or to provide a previously trained trajectory to a robot.

In the case of movement recognition, machine learning has been used to detect and classify a surgeon's movements to identify the stage of the surgical procedure during its execution (Lin et al., 2005). Moreover, it has been used to perform a robot-human collaboration, where a manipulator is able to assist the surgeon depending on the surgical procedure stage (Bauzano et al., 2014). However, the use of machine learning algorithms to detect the interaction of instruments with the patient has not yet been investigated. It will be useful during the estimation of the fulcrum point to determine whether the instrument is interacting with soft tissue, which signifies that the fulcrum point cannot be estimated using the measured forces along the instrument.

Machine learning has also been widely used to generate temporally continuous trajectories based on the manipulator position or contact force measurements. Thus, the Learning from Demonstration (LfD) approach was defined (Calinon, 2009). It demonstrated that robots can perform tasks that have been previously trained by a human, such as pouring a glass of water using a bimanual robot (Jakel et al., 2010), hitting a table tennis ball, feeding a robotic doll (Calinon et al., 2010) or placing a ball in a hole inside a box (Rozo et al., 2013). However, none of these studies addressed the use of the LfD approach for haptic guidance. The key difficulty is to generate appropriate guidance force references to aid the task in real-time while considering the momentary input of the teleoperator, i.e., a desirable guidance system should not encourage the operator to follow a time based trajectory but rather provide support in response to operator commands.

These technologies, once extended and integrated, could solve inherent issues that arise during SPAS and result in a smart navigation system that can assist surgeons during SPAS procedures.

1.2 Contributions

This thesis provides theoretical and experimental results related to the smart navigation of surgical instruments handled by robots in SPAS. The instrument navigation has to consider the inherent constraints of SPAS to avoid undesirable forces on the patient and to perform the movements accurately. Moreover, the use of haptic guidance to assist the surgeon during the teleoperation appears to be a promising method that has not been thoroughly investigated. Therefore, this thesis proposes a smart navigation approach that integrates these issues into a global architecture, therefore, the main contributions of this thesis are:

- **Position control algorithm for surgical instruments in SPAS**

A navigation method based on a parallel force-position control that has been adapted to SPAS is proposed. This method uses a parallel force-position control to perform the instrument’s movements taking into account the fulcrum point constraint and minimizing the forces exerted on the abdominal wall. Moreover, taking advantage of SPAS (i.e., both instruments are inserted by the same trocar) the fulcrum point is estimated using a soft tissue interaction detector and a fusion measurement method.

- **Soft tissue interaction detector**

It is necessary to detect interactions between instruments and soft tissue in order to reduce fulcrum point estimation errors. Thus, a machine-learning algorithm based on previously trained surgeon movements is proposed. These movements are divided into different states that provide information about the interactions of the instruments with the soft tissue inside the abdomen. Therefore, the purpose of this algorithm is to recognise the surgeon’s movements and the current state in real time during the surgery and thereby to detect whether any interaction with soft tissue occurs.

- **Learning from Demonstration haptic guidance approach**

To extend the surgeon's capabilities during teleoperation, a generic approach for haptic guidance using the learning from demonstration methodology has been defined. This approach takes into account previously trained movements and their relation to sensory information during the movement. Using this information, haptic forces are generated and transmitted to the surgeons to assist them during teleoperation.

- **Implementation of the proposed smart navigation method and experimental results**

The proposed methods were implemented in two teleoperated robotic platforms. To experimentally validate the proposed smart navigation method for SPAS, the CISOBOT platform, which was developed by the Universidad de Málaga, was used. Its experimental results demonstrated the performance of the navigation method. The haptic guidance approach, based on the LfD method, was implemented in the LWR Taskboard, which was developed by the Telerobotics and Haptics Laboratory at ESA-ESTEC. Using this platform, a generic task (i.e., peg-in-hole insertion) was performed to validate the proposed method for haptic guidance.

1.3 Context and Motivation

This thesis has been conducted in the context of research studies on surgical robotics in the System Engineering and Automation research group at the Universidad de Málaga. Furthermore, a three-month research stay was performed in the Telerobotics and Haptics Laboratory at the European Space Agency to research the use of haptic guidance in space teleoperation.

The results of this thesis extend previous studies related to the teleoperation of surgical robotic platforms (Bauzano, 2012) and surgeon gesture detection (Estebanez, 2013). These studies formed part of the surgical robotic research activities performed at the Universidad de Málaga, whose first primary achievement was the design and implementation of a laparoscopic camera robotic assistant that was successfully used in human surgeries (Muñoz et al., 2006). Currently, the activities of this team are focused on developing perceptual, navigational and cognitive technologies for application to surgical robotics. This thesis was supported by Spanish national projects under action DPI2010-21126-C03-01 and DPI2013-47196-C3-1R. The primary objective of these projects has been to provide autonomy in surgical navigation using cognitive and machine learning algorithms (Rivas-Blanco et al., 2014) (Bauzano et al., 2013) (Bauzano et al., 2014).

Furthermore, the Telerobotics and Haptics Laboratory at ESA-ESTEC are involved in different projects related to the teleoperation of robots using haptic devices. One of them is the METERON project, which consists of developing technology for haptic teleoperation from space. During the performed research stay, a generic approach for haptic guidance was developed. It has been used to solve generic tasks, e.g., peg-in-hole, and it is proposed in this thesis to solve specific tasks in surgical teleoperation.

1.4 Thesis Outline

This thesis is divided into six chapters, six appendices and bibliographical references. Each chapter, except this one, starts with an introduction that presents the problem to be solved, followed by the structure of the chapter, and ends with the conclusions that highlight the contributions and/or results that have been obtained.

The structure of this thesis is focused on solving different issues that arise when a teleoperated surgical robotic platform is developed for SPAS: a navigation method that considers the fulcrum point constraint as a virtual Remote Centre of Motion, a soft tissue interaction detector based on Machine Learning, used to improve the fulcrum point estimation, and a haptic guidance system based on Learning from Demonstration that is able to assist the surgeon during teleoperation. Then, Chapter 2 introduces the background to the current and past history of surgical teleoperation, providing a brief description of the most important surgical robotic platforms. Later on, the state of the art in Machine Learning for surgical robotics is described. This methodology has been specifically used to solve two issues that are relevant to this thesis: the use of Hidden Markov Models for gesture recognition and the use of Learning from Demonstration for robot trajectory generation based on previous training. Finally, this chapter describes the use of haptic guidance in surgical robotics.

Chapter 3, titled “Smart Navigation for Single Port Access Surgery”, proposes a navigation method for robotic SPAS that can perform movements of two surgical instruments inserted through a multiport trocar based on the fulcrum point constraint that is estimated from the forces and torques measured along the instrument. To improve the fulcrum point estimation, the proposed navigation method uses a measurement fusion method that takes advantage of the fact that both instruments are inserted through the same trocar. However, the use of the forces and torques along the instrument to estimate the fulcrum point requires that the instruments interact only with the patient’s abdomen. Therefore, a soft tissue interaction detector, also presented in this chapter, is used to detect whether the instrument tip interacts with the soft tissue inside the abdomen. Finally, a parallel force-position control scheme is used to minimize the forces exerted on the patient’s abdomen.

Having defined a navigation method for teleoperation in SPAS, Chapter 4 proposes the use of Learning from Demonstration for haptic guidance. Thus, a generic approach is presented in this chapter for solving generic tasks. This approach is divided into two stages. The first stage consists of dividing the task into different gestures and training them by expert demonstrations, and the second stage uses this training during the reproduction of the task to assist the operator in haptic guidance.

Chapter 5, “Implementation and Experiments”, describes the experiments that were conducted to validate the proposed smart navigation system for SPAS. The navigation method for SPAS was tested by analysing the exerted forces, fulcrum point estimation errors and soft tissue interaction detector delays. Conversely, the haptic guidance method, which is based on LfD, was validated using the generic peg-in-hole insertion task, whose experimental results demonstrate the performance of the training and reproduction stages.

Chapter 6 highlights the most relevant contributions of this thesis and proposes future research topics. Finally, the appendices provide a stability analysis of the position control scheme used in Chapter 3 and the theory of the mathematical models that have been used.

2 Background

2.1 Introduction

It has been nearly 30 years since robotic systems were first used in surgery, and they have been adapted to different surgical techniques, which have evolved from laparotomy (open surgery) to Minimally Invasive Surgery (MIS) or Laparoscopic Surgery (LS). This evolution followed the hypothesis that reducing the number and size of incisions reduces both patient recovery time and the likelihood of postoperative complications (Reza et al., 2006; Eskicorapci et al., 2007). The LS technique involves the use of specialized instruments and a camera that are introduced into the abdomen through small incisions to allow the surgeon to manipulate the inner organs. Three primary LS approaches are currently considered:

- Single Port Access Surgery (SPAS), a laparoscopic surgical procedure in which the surgeon operates almost exclusively through a single incision or entry point, typically the patient's navel.
- Natural Orifices Trans-luminal Surgery (NOTES), which is performed with a flexible endoscope that is passed through a natural orifice (mouth, urethra, anus, etc.) with the instrument introduced into the abdominal cavity through an internal incision in the stomach, vagina or colon.
- Hand Assisted Laparoscopic Surgery (HALS), a recent approach that combines a mini-laparotomy with laparoscopic surgery: the surgeon inserts one hand into the abdominal cavity to directly manipulate the inner organs while the other hand uses a standard laparoscopic instrument.

Although current trends are moving towards the development of the previously described techniques, several studies (Perino et al., 1999; Schlachta et al., 2001; Agachan et al., 1997) have revealed that the implied difficulty in the use of instruments for manipulating the organs using standard video image feedback implies that surgeons need an extensive training process to acquire the skills required to successfully perform these surgical procedures. In fact, the surgeon's learning curve significantly increases when performing LS compared with NOTES.

Hence, teleoperated robotic systems for laparoscopic surgery have been designed with the goal of addressing the above difficulties by providing higher precision and intuitive movements, i.e., a more natural management of the surgical instruments and a three-dimensional view of the surgical area. However, although the enthusiasm for laparoscopic techniques is rapidly increasing, the use of robotic assistants presents several issues. The first one is related to the navigation of instruments inside the abdomen. Depending on the approach, these instruments are moved considering their inherent constraints, e.g., in traditional LS, each instrument is moved based on the fulcrum point at which each instrument is inserted. However, in SPAS, all of the instruments are inserted through the same fulcrum point, which could lead to more constraints and collisions between the instruments inside and the manipulators outside the patient, e.g., the current commercialized surgical robotic platform (da Vinci) uses a mechanical remote centre of motion that entails large kinematic structures that could collide during SPAS. Moreover, the fact that both instruments are inserted through the same trocar and crossed, as indicated in Figure 2-1, requires better surgical abilities when the instruments are handled in this manner.

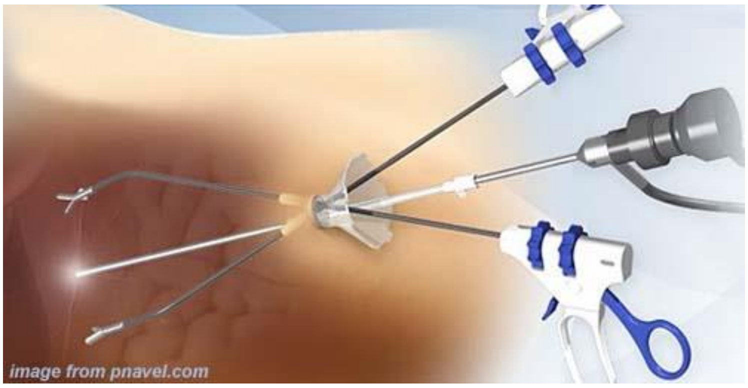
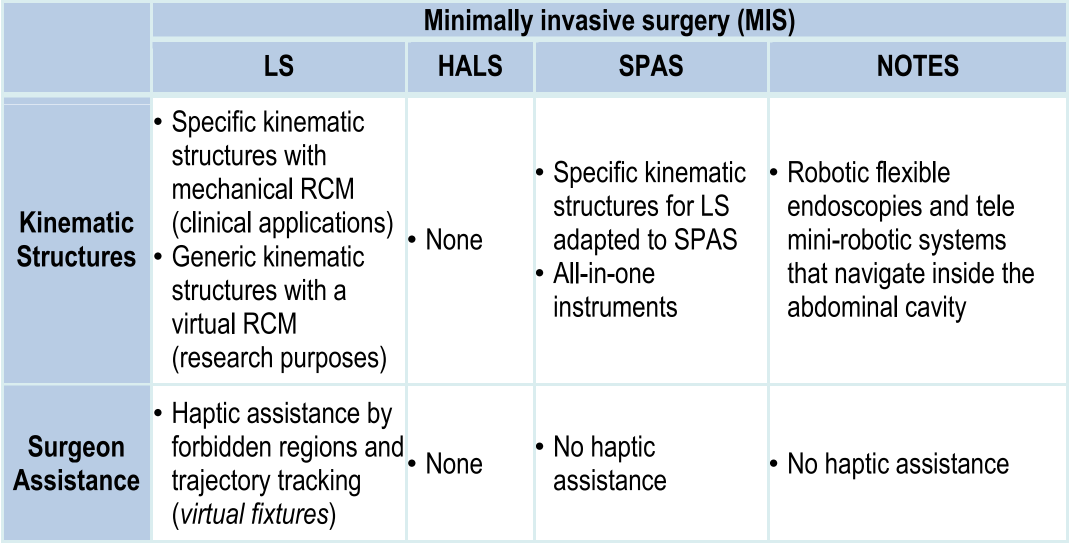


Figure 2-1. Position of instruments in SPAS

These issues have implied that teleoperated surgical robotic systems have not experienced a substantial growth in the number of procedures that can be performed by new minimally invasive techniques. Moreover, these systems do not provide the required assistance to surgeons, which would be useful in improving surgeons' skills. In fact, it should be noted that robotic surgery is currently stagnant owing to the lack of available commercialized teleoperated robots: current robotic technologies are not useful for new minimally invasive surgical procedures such as SPAS or NOTES. The following table presents the current state of the application of robotic technologies for each of the mentioned surgical techniques considering the stated issues.

Table 2.1. Current Applications of Robotics in Different Surgical Techniques



	Minimally invasive surgery (MIS)			
	LS	HALS	SPAS	NOTES
Kinematic Structures	• Specific kinematic structures with mechanical RCM (clinical applications) • Generic kinematic structures with a virtual RCM (research purposes)	• None	• Specific kinematic structures for LS adapted to SPAS • All-in-one instruments	• Robotic flexible endoscopies and tele mini-robotic systems that navigate inside the abdominal cavity
Surgeon Assistance	• Haptic assistance by forbidden regions and trajectory tracking (*virtual fixtures*)	• None	• No haptic assistance	• No haptic assistance

Consequently, and considering the previous information, it is clear that it is necessary to develop new navigation methods that use lightweight structures to perform instrument movements based on the inherent constraints of these new surgical techniques. Moreover, robotic technologies need to lead to a more natural interaction involving greater comfort for surgical staff, thus making the learning process quick and effective (Gomes, 2011). In this context, the use of robotic assistance appears to be an appropriate tool for improving the surgeon's experience by taking advantage of robotic precision and human intelligence.

Therefore, a smart navigation system integrated with a teleoperated robotic platform is needed. This smart navigation system should be able to move surgical instruments by teleoperation, taking into account the inherent constraints of the surgical technique and assisting surgeons during the procedure to improve their skills. Different technologies must be combined and expanded to achieve the proposed smart navigation system, as depicted in Figure 2-2 and described below.

The navigation of surgical instruments inside the patient's abdomen has been addressed using different methods that have been implemented in surgical teleoperated robotic platforms, as described in Section 2.2. These navigation methods are primarily based on specific large kinematic structures or robotized all-in-one instruments, which are not useful for SPAS because lightweight structures based on external manipulators that apply enough forces to push up organs are required, as described in Section 2.2.3.

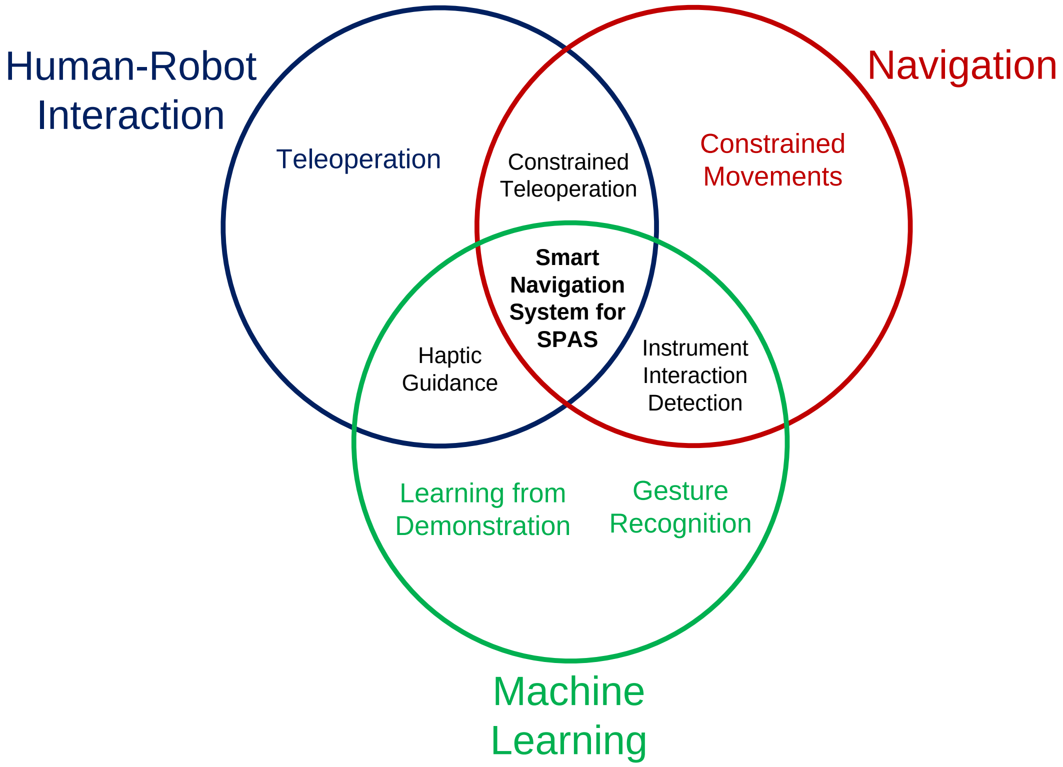


Figure 2-2. Use of different technologies to create the proposed smart navigation system

Therefore, a new smart navigation method for generic and lightweight structures that takes into account the inherent constraints of the surgical technique is needed. Thus, the proposed navigation method should recognize the movements of the surgeon and detect the interaction between the instrument and the patient to adapt the movements to the required constraints. In this sense, machine-learning algorithms has been widely used in robotic surgery to recognize the stage of a surgical procedure that is being performed, as explained in Section 2.3.1. However, these algorithms have not yet been used to detect the interaction of the instrument with the patient, which is useful for performing instrument-constrained movements.

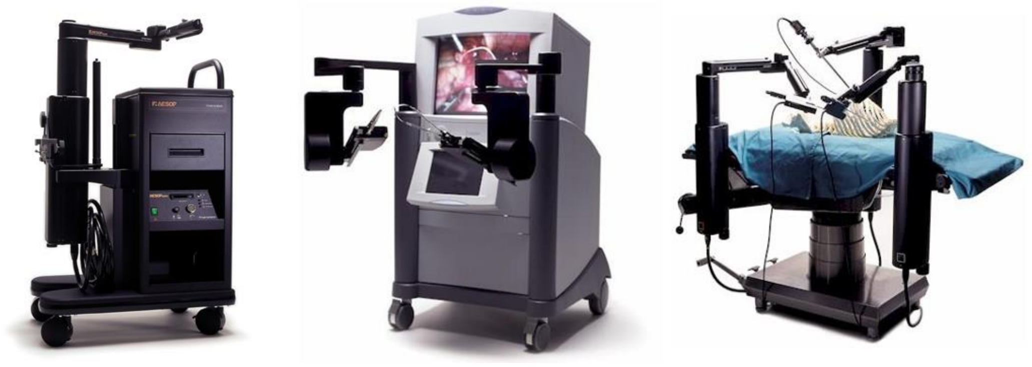
Once the navigation problem is solved, the surgical instruments must be moved by surgeons using a human-robot interaction that allows constrained teleoperation in an easy and comfortable way for the operator. For this purpose, several master consoles, which provide human-robot interactions, have been developed, as described in Section 2.2. These consoles provide direct or haptic teleoperation, with certain consoles providing force feedback to the surgeon. The force feedback can be used to perform haptic guidance during the reproduction of a task to follow a predefined trajectory or avoid forbidden regions that were previously defined using images or marks, as described in Section 2.4. However, the use of sensory information from the robot, e.g., position, velocity, and force and/or torque, to derive haptic guidance trajectories in real-time has not been explored. The primary problem is to generate guidance force references based on sensory information. Learning from demonstration is a machine Learning based approach that is able to learn human skills by guiding a robot to perform a task using its sensory information. This approach has been successfully used to learn and reproduce human skills such as pouring a glass of water, hitting a table tennis ball or feeding a robotic doll, as described in Section 2.3.2. However, this approach remains to be investigated for providing haptic guidance trajectories that depend on sensory information and have been previously learned from multiple demonstrations.

2.2 Teleoperated Surgical Robotic Platforms

Although the use of autonomous robotic systems to perform surgeries is in the distant future, we currently cannot simulate human intelligence using computers. Meanwhile, robotic systems are remotely controlled by humans that use master devices to guide them. This methodology is commonly known as teleoperation or telesurgery. Since the first teleoperated surgical robotic platform (in the 1990s), different platforms have been developed for either commercial or research purposes.

2.2.1 Commercialized Robotic Platforms

The first commercialized surgical robotic platform was the AESOP arm (Mettler et al., 1998) (Figure 2-3.a). It was developed by Computer Motion in the 1990s under a NASA contract. AESOP was approved by the US Food and Drug Administration (FDA) to perform surgeries in 1994, and it was the first robot used to perform a surgery. The primary limitation of AESOP was that it was designed to handle only the laparoscopic camera, and it was guided by voice. This was the initial step towards ZEUS (Butner and Ghodoussi, 2001), which was the first teleoperated surgical robotic platform; its first prototype was shown in 1995 and tested on animals in 1996. In 2001, after several clinical trials on humans, the FDA approved its commercialization. The ZEUS was composed of a master console (Figure 2-3.b) with which the surgeon could handle two passive devices with his hands (no haptic feedback), and these movements were reproduced by two slave arms (Figure 2-3.c). Moreover, visual information was provided by a laparoscopic camera and two screens embedded within the master console, which were used to show the laparoscopic image and provide information about the robotic system.

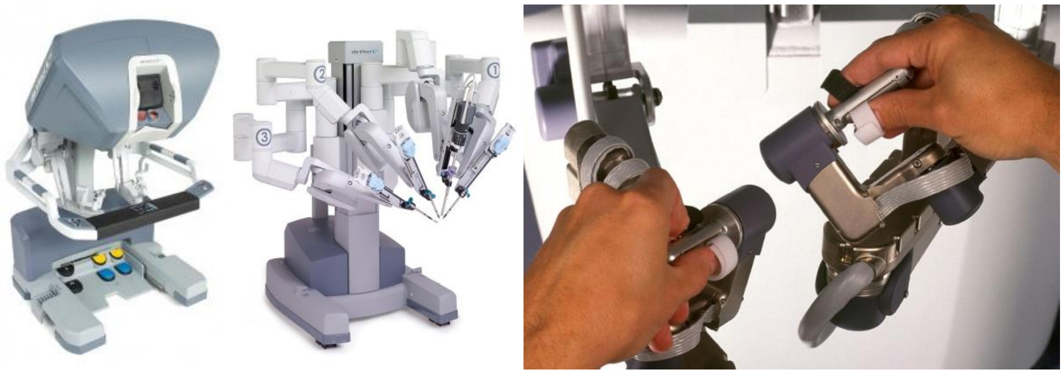


(a) (b) (c)

Figure 2-3. (a) Aesop; (b) ZEUS master console; and (c) ZEUS slave arms

In 2003, the ZEUS system was discontinued because Computer Motion and Intuitive Surgical merged into a company to use their efforts in developing the most well-known commercialized surgical robotic platform, the da Vinci surgical system (Guthart and Salisbury, 2000). In 1995, Intuitive Surgical started licensing technology from IBM, NASA, SRI International and several universities to develop the first prototype of a surgical platform in 1997. This prototype, called Lenny, was used for animal trials. Afterwards, a new prototype for human trials was developed and used in vascular and gynaecological procedures in the same year. The market-ready version of the robot was called da Vinci, and clinical tests started in 1999. The FDA approved its use on humans in 2000. Since then, the system has been improved with HD-3D vision, advanced ergonomic features and the possibility of using two consoles for assisted surgery. As of December 31st, 2014, this company had sold approximately 3,000 units worldwide, and more than 1.5 million surgeries had been performed (Intuitive Surgical, 2014). The most successful application of this platform is the prostatectomy, approximately 90% of these surgeries are performed using this platform in the USA (Haidegger, 2011).

The da Vinci system is composed of a master surgeon’s console (Figure 2-4.a) and four slave robotic arms (Figure 2-4.b) that hold the surgical instruments. The surgeon’s fingers grasp the master console controls (Figure 2-4.c), and his movements are reproduced by the slave arms, thus providing enhanced dexterity, better precision and ergonomic comfort. Although this system was conceived for telesurgery, only the latest version, the da Vinci Si (launched in April 2009), provides further displacement of the master console. Previous versions used optic fibre-based short-distance protocols to communicate between the master and slave systems.

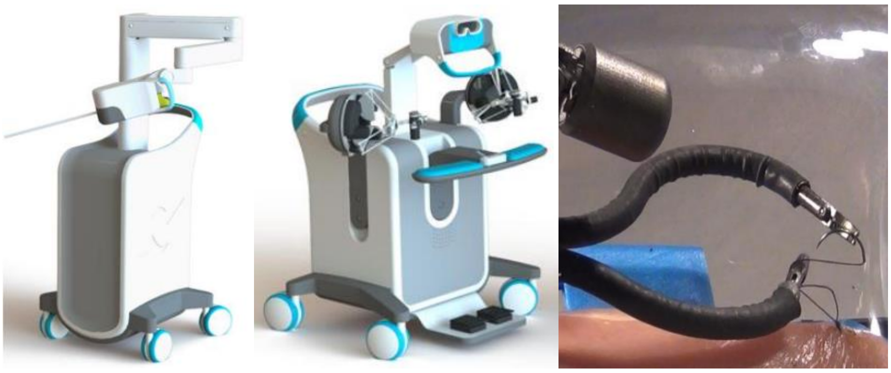


(a) (b) (c)

Figure 2-4. (a) da Vinci master console; (b) slave arms; and (c) master console controls

None of these described platforms has been conceived for SPAS, which provides some benefits for patients, as explained in Section 1.1. In the case of the da Vinci system, even though several SPAS procedures have been performed using special instruments (Kroh et al., 2011), the system is too large to perform SPAS without its arms colliding. The primary reason for its large structure is the use of a mechanical Remote Centre of Motion (RCM), which is managed by parallelogram arms. Furthermore, this system does not provide haptic feedback to surgeons, which could improve tissue characterization (Tholey et al., 2005).

Currently, Titan Medical Inc. is developing the SPORT (Single Port Orifice Robotic Technology) Surgical System (Figure 2-5), which is expected to be commercialized in 2017 (Titan Medical, 2015). The goal of this company is to extend SPAS into general surgeries, such as cholecystectomies or appendectomies. This system is composed of a slave robotic system and a master console. The slave robotic system includes one arm that holds an all-in-one instrument composed of two articulated tips and a 3D camera. Additionally, the master console provides a 3D endoscopic view of the patient’s body, as well as two haptic devices that are used to teleoperate the instrument tips.

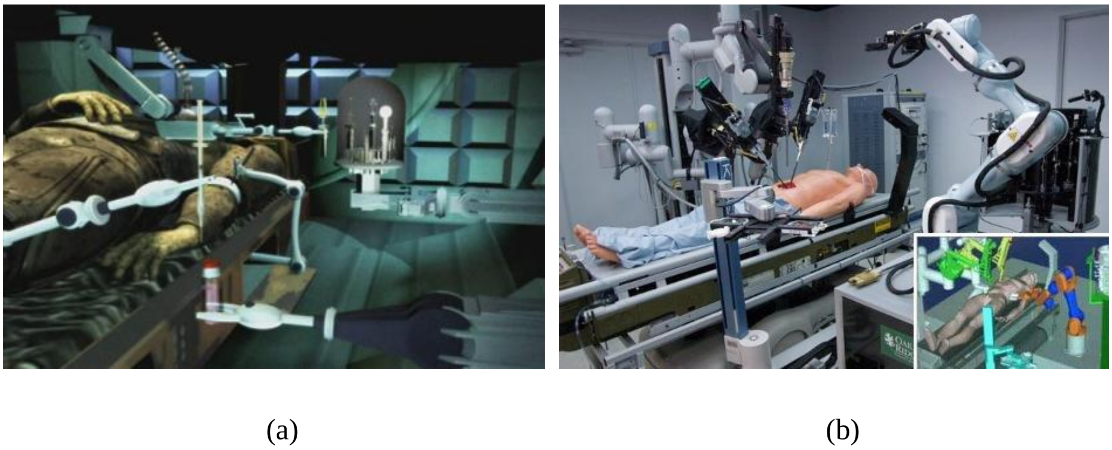


(a) (b) (c)

Figure 2-5. (a) SPORT slave robotic system; (b) master console; and (c) instrument tips with a camera

2.2.2 Research Prototypes

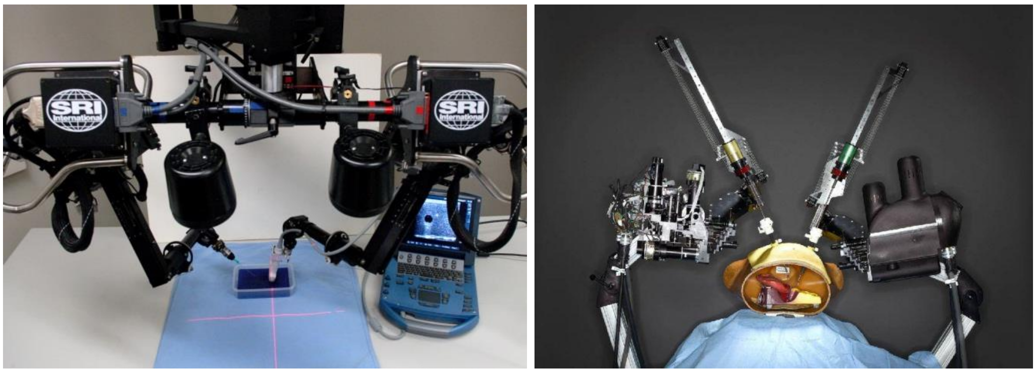
In addition to the previously described surgical systems in the research field, the Trauma Pod was one of the most ambitious projects related to surgical robotics (Garcia et al., 2009). The goal of this project was to create a teleoperated surgical robotic platform that would be integrated into an unmanned mobile robot to assist soldiers in the battlefield (Figure 2-6.a). This robotic platform would be controlled by a human surgeon from a distance (telesurgery), and they would perform vital surgery to save the soldier's life. This project was initiated in 1994 by the Defence Advanced Research Projects Agency (DARPA). The first stage, which was completed in 2008, used a da Vinci surgical system with a Mitsubishi robot that provided the surgical tools to the da Vinci platform using a tools exchange machine, as depicted in Figure 2-6.b.



(a) (b)

Figure 2-6. Trauma Pod project: (a) Frame of the video; and (b) First stage demonstration

Others approaches have focused on lightweight structures. SRI International developed the M7 (Figure 2-7.a) surgical robot in 1998, which is a portable and deployable lightweight robot that consists of two 7 DOF arms equipped with motion scaling, tremor filtering and haptic feedback. This system has been primarily used to test telesurgery under variable gravity (King et al., 2008). However, the most famous lightweight robot currently used for research is the Raven II (Figure 2-7.b), which was developed by the University of California at Berkeley, Davis and Santa Cruz (Lum et al., 2009). It has two articulated arms fitted with a shaft for different surgical tools. This system has been designed for research, and the manufacturer provides full access to its software interfaces using ROS (Robotic Operating System).



(a) (b)

Figure 2-7. (a) M7; and (b) Raven II

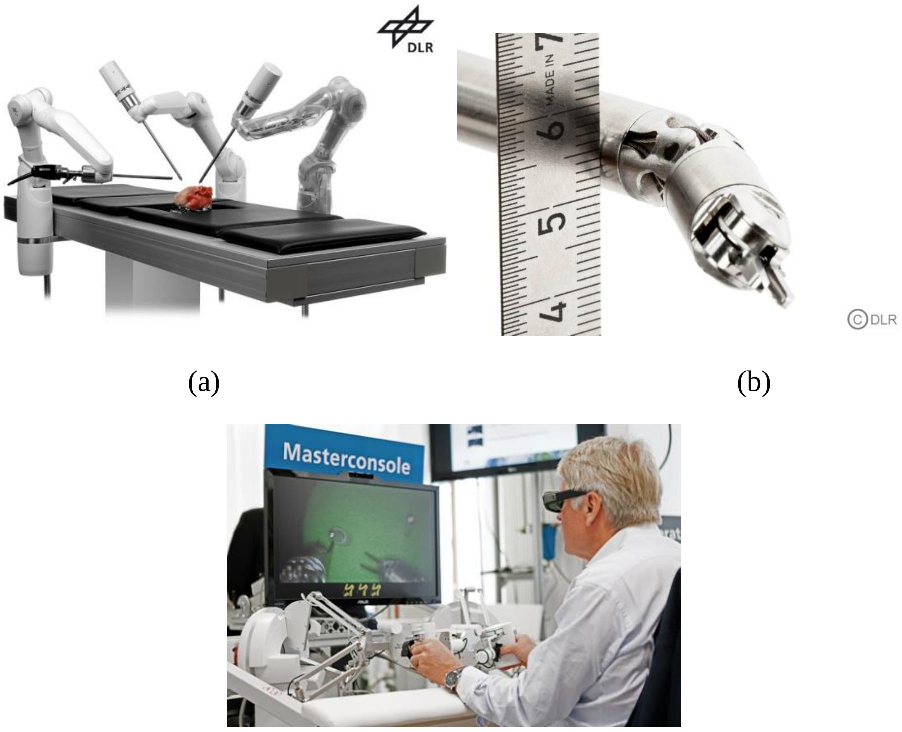
The German Aerospace Centre (DLR) has developed a lightweight manipulator-based surgical system known as MiroSurge (Hagn et al, 2010). The primary objective of this system is to provide a versatile platform for open and minimally invasive surgery. This versatility involves the following parameters:

- Multiple surgical procedures: visceral, orthopaedic and neurosurgery
- Adaptable setup: the number of manipulators and their location can vary
- Different modes: position, compliance and admittance control
- Different interactions: hands-on, teleoperation and haptic
- Additional technologies can be integrated: e.g., visual servoing, augmentation, etc.

To achieve these features, this system is composed of specific manipulators for medical applications (Figure 2-8.a) known as MIRO robots, which were also designed by the DLR (Hagn et al., 2008). These manipulators hold a specific instrument (Figure 2-8.b), called the MICA, whose tip is articulated and serves as an end effector inside the abdomen, thus extending the dexterity of the system. In 2010, this instrument was a gripper; however, the DLR was developing three different instruments: a Metzenbaum scissor, a Maryland dissector and a conventional needle holder. Moreover, these instruments could provide force and torque feedback inside the body using a sensor placed on the instrument tip (Seibold et al., 2008). Owing to its position, the measured data are affected by neither friction in the trocar nor exerted forces in the abdominal wall, which improves the haptic perception. However, as stated in the previous section, this type of sensor causes several issues related to sterilization and temperature changes when the sensor is in contact with tissue.

To teleoperate this system, the DLR also developed a surgical master console that uses two sigma.7 haptic devices (Figure 2-8.c). These haptic devices, which have been developed in collaboration with Force Dimensions Inc., have seven fully actuated degrees of freedom: three for translational motion, three for the rotation of the wrist and one for the gripper. Using this configuration, the device can provide up to 20 N of force feedback and a rotational sphere of approximately 120 mm diameter (Tobergte, et al., 2011).

The main advantage of this system is the flexibility of its slave arms (MIRO robots) because they can be placed in different locations based on the surgical procedure to be performed. In comparison with the da Vinci surgical system, their arms are thinner because they use a generic kinematic structure with a virtual RCM that is calculated using the force-torque information from the manipulator end-effector.



(a) (b)

(c)

Figure 2-8. (a) DLR Miro Robots; (b) MICA instrument tip; and (c) mirosurge master console.

All of the previously described platforms were conceived for traditional laparoscopic surgery and have not been used in SPAS. Several robotic platforms have been developed within the last five years that are more suitable for SPAS. These platforms are typically all-in-one robotized instruments that contain everything to perform a simple surgery.

The novel Insertable Robotic Effectors Platform (IREP) has been developed (Xu et al., 2009) and improved (Ding et al., 2010) by Columbia University since 2009. This system is composed of a 3D stereo vision module mounted on a 3 DoF articulated arm that provides pan, tilt and zoom and two 8 DoF dexterous arms with two grippers (Figure 2-9). These arms are initially folded in a cylindrical configuration that allows the instrument to be introduced through a 15 mm diameter trocar. Once inserted, two five-bar mechanisms deploy the dexterous arms and adjust the distance between them. Furthermore, the two dexterous arms can exert a force of up to 56.2 N with an average of approximately 20 N. The maximum force is reached only for a concrete articular configuration of the arm.

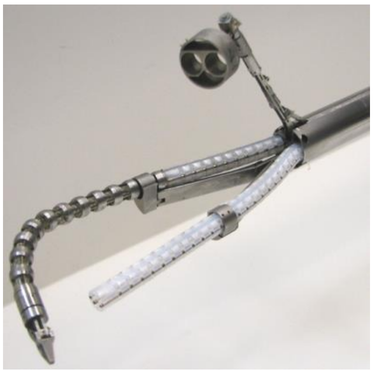


Figure 2-9. IREP Surgical system

The Imperial College of London is involved in the iSnake project, which consists of designing different multitasking robotic platforms for Minimally Invasive Surgery (MIS). In (Kwok, 2013), they developed a system similar to the IREP. The primary advantage of this system is the use of two flexible arms based on tendons driven by DC motors. This system is wider than the IREP because it is introduced through a 30 mm diameter trocar. The maximum exerted forces of the arms have not been documented. However, this is the only system that currently provides haptic guidance to avoid manipulation limits during teleoperation. It has been validated through in vivo trials with pigs, as depicted in Figure 2-10.

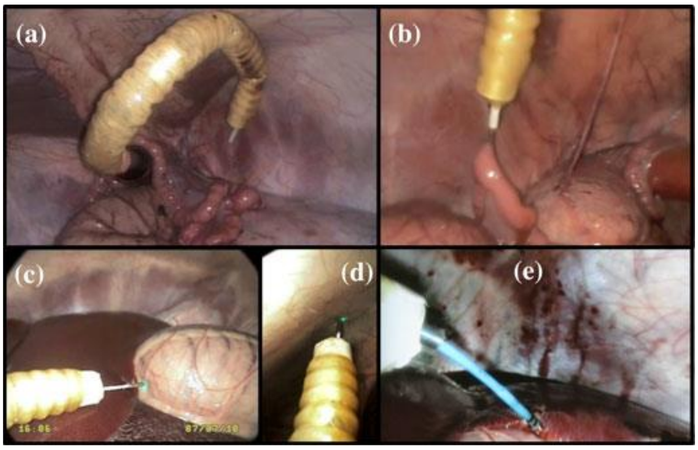


Figure 2-10. iSnake experimental trials (Kwok, 2013): (a) In vivo trial; (b) Grasping the uterine horn; (c) and (d) Laser endomicroscopy; and (e) Diathermy conducted on the liver

Another all-in-one instrument has been developed by the BioRobotics Institute at the Scuola Superiore Sant'Anna (Piccigallo et al., 2010; Niccolini et al., 2012). The SPRINT (Single-Port lapaRoscopy bimanual robot) is a teleoperated surgical system that is composed of two high-dexterity 6 DoF arms and a 3D camera (Figure 2-11). The arms are controlled by two external DC motors that actuate the shoulders, as well as four internal motors that actuate the elbows and the wrists. This configuration provides forces of up to 5 N at the instrument tip, which is not sufficient for complex surgeries where organs need to be pushed up. The result is a robotic system that can be introduced through a 30 mm diameter trocar. It is teleoperated by a dedicated master console that uses two Phantom Omni devices (Sensable Technologies Inc.) to provide haptic teleoperation. To improve the end-effector control, each Phantom Omni device has been modified by adding a customized handle interface that converts the position of the surgeon’s fingers into movements of the slave arm end-effectors. This system has internal motors to actuate the arms whereas the previously described systems for SPAS are actuated using external motors and tendons or cables.

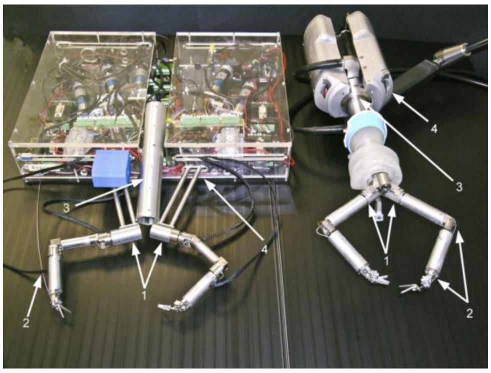


Figure 2-11. SPRINT surgical system

2.2.3 Comparison of Described Platforms

As stated in this section, surgical robotic platforms can be divided into all-in-one instruments and external robotic platforms. The primary advantage of the first type is size; they are compact and do not need special features in the operating room. However, external robotic platforms have a larger size and typically require special features in the operating room because they are composed of a minimum of three manipulators: two for surgical tools and one for the camera. Nevertheless, because of their size, the manipulators can apply higher forces inside the abdomen. In fact, in the case of a cholecystectomy, the required peak forces inside the abdomen are higher than 20 N (Richards, C. et al., 2000). These forces are easily applied by external manipulators owing to their dimensions; however, in the case of internal robots, size is a significant limitation that effects the maximum torque of its motors, which are unable to exert enough forces to hold organs during a surgery; e.g., during a cholecystectomy, the liver needs to be pushed up. However, the use of external robotic platforms for SPAS involves several issues. One issue is the RCM, which is needed to move the surgical instruments correctly and has been determined using a mechanical RCM involving specific kinematic structures (da Vinci) and virtual or software RCMs used in generic kinematic structures (DLR MiroSurge).

2.3 Machine Learning in Surgical Robotics

Machine learning is a scientific discipline and a subfield of Artificial Intelligence, which studies the use of algorithms that learn from previous information, thus building a probabilistic model that can predict, decide and/or recognize with high accuracy. These algorithms have been widely used in surgical robotics for recognizing gestures, i.e., movements that provide information about the surgical procedure stage, and robotic assistance, e.g., to evaluate the surgeon's skills, thus comparing the performed gestures with ones that have been previously trained by expert surgeons. In human-robot collaboration, it has been used to detect the performed gesture and the stage of the surgical procedure when moving a robot to assist the surgeon. Finally, machine learning has been used to perform automatic movements that have been previously trained by an expert surgeon and encoded using the Learning from Demonstration (LfD) approach. In this section, recent contributions in gesture recognition and LfD are presented. These contributions will be extended in this thesis for two purposes. The first one is the recognition of surgeons' gestures to detect instrument interactions with the patient's body, which is useful to perform constrained navigation taking into account the limitations of SPAS. The second one is the use of this recognition and the LfD approach to detect the gesture that is being performed and to derive haptic guidance trajectories based on sensory information from the robot.

2.3.1 Gesture Recognition

In general, a gesture can be defined as a corporal expression composed of the physical movements of the fingers, hands, arms, head, face or body that provides meaning in a similar way to the use of language to communicate with other people. In MIS, these gestures can be performed using surgical instruments (e.g., pushing up the liver, holding or cutting tissue, or tying knots), and these gestures provide information about the stage of the surgical procedure and even about the ability of the surgeon if the gesture is analysed. Therefore, the use of machine learning algorithms appears to be a useful tool for encoding surgical procedures into statistical models and detecting the stage of a procedure by recognizing the surgeon's gestures.

Some studies have demonstrated the use of different machine learning methods, such as Markov Models and Bayes classifiers, to detect and classify a surgeon's gestures. In particular, (Rosen et al., 2001) proposed the use of Markov Models to classify surgeons' skills into two sets: novice surgeons and expert surgeons. For this purpose, a laparoscopic grasper was equipped with a three-axis force/torque sensor that was used to measure the forces and torques at the hand-tool interface; these measurements were synchronized with a video that was used to identify the performed manoeuvres. An analysis of the video and a vector quantization algorithm allowed 14 different types of tool-tissue interactions to be defined with their corresponding force-torque signatures. Two Markov Models were developed for each type of interaction, which represented the performance of the surgeons (grouped into novices and experts). Then, these models were used to generate an index that indicated the level of experience during the surgical procedure. In the field of teleoperation, (Lin et al., 2005) used the da Vinci surgical system to develop a method for detecting and segmenting surgical gestures, which were used to evaluate overall proficiency and specific skills during teleoperation. To do this, they used a Bayes classifier to recognize the performed gestures using the Cartesian position of the da Vinci master manipulators by classifying them into two sets (expert and intermediate). When using this technique, nearly 90% of gestures were recognized, and the use of the Hidden Markov Model was proposed to increase the recognition rate to 95%.

Some researchers have modelled a surgical procedure or a particular surgical task as a set of basic actions or subtasks, which can be combined to form the overall task (Muradore et al., 2011). Depending on the current state of the procedure, the robotic assistant will perform the corresponding action. Based on this paradigm, collaboration during a suturing task has been proposed in (Padoy and Hager, 2011) using the da Vinci Surgical System. In (Weede et al., 2012), the different phases of a single-port sigma resection were identified based on the instruments and coagulation analysis. Other studies proposed a sensor platform to detect the procedure stage using a range of sensors located in different elements of the operating room, as well as attached to the clinicians' bodies (Bardram et al., 2011).

By considering the modelling of a surgical procedure as a state chart, (Ko et al., 2010) proposed a surgical procedure model that provided awareness of the current surgical stage being performed. To do this, they defined a surgery procedure model (SPM) that is modelled using a state-transition diagram, where states are defined as sub-procedures and transitions are defined as changes in state. These transitions are triggered by events obtained from the image analysis, i.e., detecting the surgical tool and the tip position, as well as voice commands. This model was integrated within a robotic surgical system, and its performance was experimentally demonstrated in a modelled porcine cholecystectomy, as depicted in Figure 2-12. In this figure, each procedure stage has been defined as a node, and their transition probabilities are represented by arrows.

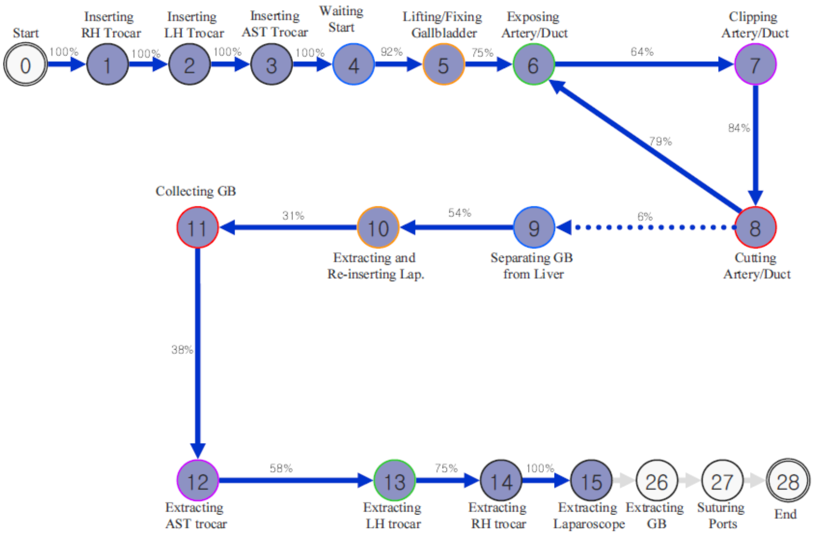


Figure 2-12. Proposed cholecystectomy state-transition diagram by (Ko et al., 2010)

As stated previously, image analysis and voice commands were used to detect the state transitions during the experiments. However, (Bauzano et al., 2014) extended this study by defining a workflow manager for robot-human collaboration that was able to follow the Rosser suture procedure (Fischer and Bland, 2007) online, thereby giving commands to a robot to assist the surgeon during the surgical procedure. To do this, the procedure was divided into four steps, known as manoeuvres, which were also subdivided into a sequence of gestures that was modelled as a state-transition diagram, as indicated in Figure 2-13.

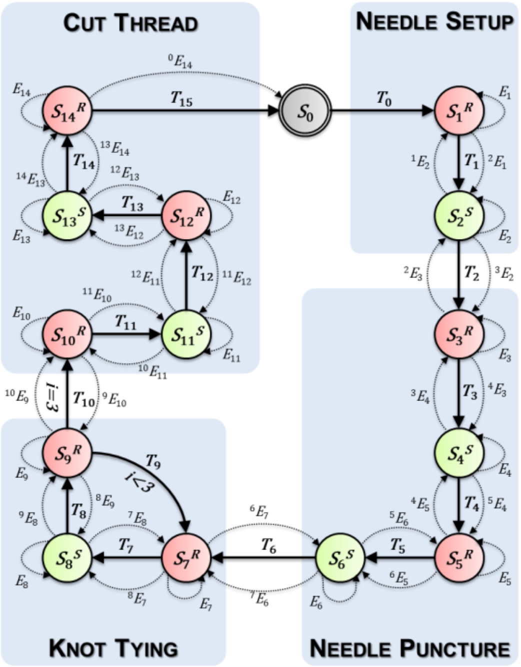


Figure 2-13. State-transition diagram that models the Rosser suture procedure (Bauzano et al., 2014)

To identify the surgeon's gestures, a recognition system was proposed. It is based on a set of previously trained gestures that was modelled using the Hidden Markov Model (HMM) and stored in a gestures library. Figure 2-14 illustrates the different modules that compose the gesture recognizer. Gesture acquisition was used to obtain the needed information from the tool position r_t to recognize the performed gesture. Once this information was obtained, the forward-backward algorithm was used to calculate the most likely gesture from the library, and the trigger condition activated the signal T_k if the recognized gesture corresponded to the reference gesture S_k^S.

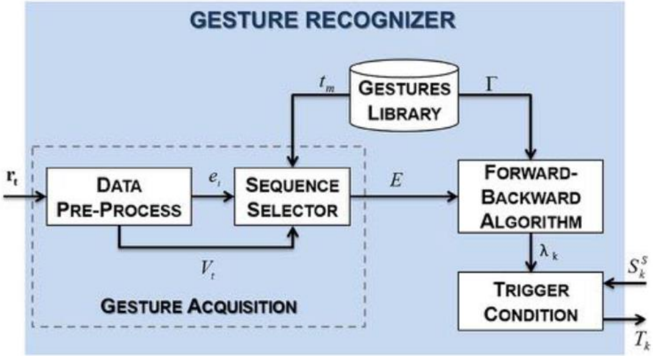


Figure 2-14. Gesture recognizer diagram (Bauzano et al., 2014)

To validate this recognizer, 15 trials were conducted by two non-specialized participants, resulting in an average of 81.25% of successfully recognized gestures. These results indicate that the recognizer is useful for reducing the use of voice commands to communicate with the robot assistant, as proposed in previous contributions, which allows the surgeon to concentrate during surgery.

As previously explained, machine learning has been used in the field of surgical robotics for gesture recognition to evaluate surgeons' skills or to recognize the surgical procedure stage. The experimental results have demonstrated the usefulness of this approach; however, it remains unclear whether the use of machine learning algorithms remains would be useful for dividing a gesture into states and for recognizing these states.

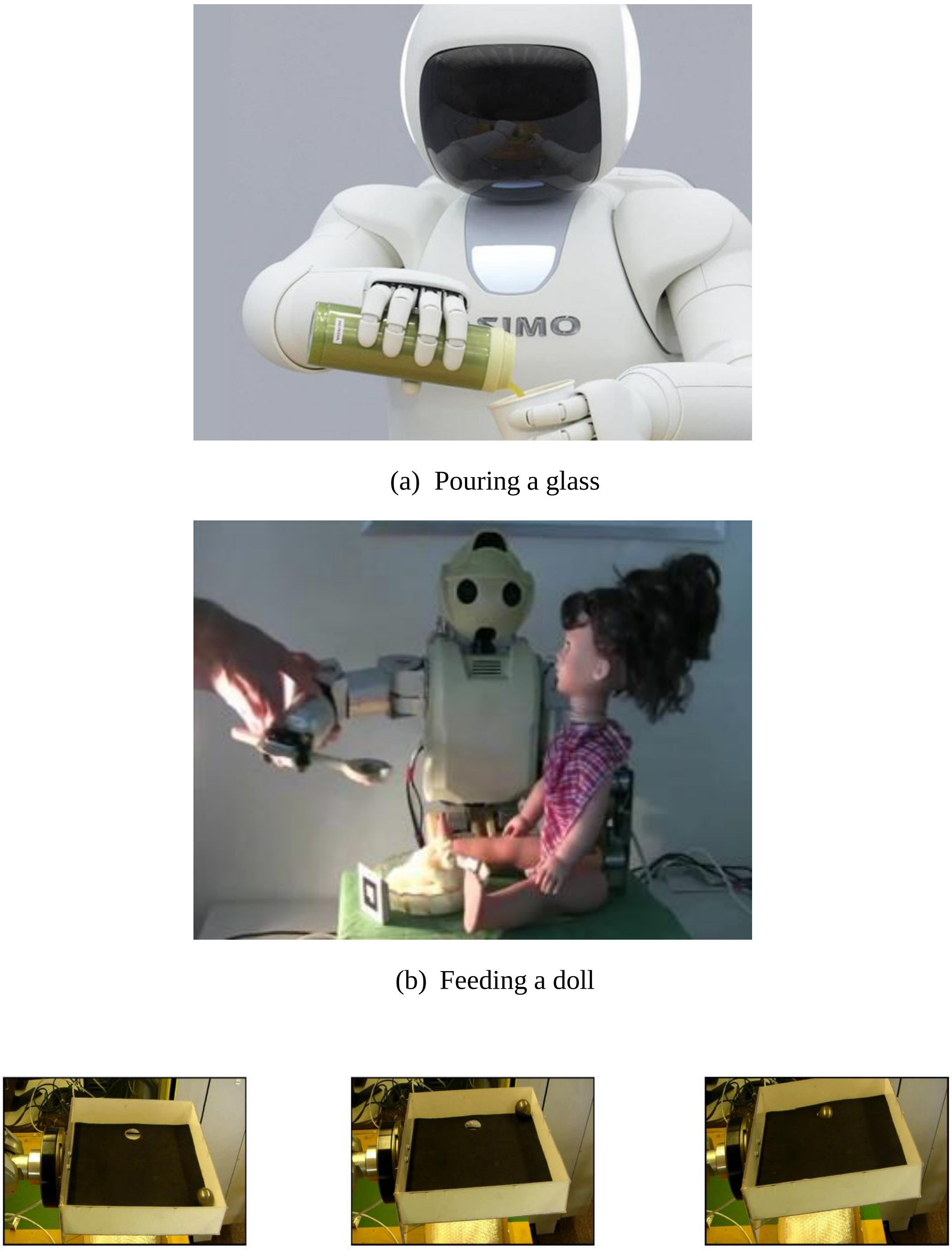
2.3.2 Learning from Demonstration

Learning from Demonstration (LfD) or Programming from Expert Demonstration (PED) is a methodology that consists of teaching a robot the movements that need to be performed to solve a task. Traditionally, a human programmer would have to code the task procedure for each concrete case, divide them into multiple subtasks and test them. If errors or new cases arise, the robot would have to be coded again. To solve this issue, LfD allows users to reproduce tasks by showing a robot how to do it without using code. If an error or a new case occurs, the user only has to provide new demonstrations.

Research on the LfD approach began in the 1980s with the goal of reducing the development and maintenance costs of manufacturing robots. Using this methodology, a company could avoid the traditional methodology for programming robots and replace it with a programming by demonstration approach. The first approaches to the LfD were based on teaching, guiding or playback (Segre, 1988). In these approaches, the demonstrations were performed through a teleoperated control and the primary task was divided into subgoals. The end-effector position and the forces applied during the movements were segmented into trajectory keypoints defined as primitives to achieve these subgoals. Later, probabilistic machine learning algorithms were proposed for the LbD approach in humanoid robotics (Shon et al., 2005).

One of the most used approaches for LfD in robotics was proposed by (Calinon et al., 2007). Essentially, this approach consists of creating a probabilistic model of a task using a Gaussian Mixture Model (GMM) and training it with multiple demonstrations that have been temporally aligned using Dynamic Time Warping (DTW) (Sakoe and Chiba, 1978). Then, the trained model is used to reproduce the task using Gaussian Mixture Regression (GMR). Afterwards, (Calinon et al., 2010) modified this approach by replacing the DTW with an HMM to handle the spatial and temporal variabilities of the task.

These approaches have been widely used to generate temporally continuous trajectories based on the manipulator position or the contact force measurements. Results of these developments have allowed robots to perform previously trained human tasks, such as pouring a glass of water using a bimanual robot (Jakel et al., 2010), hitting a table tennis ball, feeding a robotic doll (Calinon et al., 2010) or placing a ball in a hole inside a box (Rozo et al., 2013), as depicted in Figure 2-15. In human-robot collaborations, this approach has also been used to solve specific tasks. (Rozo et al., 2014) presented a modified LfD approach that allowed a robot to learn the desired path and the needed forces to collaborate with a human during the movement of bulky loads (Figure 2-16.a), and (Gu et al., 2011) solved the table-lifting task through the collaboration of a humanoid robot and a human (Figure 2-16.b). The latter task was divided into two stages: the first one enabled the robot to hold the table using the previously stated LfD approach, and the second stage involved the robot learning how to collaborate with the human to hold the table using a guided reinforcement learning algorithm.

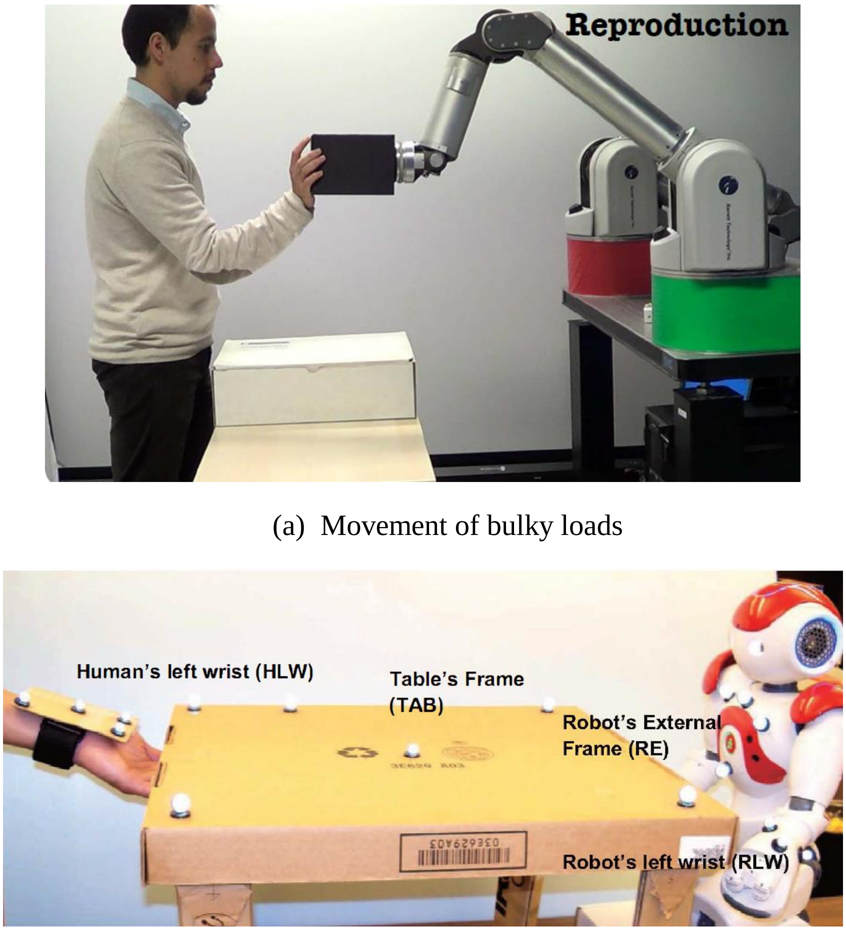


(a) Pouring a glass

(b) Feeding a doll

(c) Ball in box task

Figure 2-15. Examples of LfD with applications in robotics



(a) Movement of bulky loads

(b) Table-lifting task

Figure 2-16. Examples of collaborative robots using an LbD approach

In the field of surgical robotics, these LfD approaches have been used to collaborate with the surgeon and even to perform simple tasks during the surgery. (Reiley et al., 2010) proposed a robotic assistance system that used a generic LfD approach to generate smooth trajectories from expert demonstration. They used the da Vinci surgical system to record the performed surgical tasks, which were divided into several subtasks (called surgemes in their report). Each surgeme was trained several times, and the obtained expert human demonstration was temporally aligned using Dynamic Time Warping. Then, the motion model was encoded by Gaussian Mixture Models, and Gaussian Mixture Regression was used to generate the robot's trajectory. Finally, this trajectory was validated by a Hidden Markov Model trained with different surgeon skill levels (expert, intermediate and novice) and used to compare the quality of the generated trajectory to that of the surgeon skills. This workflow is illustrated in Figure 2-17. The experimental results were limited to simple trajectory generation without solving any concrete surgical tasks.

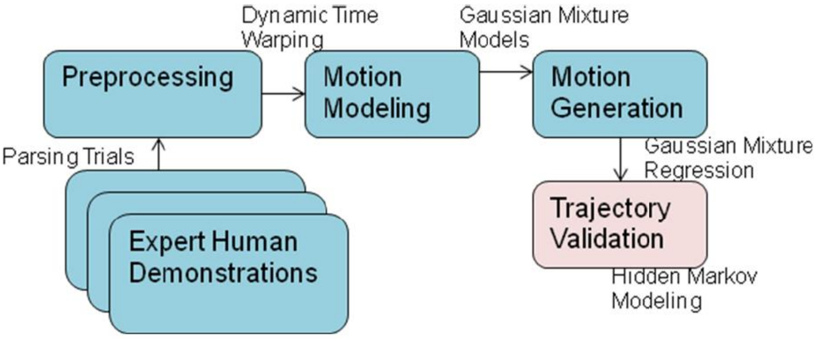


Figure 2-17. Workflow of the method proposed by (Reiley et al., 2010)

Furthermore, (van den Berg et al., 2010) proposed the use of LfD to allow surgical robotic assistants to execute specific tasks with superhuman performance in terms of speed and smoothness. Thus, they addressed this proposal by extending the previously stated LfD approach as follows: they used a Kalman smoother (Evensen, 2000) to improve the demonstration trajectories and a Linear-Quadratic Regulator (LQR) (Anderson and Moore, 1989) to execute the trajectory using a quadratic cost function to penalize deviations and non-smoothness in the execution. Using this approach, the Berkeley Surgical Robot was trained to tie a knot in a thread around a ring following the three-stage procedure shown in Figure 2-18. The results of this experiment demonstrated that the robot was able to successfully execute this task up to 7x faster than the demonstration.

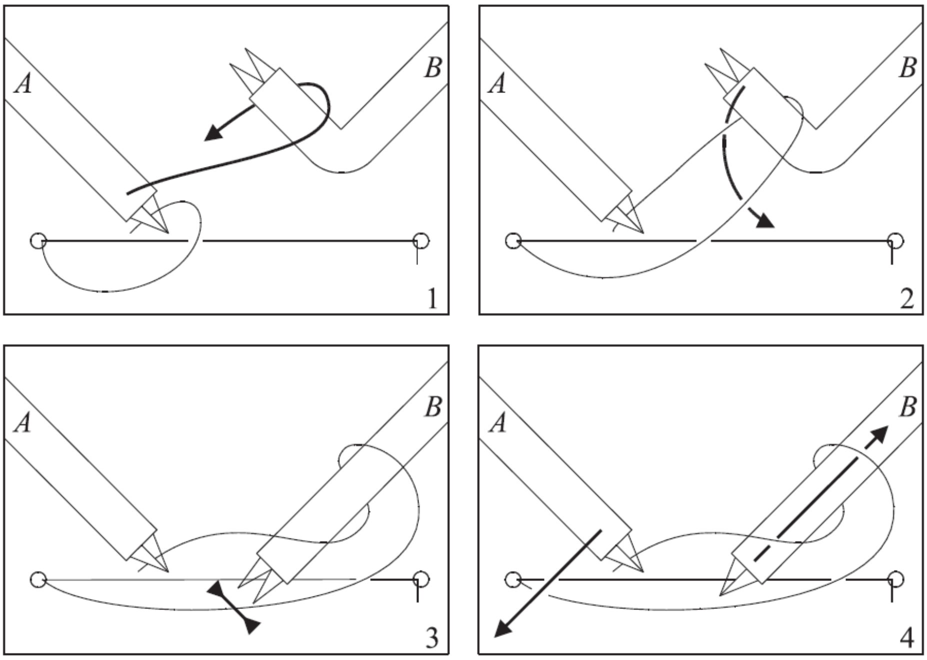


Figure 2-18.Three-stage knot-tying decomposition (van den Berg et al., 2010): first (1), robot loops the thread around the robot B gripper; second (2, 3), robot B grasps the thread and closes its gripper; and finally (4), both robots are moved away to tighten the knot.

Lastly, (Osa et al., 2014) proposed an extension of the previous approaches by adding Locally Weighted Regression (LWR), which was used to model the spatial variance in the trajectories over the initial conditions and allowed previously demonstrated trajectories to be generated regardless of the initial conditions. Using this approach, the task of bimanual looping of a surgical thread was tackled by changing the initial position of the manipulators in simulation and using an actual robotic surgical system.

Although these LfD approaches have demonstrated great potential in the use of surgical robots to assist in surgeries, they are still limited to simple tasks. Therefore, the use of autonomous robots in clinical applications is far from reality owing to the limitations of these types of approaches and the ethical issues that involve care of patients by surgeons. For example, it would be difficult to obtain approval for an automated surgical procedure performed by a robot from the ethical review board of any hospital or government (Senapati and Advincula, 2005). However, the use of artificial intelligence to guide a surgeon in performing a surgical task using the intelligence of humans and the accuracy of robots appears to be an intermediate solution that would be accepted by surgeons and ethical committees. Thus, haptic guidance arises as a novel methodology that uses force feedback to assist users in solving specific tasks by robotic teleoperation.

2.4 Haptic Guidance

In robotic teleoperation, a human operator performs a task using a master console, which controls a slave robot remotely. To allow efficient operation of the system, the operator needs to receive rich sensory information from the remote site. Despite several years of research on optimizing this feedback, teleoperation is still associated with a high workload for the operator owing to the lack of sensory information. Therefore, methods to provide additional (synthetic) support during task execution are being investigated. Haptic guidance was demonstrated to be a promising method to reduce operator workload and improve performance in teleoperated tasks (O’Malley et al., 2005; Boessenkool et al., 2013). Conventionally, it is implemented as a collaborative control strategy that improves and assists teleoperation by analysing the operator’s motions with respect to predefined trajectories or known forbidden regions and guiding these motions using a haptic device. This control strategy, which is commonly known as *Virtual Fixtures* or *Active Constraints* (Bowyer et al., 2014), has been used to solve specific tasks in surgical robotics by the following approaches.

The first contribution to haptic guidance with applications in surgical robotics was (Taylor et al., 1999). The authors described the development of a surgical robotic system with the goal of extending a human’s ability to perform tasks that require small-scale movements based on human judgement, sensory information and hand-eye coordination. Thus, they defined a novel approach, called *steady-hand micromanipulation*, which focused on hands-on micromanipulators that assisted the surgeon in providing smooth, tremor-free, precise position control and force scaling. The result of this contribution was a hands-on surgical robot with the accuracy and sensitivity of a machine and the manipulative transparency of hand-held surgical instruments. Once the first prototype of the robotic system was completed, they performed an experimental study comparing the ability of unassisted humans and *steady-hand micromanipulation* to insert a surgical needle into holes ranging from 150 to 250 μm in diameter. This study demonstrated that the *steady-hand* approach improved the success rate from 43% unassisted to 79% for the 150 μm holes and from 49% unassisted to 78% for the 250 μm holes (Kumar et al., 1999). Two years later, (Park et al., 2001) proposed the use of virtual fixtures in robot-assisted coronary artery bypass graft procedures to define forbidden regions. To do this, they used CT scans to locate forbidden regions relative to metal pins that were placed into the patient as reference frames. This approach was implemented in the ZEUS surgical system, and virtual walls were used to define the forbidden regions. During the teleoperation, the robot-handled instrument was free to follow the positional references provided by the surgeon using the master console outside the forbidden region. However, if the position reference from the master console was within the forbidden region, only lateral movements on the virtual wall surface were allowed. The primary objective of this approach was to reduce surgery time. To demonstrate this improvement, four subjects were trained to dissect a segment from a simulated tissue by avoiding penetrating the wall, which was marked with lines on the simulated tissue. The use of virtual fixtures on this task reduced the completion time by over 27% and avoided overruns beyond the desired region. Although there have been numerous contributions on this topic in recent years, only (Chowriappa et al., 2013) used Prediction from Expert Demonstration (PED) for haptic guidance in surgical robotics. In particular, they used this approach to solve the trocar placement in MIS. To do this, they collected a set of force, torque and trajectory data from multiple demonstrations of the task and encoded them into a GMM. Then, a generalization of this set of trajectories with its associate parameters was generated using GMR and was used to perform haptic guidance through *virtual fixtures*. Figure 2-19 illustrates the training trajectories: the green lines in (a), (d) and (g) represent the encoded GMM; the green ellipses are the same as in the previous graphs; and the profiles obtained by the GMR are presented in (b), (c), (e), (f), (h) and (i).

Once the trajectory profiles were obtained, the haptic guidance forces were provided by the master console taking into account the difference between the performed trajectory and the predicted trajectory using LfD as follows:

$$M_m = M_s + K \cdot P_n + b \cdot \dot{P}_n \qquad (2.1)$$

In this equation, a simple mass-damper system is used to provide assistance forces, where M_m is the force and torque reflection, M_s is the force and torque sensed by the slave robot, P_n is the difference between the estimated model prediction forces and torques p_n and the real exerted forces and torques by the slave robot p_s as $P_n = (p_n - p_s)$. K is the stiffness, and b is the damping constant.

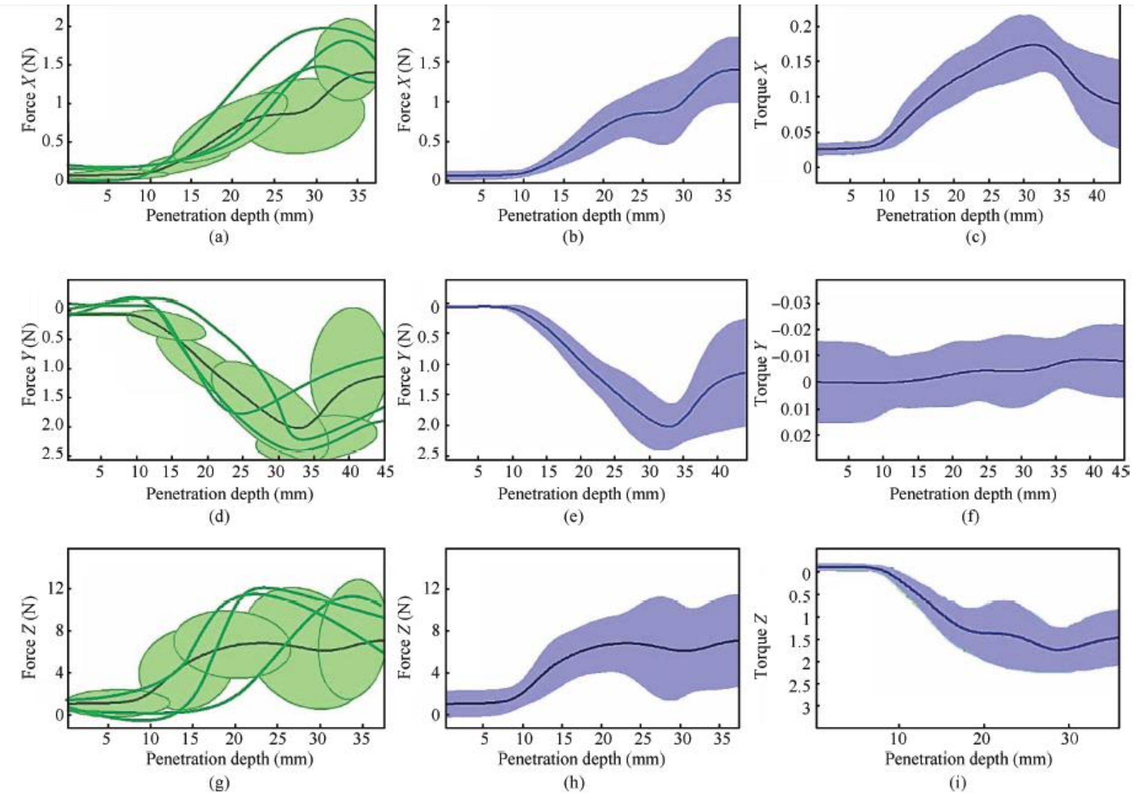


Figure 2-19. Force and torque trajectories (Chowriappa et al., 2013): (a), (d) and (g) represent the encoded GMM; and (b), (c), (e), (f), (h) and (i) represent the force-torque GMR generated profile

This approach was tested experimentally in two different scenarios: telesurgery, where information could be delayed or lost due to communication failures, and haptic guidance, where excessive force-torques could be avoided during the trocar insertion. The first experiment was conducted with different transmission error rates; although associated with low error rates (lower than 20%), the use of the PED approach and haptic feedback alone yielded similar performances during the execution of the task. However, in the case of error rates between 20% and 30%, the performance of the task improved when the PED approach was used, as indicated in Figure 2-20.

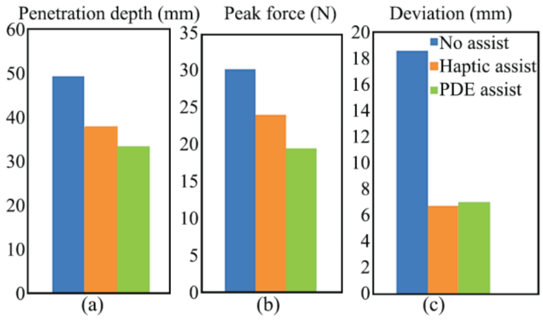


Figure 2-20. Task performance (Chowriappa et al., 2013): (a) Mean penetration depth; (b) trocar deviation; and (c) peak force

The second experiment was conducted to validate haptic guidance when the surgeon exceeded the predicted penetration depth during the insertion, the insertion trajectory deviated from the predicted model and the applied force-torque was different from the predicted one. To perform this experiment, three groups of 5 participants were created to perform three trials and compare the results. The first group performed the task with neither haptic guidance nor haptic feedback, the second one used only haptic feedback, and the last one used both haptic guidance and haptic feedback. Overall, the last group performed the task better and more consistently than the other groups in terms of maximum penetration depth and trajectory deviation.

In conclusion, although there have been numerous studies related to haptic guidance, they are primarily focused on virtual fixtures, which means that the guidance is performed using predefined trajectories or regions. However, the use of approaches such as LfD could lead to the use of haptic guidance by previous demonstrations based on the relationship between the sensor measurements (e.g., F/T sensors, velocity sensor, etc.) and the expert demonstration.

2.5 Summary

As demonstrated in this chapter, research on surgical robotics has evolved into two primary topics: the design of new robotic platforms that allow surgeons to perform MIS by new techniques, such as SPAS, which improve patient outcomes by reducing the number of scars and the duration of the hospital stay, and the use of Machine Learning Algorithms, which allow robots to collaborate or assist surgeons using previously trained information.

However, more progress is needed to place smart surgical robots into operating rooms on a large scale. Thus, the use of lightweight manipulators with generic kinematic structures would reduce the robot dimensions, thus allowing teleoperated SPAS to be performed. Nevertheless, using generic kinematic structures, the estimation of the fulcrum point (virtual RCM) becomes a problem for which, even though it has been solved in different ways, there is currently no definitive solution.

Furthermore, it is desirable that these teleoperated surgical robotic systems assist or collaborate with the surgeon in an intelligent way. Thus, Machine Learning appears to be a useful tool for retrieving knowledge from surgeons to create recognition systems or learning models that can be used to provide intelligence for robotic assistance. Therefore, haptic guidance emerges as a promising method of assistance that would reduce surgeons' workload and improve the performance of teleoperated surgical tasks. However, this methodology has not yet been sufficiently applied to surgical robotics.

3 Smart Navigation for SPAS

3.1 Introduction

In most laparoscopic surgery procedures, surgical instruments are inserted through small incisions in the abdomen. Because of this, the working environment is constrained when rigid instruments are used. When manipulators are handling these instruments, their movements are restricted by the entry port, which is commonly known as the fulcrum point because the instrument must pivot around it to avoid undesirable forces on the abdominal wall. Therefore, an instrument with no distal joints (no articulated tip) is limited to four DoF: three rotational (Figure 3-1.a, b and c) and one translational (Figure 3-1.d). Thus, such instruments cannot be moved freely to any position and orientation inside the patient’s abdomen. Therefore, surgical instruments with distal joints are used, and the fulcrum point constraint is solved by a remote centre of motion (RCM) that coincides with the fulcrum point.

The RCM can be implemented in different ways. One of them is the use of a mechanical RCM, i.e., the robotic system arms are designed with a specific kinematic structure that allows them to perform movements based on a mechanical fixed RCM. Accordingly, the surgeon has to match the robot RCM with the fulcrum point, which remains invariant during the surgery. This method has demonstrated its utility in clinical applications because of its robustness, e.g., a fault in the system will not damage the patient. In fact, the most used surgical systems in clinical applications, the da Vinci Surgical System and Zeus, implement this method. However, these kinematic structures are larger than generic structures, which are composed of lightweight arms that are useful for performing surgeries by SPAS and avoiding collisions.

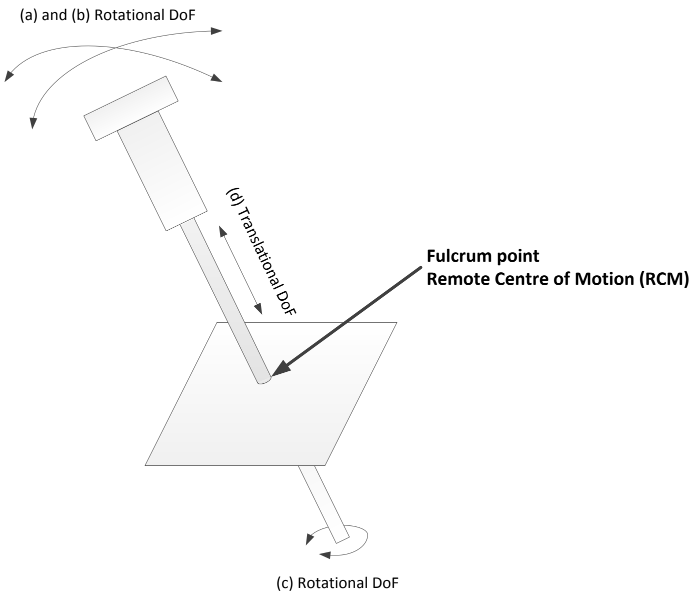


Figure 3-1. Four DoF workspace constraint of an inserted surgical instrument through a trocar

However, when generic kinematic structures are used, a virtual RCM is commonly implemented (Locke et al., 2007). The virtual RCM consists of estimating the fulcrum point position to perform the movements of the manipulator pivoting around it. This estimation has been typically performed using the measured force-torques throughout the surgical instrument, which allows estimation of the fulcrum point when the instrument is inserted into the patient's abdomen.

Nevertheless, if only these sensors were used to estimate the fulcrum point, the estimation could be wrong when the instrument interacts with the soft tissue inside the abdomen because the sensor measures not only the interaction forces between the instrument and the abdomen but also the interaction forces of the instrument tip with the soft tissue. Therefore, a method is required that can determine whether the instrument tip is interacting with soft tissue.

In SPAS, all of the instruments are inserted through a multiport trocar. Taking advantage of this aspect, the RCM estimation can be improved using redundant information provided by the F/T sensors from the manipulators that handle each inserted instrument.

Consequently, a navigation method for surgical teleoperation in SPAS is proposed in this chapter. This method has been designed for the haptic teleoperation of surgical robots with generic kinematic structures, where the fulcrum point is estimated using F/T sensors placed on the manipulator end effectors. Additionally, a soft tissue interaction detector has been included to detect when there is interaction with soft tissue. Using this information and a measurement fusion method, the accuracy of the RCM estimation has been improved.

Therefore, this chapter introduces the previous contributions and the problems that are solved in this thesis. Then, the interaction forces during the teleoperation of surgical instruments are stated. Using them, the proposed navigation method is described based on two manipulators that handle instruments inserted through a multiport trocar. Finally, a soft tissue interaction detector is described.

3.2 Previous Work

This section introduces the previous contributions related to surgical instrument navigation using a virtual RCM, which have been considered to be a starting point to develop a new smart navigation method that solves the previously stated issues. (Krupa et al., 2004) presented a contribution in which they addressed the virtual RCM for generic kinematic structures. In particular, they used a 6 DoF robot with a generic kinematic structure and an F/T sensor placed between the manipulator end effector and the instrument to perform a proportional force feedback control and to estimate the fulcrum point. This experimental setup is shown in Figure 3-2.

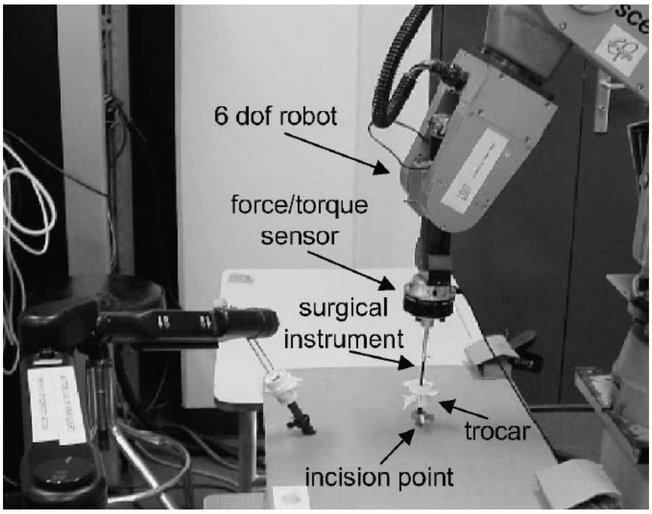


Figure 3-2. Experimental setup (Krupa et al., 2004)

The objective of the proposed force feedback control was to minimize undesirable forces exerted by the instrument on the abdomen during execution of the movements. Their experimental results demonstrated that this controller reduced the forces exerted on the abdomen even if the fulcrum point was not correctly estimated. Figure 3-3 illustrates the exerted forces during the movement of an instrument inserted through a trocar with three different estimation errors. In this experiment, the fulcrum point was estimated using the distance between the F/T sensor and the real fulcrum point. Thus, d is the distance from the F/T sensor to the real fulcrum point, and $\hat{d}$ is the distance from the F/T sensor to the estimated fulcrum point. Therefore, the error between the real and estimated fulcrum points can be represented by $\hat{d} - d$. The first configuration (first row of Figure 3-3) used $\hat{d} = 0.15\ m$ and $d = 0.2\ m$, and, as shown, the maximum measured forces on the sensor were approximately 2 N for F_x and F_y. In the second configuration (second row), these values were changed to $\hat{d} = 0.3\ m$ and $d = 0.1\ m$ with maximum obtained forces of $F_x \approx 4\ N$ and $F_y \approx 5\ N$. Finally, the third configuration was $\hat{d} = 0.02\ m$ and $d = 0.2\ m$, where the maximum obtained forces in this configuration were $F_x \approx 5\ N$ and $F_y \approx 4\ N$.

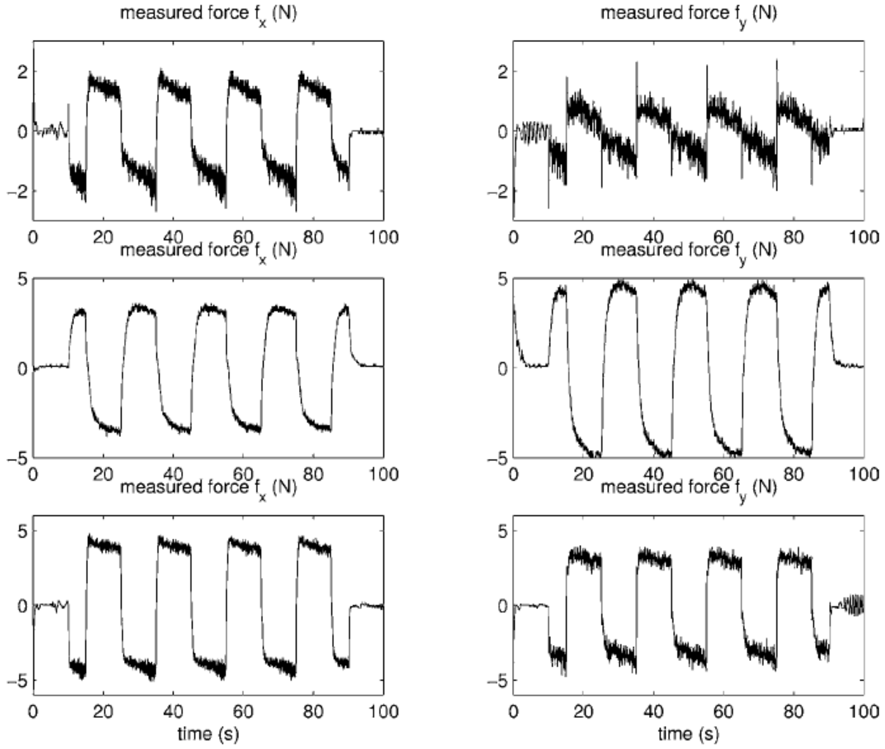


Figure 3-3. Proportional force feedback control experiment (Krupa et al., 2004)

The estimation of the fulcrum point was performed using the force and torque measurements provided by an F/T sensor placed on the manipulator end effector, which allows estimation of the distance between the F/T sensor and the fulcrum point as follows:

$$\hat{d} = \frac{\sqrt{{T_x}^2 + {T_y}^2}}{\sqrt{{F_x}^2 + {F_y}^2}} \tag{3.1}$$

where T_x and T_y represent the measured torques on the F/T sensor. To improve the accuracy of the estimation, the authors used a weighted least-squares algorithm with a sliding window and a dead-zone. Using this algorithm, they performed two experiments in which the forces exerted on the abdomen and the estimation error were analysed. Figure 3-4 shows the results. The left column represents an experiment in which the initial estimation was $\hat{d}_0 = 0.3\ m$, and the real fulcrum point distance was $d = 0.1\ m$. The right column parameters are $\hat{d}_0 = 0.02\ m$ and $d = 0.2\ cm$. It can be observed that the estimation converges as soon as the instrument starts to move, and the maximum peak forces are approximately 2 N.

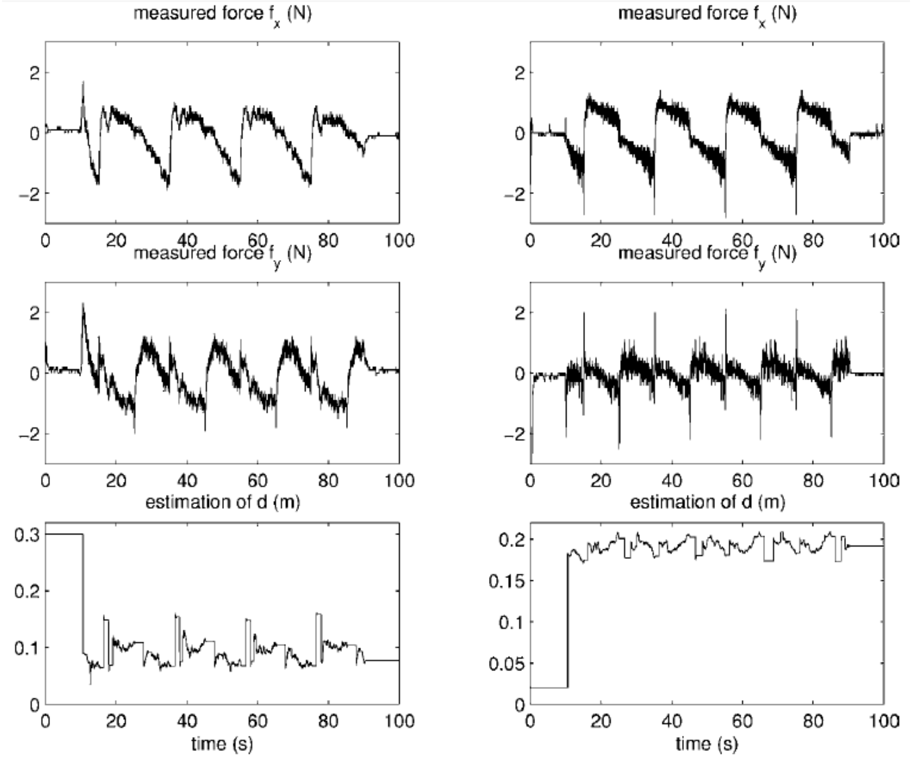
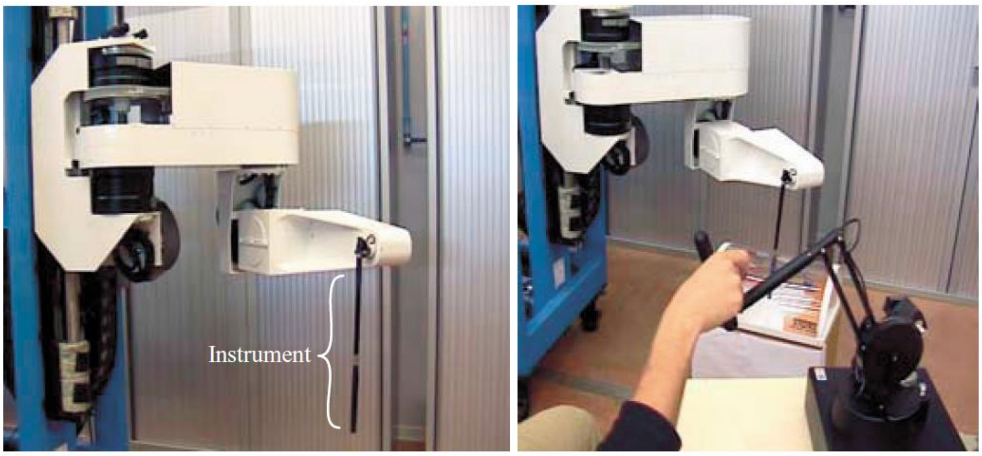


Figure 3-4. Fulcrum point estimation experiment (Krupa et al., 2004)

This contribution shows how an F/T sensor that is placed on the manipulator end effector can be used to estimate the fulcrum point and perform force control to minimize the forces on the abdomen. A similar method has been used in a recent patent to control the movements of surgical instruments using manipulators (Ruiz and Correcher, 2012).

(Michelin et al., 2004; Michelin et al., 2006) proposed the use of joint torques to estimate the fulcrum point using a dynamic task/posture decoupling control algorithm. Thus, the contact forces applied to the trocar were minimized, hence guaranteeing that the fulcrum point constraint was satisfied. They tested this algorithm in a generic kinematic 5 DoF robot (Figure 3-5.a) that was teleoperated by a Phantom master device (Figure 3-5.b).

The experimental results demonstrated that this method was also suitable for estimating the fulcrum point. Figure 3-6 depicts the results for a 20 cm lateral straight-line path. It can be observed that the maximum error between the estimated fulcrum point and the real point was approximately 1.25 cm. However, the forces exerted on the patient’s abdominal wall were not addressed in this study.



(a) (b)

Figure 3-5. Experimental setup (Michelin et al., 2004): (a) D2M2 slave robot; and (b) Phantom master device

(Cortesao et al., 2006) proposed a null space control scheme with Active Observers (AOBs) to estimate the fulcrum point in a similar way to that used by the previously described contribution. This approach was implemented in the same manipulator as the previous contribution (D2M2) and used the same experimental setup as illustrated in Figure 3-5. The use of AOBs improved the fulcrum point estimation by reducing this error (d_{tr}) to below 8 mm, as shown in Figure 3-7.

The DLR MiroSurge addresses this issue by using a position and force control method based on integrated joint torque sensors, similar to the studies described above (Hagn et al., 2009).

More recent studies refer to the RCM as a fixed point without proposing any estimation method. In particular, (Chawaphol et al., 2011) developed a new robot for laparoscopic surgery using a parallelogram mechanism that performed constrained movements with respect to the RCM. To adjust the RCM with the fulcrum point, they used two laser pointers for visual alignment, i.e., when both laser dots, which are projected onto the patient’s skin, converge towards one dot, the RCM is aligned with the abdominal wall. Figure 3-8 illustrates this effect.

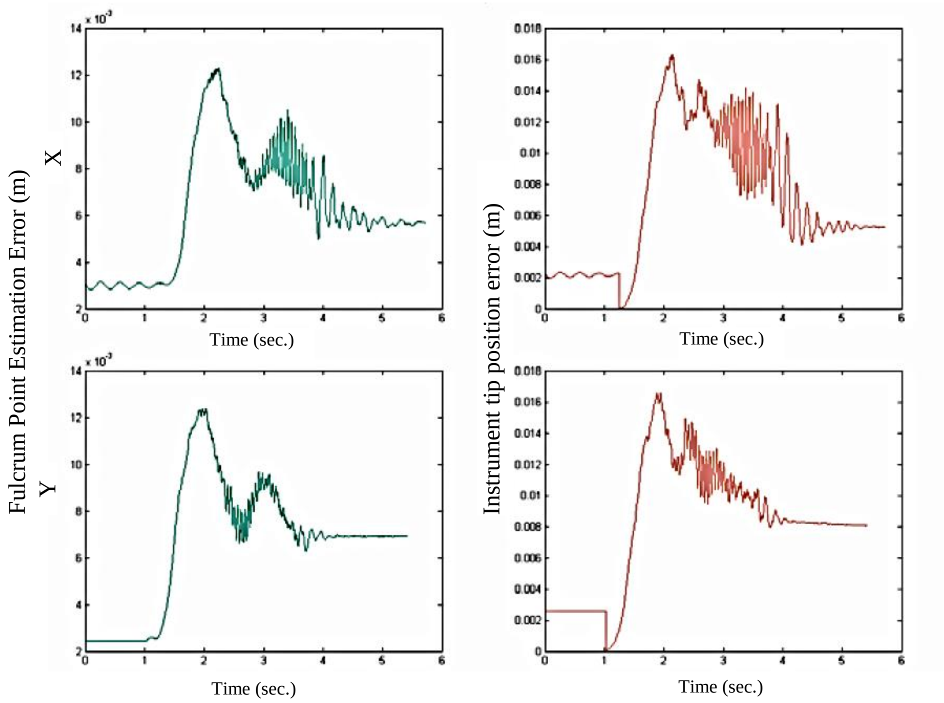


Figure 3-6. Joint torque fulcrum point estimation experiment (Michelin et al., 2006). Left column: Distance between the estimated fulcrum point and the actual one; and right column: Tracking error at the instrument tip

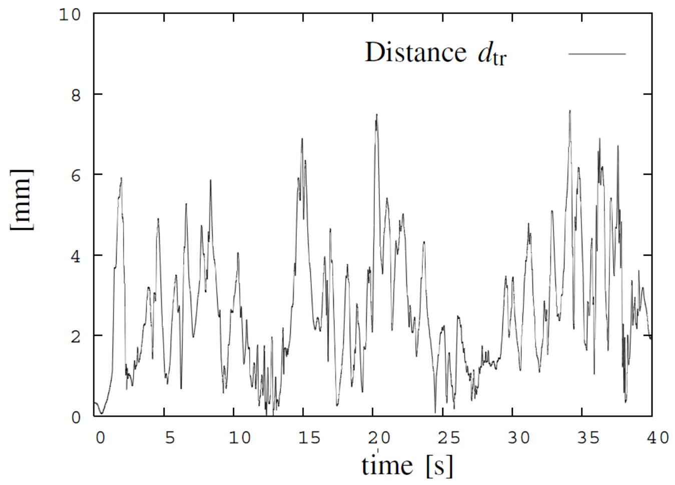


Figure 3-7. AOBs experimental results (Cortesao et al., 2006)

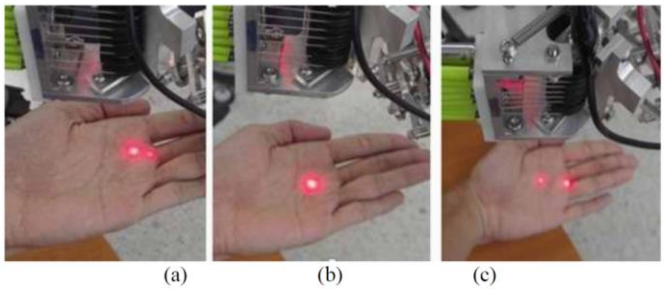


Figure 3-8. Laser pointers for RCM alignment in (Chawaphol et al., 2011): (a) the RCM is closer to the fixed RCM using the robot; (b) the robot RCM is aligned with the hand; and (c) the RCM is too far from the robot

In another study, (Dalvand and Shirinzadeh, 2012; Dalvand and Shirinzadeh, 2013) proposed a navigation system for milli/micro-manipulations under the constraint of the RCM using a generic kinematic structure with 10 DoF. Figure 3-9 depicts the developed robot, which was used to perform different experiments. The results of this study showed that the robot was able to perform precise movements under the RCM constraint; however, the RCM was programmatically fixed, i.e., it was not estimated during the experiments.

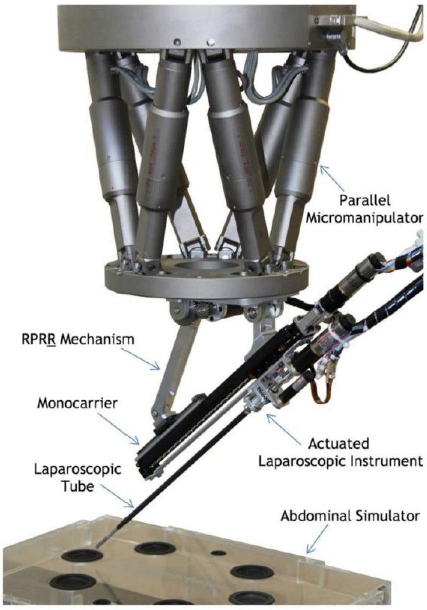


Figure 3-9. Proposed parallel robot for minimally invasive surgery (Dalvand and Shirinzadeh, 2013)

Therefore, it is necessary to achieve estimation of the fulcrum point, which coincides with the robot’s RCM. Previous contributions have demonstrated the feasibility of using force and torque information to estimate the fulcrum point and to minimize the forces exerted on the patient’s abdomen. However, in these contributions, it is assumed that there is no interaction of the instrument tip with the soft tissue inside the abdomen, which means that this method cannot be used when this interaction occurs. However, they do not propose a method for detecting this interaction during the surgery. Furthermore, when several instruments are introduced by the same trocar, the fulcrum point estimation could be improved using redundant information from the estimation of the fulcrum point using several F/T sensors. These are the issues that will be addressed in this chapter.

3.3 Force Interaction

This section presents the interaction forces and torques in an instrument that is handled by a manipulator and has been inserted into a patient's abdomen. First, all of the used reference frames are described to show the interaction forces during the movement of the instrument. Then, the interaction force equations are obtained, which have been used to estimate the fulcrum point and analyse the interaction of the instruments with the soft tissue inside the abdomen.

Figure 3-10 illustrates the instrument movement from an initial point *A* to a final point *B*, where $\{O\}$ is the orthonormal reference frame, which corresponds to the base of the manipulator platform, and $\{H_A\}$ and $\{H_B\}$ are the manipulator end effector positions in *A* and *B*, respectively. The instrument is attached to the end effector such that the Z-axes of $\{H_A\}$ and $\{H_B\}$ coincide with the longitudinal axis of the instrument. Moreover, a sensor is placed between the end effector and the instrument to provide force and torque measurements. Finally, $\{P_A\}$ and $\{P_B\}$ represent the position and orientation of the instrument tips, whose orientations coincide with $\{H_A\}$ and $\{H_B\}$, respectively. These frames are represented by a tuple (3.2) that contains the Cartesian position vector ${}^{j}\vec{p}_i$ and orientation ${}^{j}\sigma_i$ of the reference frame i based on the reference frame j. Moreover, these frames can be represented by the homogeneous transformation matrix ${}^{j}T_i$ as follows:

$$\begin{gathered} {}^{j}T_i = \left({}^{j}\vec{p}_i,\ {}^{j}\sigma_i \right); \{i\} = {}^{O}T_i \\ i, j \in \{O, H_A, H_B, P_A, P_B, F_A, F_B, F'_A, F'_B\} \\ {}^{j}\vec{p}_i = \left({}^{j}x_i,\ {}^{j}y_i,\ {}^{j}z_i \right);\ {}^{j}\sigma_i = \left({}^{j}\alpha_i,\ {}^{j}\beta_i,\ {}^{j}\gamma_i \right) \end{gathered} \tag{3.2}$$

At the initial point *A*, the estimated fulcrum location is $\{F'_A\}$, which is different from the real point $\{F_A\}$ because of estimation errors. It should be noted that each reference frame placed along the instrument has the same orientation. If a movement from *A* to *B* were conducted, undesirable forces $\vec{f}_{F_B}$ would be exerted on the patient's abdomen owing to the estimation error. It can be observed that this movement causes a displacement from the real fulcrum point $\{F_A\}$ to $\{F_B\}$, and the distance between them is represented by $\overrightarrow{r_F}$. If the abdomen is modelled as a linear spring (Huang et al., 2007), $\overrightarrow{f_{F_B}}$ represents the lateral exerted forces from $\{F_B\}$ that are calculated by (3.3), where K_a is the skin elasticity constant, which depends on the patient's skin.

$$\overrightarrow{f_{F_B}} = K_a \times \overrightarrow{r_F} \tag{3.3}$$

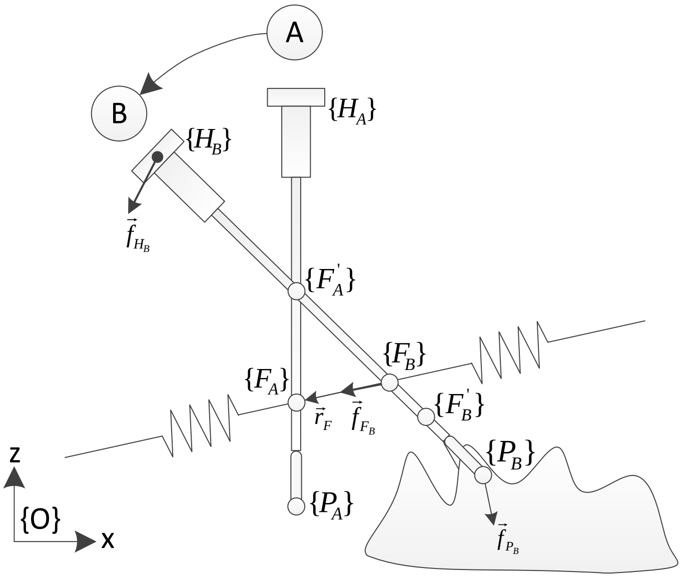


Figure 3-10. Movement of the instrument from A to B

Moreover, when there is an interaction with the soft tissue, additional forces $\overrightarrow{f_{P_B}}$ are exerted on the tip of the instrument. Assuming that the flexibility of the instrument is negligible, the sum of $\overrightarrow{f_{F_B}}$ and $\overrightarrow{f_{P_B}}$ is transmitted to the end effector $\{H_B\}$; thus, (3.4) is obtained, where $\overrightarrow{f_{H_B}}$ can be measured using a F/T sensor placed on $\{H_B\}$.

$$\overrightarrow{f_{H_B}} = \overrightarrow{f_{F_B}} + \overrightarrow{f_{P_B}} \tag{3.4}$$

Additionally, because of this movement, a torque $\overrightarrow{\tau_{H_B}}$ is generated (3.5) that can be measured using the F/T sensor. In this equation, ${}^{H_B}\vec{p}_{F'_B}$ represents the distance between the position of the manipulator end effector and the position of the new estimation of the fulcrum point $\{F'_B\}$.

$$\overrightarrow{\tau_{H_B}} = \overrightarrow{f_{H_B}} \times {}^{H_B}\vec{p}_{F'_B} = \overrightarrow{f_{F_B}} \times {}^{H_B}\vec{p}_{F_B} + \overrightarrow{f_{P_B}} \times {}^{H_B}\vec{p}_{P_B} \tag{3.5}$$

Taking into account that $\{P_B\}$, $\{F_B\}$ and $\{F'_B\}$ are placed along the Z-axis of $\{H_B\}$, the distance vectors along the instrument ${}^{H_B}\vec{p}_{F'_B}$, ${}^{H_B}\vec{p}_{F_B}$ and ${}^{H_B}\vec{p}_{P_B}$ have magnitudes only on the Z-axis. Therefore, $\left|{}^{H_B}\vec{p}_{F'_B}\right|$ can be defined as follows:

$$\left|{}^{H_B}\vec{p}_{F'_B}\right| = \frac{\left|\overrightarrow{f_{F_B}}\right|}{\left|\overrightarrow{f_{H_B}}\right|} \cdot \left|{}^{H_B}\vec{p}_{F_B}\right| + \frac{\left|\overrightarrow{f_{P_B}}\right|}{\left|\overrightarrow{f_{H_B}}\right|} \cdot \left|{}^{H_B}\vec{p}_{P_B}\right| \tag{3.6}$$

where the Z component of the force vectors has not been considered because this component is aligned with the distance vectors. Using (3.6), the estimation error can be represented as

$\left|{}^{F'_B}\vec{p}_{F_B}\right|$. Figure 3-11 illustrates an example of this error, where the instrument length $\left|{}^{H_B}\vec{p}_{P_B}\right|$ is 240 mm, the real fulcrum point distance $\left|{}^{H_B}\vec{p}_{F_B}\right|$ is 120 mm, the exerted forces on the fulcrum point $\left|\overrightarrow{f_{F_B}}\right|$ are 1.5 N, and the exerted instrument tip forces $\left|\overrightarrow{f_{P_B}}\right|$ depend on different surgeon tasks, i.e., between approximately zero during thread handling and greater than 10 N for tissue grasping (Brown et al., 2004; Richards et al., 2000). It can be observed that the estimation error increases with the forces on the instrument tip. Owing to this effect, the instrument movements will not be performed correctly. In fact, this example indicates that a force of 2 N on the tip generates a fulcrum point estimation error of 80 mm. Thus, a method to detect soft tissue interaction is needed to reduce the estimation error.

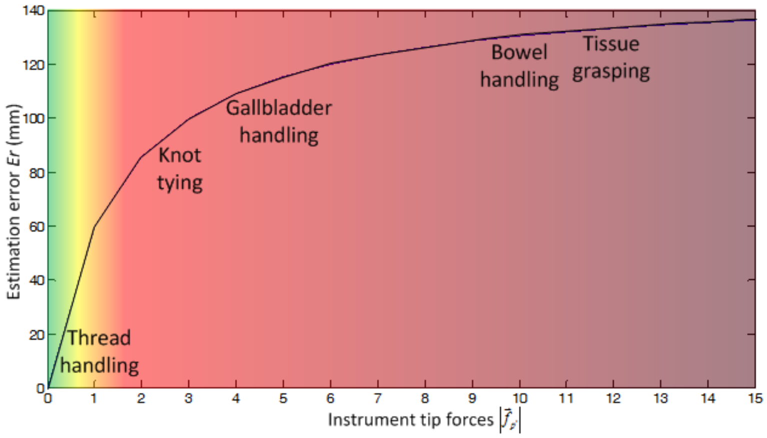


Figure 3-11. Example of estimation errors for different surgical tasks

Finally, by analysing (3.6), the conditions in (3.7) could be used to detect when an interaction occurs with the soft tissue. However, this detection depends on the difference between $\left|\overrightarrow{f_{F_B}}\right|$ and $\left|\overrightarrow{f_{P_B}}\right|$. In the case of a small difference, i.e., the fulcrum point is close to the instrument tip or there are low abdominal and instrument tip forces, an interaction with the soft tissue would not be detected. Moreover, a variation in $\left|{}^{H_B}\vec{p}_{F'_B}\right|$ would not provide information on an interaction with tissue when forces on the tip are extremely low, e.g., during thread handling. Therefore, a method that is not only based on force and torque measurements may be used to detect this situation. The proposed method will be explained in Section 3.5.

$$\begin{gathered}\left|\overrightarrow{f_{F_B}}\right| \gg \left|\overrightarrow{f_{P_B}}\right| \rightarrow \left|{}^{H_B}\vec{p}_{F'_B}\right| \approx \left|{}^{H_B}\vec{p}_{F_B}\right| \\ \left|\overrightarrow{f_{P_B}}\right| \gg \left|\overrightarrow{f_{F_B}}\right| \rightarrow \left|{}^{H_B}\vec{p}_{F'_B}\right| \approx \left|{}^{H_B}\vec{p}_{P_B}\right|\end{gathered} \qquad (3.7)$$

When there is no interaction with the soft tissue ($\left|\overrightarrow{f_{P_B}}\right| \approx 0$), the distance between the end effector and the fulcrum point $\left|{}^{H_B}\vec{p}_{F_B}\right|$ can be estimated by (3.8); therefore, using (3.9), $\{F_B\} \approx \{F'_B\}$ is obtained.

$$\left|{}^{H_B}\vec{p}_{F_B}\right| \approx \left|{}^{H_B}\vec{p}_{F'_B}\right| = \frac{\left|\overrightarrow{\tau_{H_B}}\right|}{\left|\overrightarrow{f_{H_B}}\right|} \tag{3.8}$$

$${}^{O}T_{F_B} = {}^{O}T_{H_B} \cdot \begin{bmatrix} I & {}^{H_B}\vec{p}_{F'_B} \\ 0 & 1 \end{bmatrix} \tag{3.9}$$

Thus, the estimation of the fulcrum point when there is no interaction with the soft tissue inside the abdomen can be performed using equations (3.8) and (3.9). Considering that a force $\left|\overrightarrow{f_{H'}}\right| \approx 0$ causes $\left|{}^{H_B}\vec{p}_{F_B}\right| \to \infty$, a dead zone may be defined around zero to avoid this behaviour when these equations are used to estimate the fulcrum point position. This dead zone is fixed depending on the precision of the F/T sensor when it measures lower forces. Moreover, taking advantage of the fact that both manipulators are inserted through the same trocar (in SPAS), the fulcrum point could be estimated by neither, one or both, as stated in the next section.

3.4 Navigation Method

Once the estimation of the fulcrum point has been defined, a navigation method is proposed that modifies the orientation of the instrument to minimize abdominal forces resulting from estimation errors. This method is based on a parallel force-position control scheme that is represented in Figure 3-12.

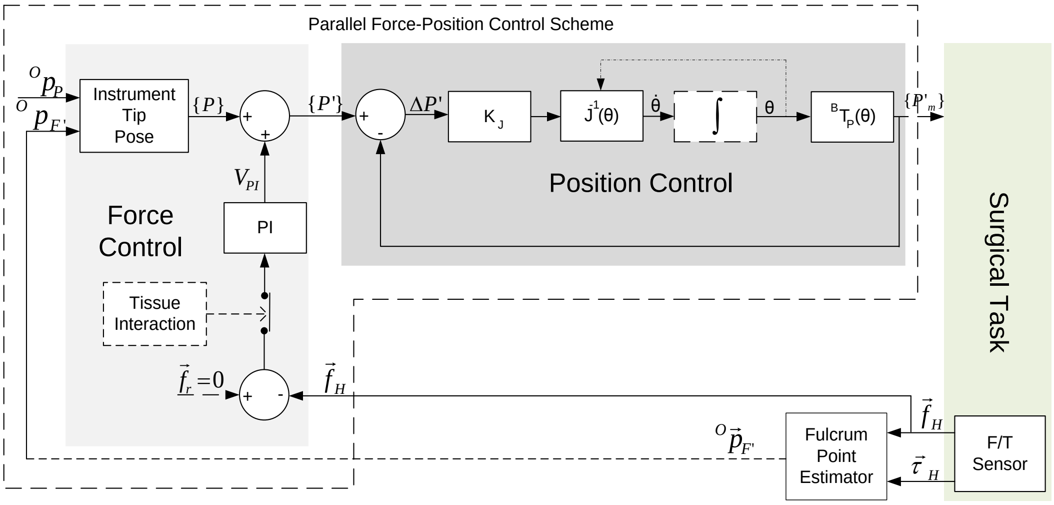


Figure 3-12. Jacobian-based parallel Force-Position control scheme

In this figure, the position reference of the instrument tip ${}^{O}\vec{p}_{P_{ref}}$ and the estimated fulcrum point position ${}^{O}\vec{p}_{F'}$ are used to obtain the instrument tip pose, whose orientation is calculated using an Euler ZYZ orientation reference ${}^{O}\sigma_{P_{ref}} = ({}^{O}\alpha_{P_{ref}}, {}^{O}\beta_{P_{ref}}, {}^{O}\gamma_{P_{ref}})$ and can align the instrument tip orientation with the estimated fulcrum point as follows:

$$ {}^{O}\alpha_{P_{ref}} = atan2\left[\left({}^{O}y_{P_{ref}} - {}^{O}y_{F'}\right), \left({}^{O}x_{P_{ref}} - {}^{O}x_{F'}\right)\right] \tag{3.10} $$

$$ {}^{O}\beta_{P_{ref}} = atan2\left[\sqrt{\left({}^{O}x_{P_{ref}} - {}^{O}x_{F'}\right)^2 + \left({}^{O}y_{P_{ref}} - {}^{O}y_{F'}\right)^2}, \left({}^{O}z_{P_{ref}} - {}^{O}z_{F'}\right)\right] \tag{3.11} $$

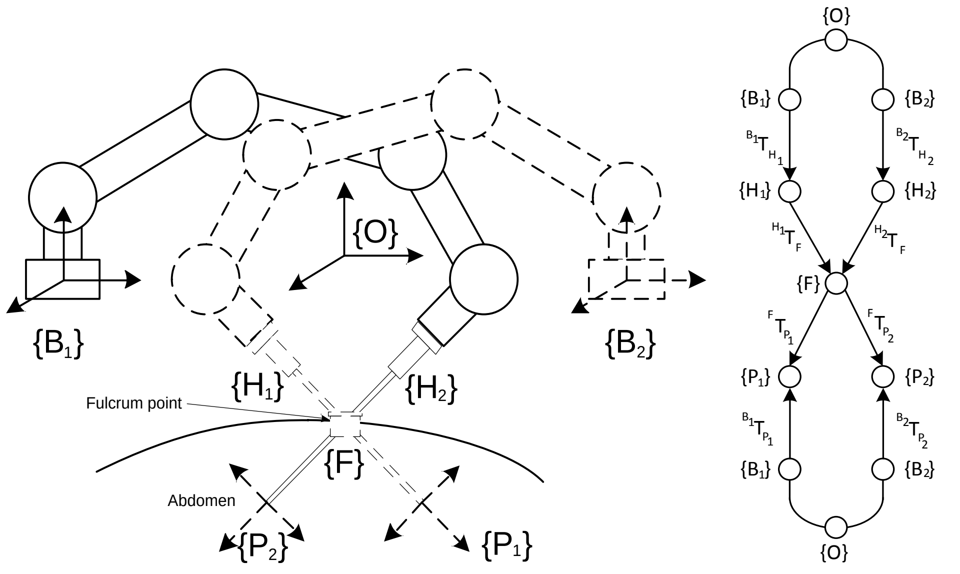
$$ {}^{O}\gamma_{P_{ref}} = -\,{}^{O}\alpha_{P_{ref}} \tag{3.12} $$

When there is no interaction with the soft tissue, a Proportional-Integrative controller, whose input is the difference between the abdominal force reference $\overrightarrow{f_r} = 0N$ and the F/T sensor measured forces $\overrightarrow{f_H}$, is used to modify the instrument orientation and thereby to minimize the forces exerted on the abdomen. This controller modifies the reference to $\{P'\} = ({}^{O}\vec{p}_{P_{ref}}, {}^{O}\sigma_{P_{ref}} + V_{PI})$, where V_{PI} is the force controller output. Using this reference, the position control scheme can provide the joint velocities $\dot{\theta}$ and positions θ based on the K_J matrix that fixes the position control dynamic using a first-order system with a time constant $\tau = 1/K_J$ as follows:

$$ \begin{gathered} \Delta\dot{P}' = -K_J \cdot \Delta P' + \{P'\} \\ \{P'_M\} = K_J \cdot \Delta P' \end{gathered} \tag{3.13} $$

This control scheme is based on the inverse Jacobian and the forward kinematic function. Owing to the nonlinearity of these functions, its stability is demonstrated in Appendix A.

SPAS consists of introducing two or more surgical instruments through the same trocar. Therefore, the proposed navigation method has been extended to consider two manipulators, as indicated in Figure 3-13.a. In this figure, $\{B_1\}$ and $\{B_2\}$ are the base reference frame of each manipulator, $\{H_1\}$ and $\{H_2\}$ are the end effector reference frames, $\{F\}$ is the common fulcrum point reference frame for both manipulators, and $\{P_1\}$ and $\{P_2\}$ are the instruments tip reference frames. The relationships between each frame are represented in Figure 3-13.b.



(a) (b)

Figure 3-13. (a) SPAS scenario with reference frames; and (b) Kinematic model graph, which represents the relation between the different frames

Figure 3-14 shows the proposed navigation method for the two manipulators that handle the surgical instruments in SPAS. In this figure, each manipulator is teleoperated by a *Haptic Guidance* system (explained in Chapter 4) that provides the instrument tip position in reference to a *Parallel Force-Position Control Scheme*, as indicated in Figure 3-12. This control scheme is used to move both manipulators, whose sensors provide a tuple for each instant *j* in time and can be defined as follows:

$$\phi_m = \left({}^{O}p_P, {}^{O}\sigma_P, {}^{O}\dot{p}_P, {}^{O}\dot{\sigma}_P, \vec{f}, \vec{\tau} \right), m = 1,2 \tag{3.14}$$

where m refers to each manipulator; ${}^{O}p_P$ is the position of the instrument tip; ${}^{O}\sigma_P$ represents the manipulator orientation; ${}^{O}\dot{p}_P$ represents the Cartesian velocity of the instrument tip; ${}^{O}\dot{\sigma}_P$ is the angular velocity of the instrument tip; $\vec{f} = (f_x, f_y, f_z)$ are the measured forces on the manipulator end effector $\{H\}$; and $\vec{\tau} = (\tau_x, \tau_y, \tau_z)$ represents the measured torques on the same location. This tuple includes the information necessary to estimate the fulcrum point position by (3.8) and (3.9) using a Least Square estimator (*Fulcrum Point Estimator* in Figure 3-14). Moreover, this information is used in the *Tissue Interaction Detector*, which is explained in the next section. The fulcrum point estimation has been improved by considering the interaction of each manipulator with the soft tissue (*Interaction Cases*), as indicated in Table 3.1, and a *Measurements Fusion Method*, which takes advantage of the fact that both manipulators are inserted through the same trocar.

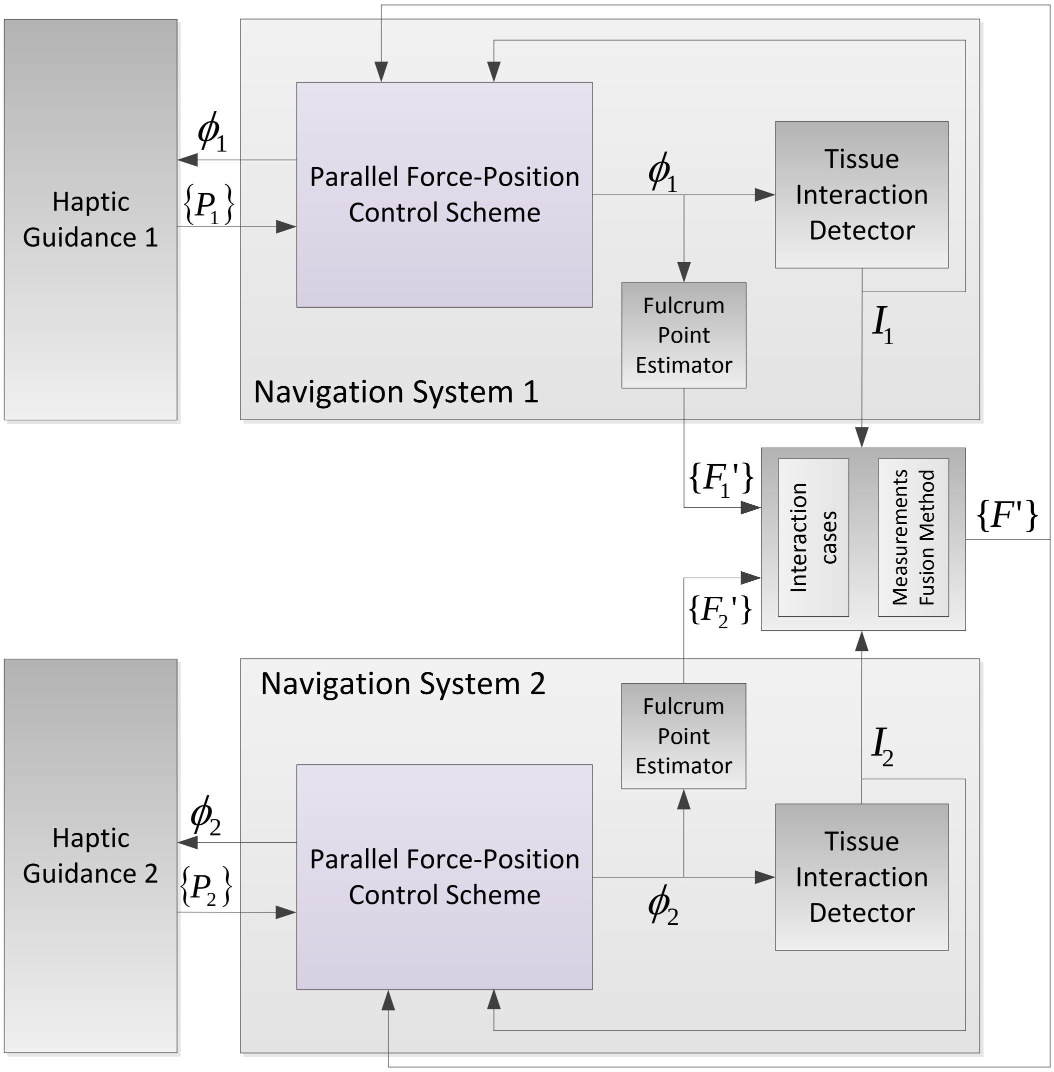


Figure 3-14. Extended navigation method

Depending on the interaction with the soft tissue, four interaction cases have been defined (Table 3.1). When both manipulators interact with the soft tissue, the fulcrum point cannot be estimated, and the previous estimation is used. Additionally, force control cannot be performed because the forces from the sensor are composed of the forces exerted on the tip, and they are not useful for compensating the fulcrum point estimation errors. In this situation, haptic feedback is provided by both manipulators to allow the surgeon to feel the interaction. However, when only one manipulator interacts with the soft tissue, only information from the other manipulator is used to estimate the fulcrum point. Furthermore, force control can be performed on the manipulator that is not interacting with the tissue, and haptic feedback is provided only for the manipulator that is interacting with soft tissue.

Fulcrum point estimation is performed for each manipulator, as previously mentioned. However, taking advantage of both instruments being inserted by the same trocar without soft tissue interaction, a Kalman filter measurement fusion method (Gan and Harris, 2001) has been implemented to improve its accuracy.

Table 3.1. Interaction Cases

Interaction with Soft Tissue		**Fulcrum Point Estimation**	**Force Control**	**Haptic Feedback**
Manip. 1	**Manip. 2**			
Yes	Yes	No estimation	No	Both
No	Yes	Manip. 1	Only Manip. 1	Only Manip. 2
Yes	No	Manip. 2	Only Manip. 2	Only Manip. 1
No	No	Measurement fusion method	Yes	No

The fulcrum point estimation for each manipulator is modelled using a discrete time state-space model in (3.15) and (3.16), where k represents the discrete-time index, and I is a 3x3 identity matrix. The state vector that represents the space coordinates for the estimated fulcrum is ${}^{O}p_F$, the fulcrum point space coordinates of each manipulator is defined by $\left[{}^{O}p_{F_1'} \quad {}^{O}p_{F_2'}\right]^T$, and w_k and v_k are zero-mean white Gaussian noise with covariance matrices Q_k and R_k, respectively, which are used as parameters of the Kalman filter and can be estimated from a set of preliminary measurements through a trial and error procedure. Using these equations, the Kalman filter behaves as an observer, where the observable variables are $\left[{}^{O}p_{F_1'} \quad {}^{O}p_{F_2'}\right]^T$, and the state variable is the fulcrum point position ${}^{O}p_F$, which can be estimated by the Kalman filter.

$$ {}^{O}p_{F_{k+1}} = I \cdot {}^{O}p_{F_k} + w_k \qquad (3.15)$$

$$\begin{bmatrix} {}^{O}p_{F_1'} \\ {}^{O}p_{F_2'} \end{bmatrix}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_k \cdot \left[{}^{O}p_F\right]_k + v_k \qquad (3.16)$$

3.5 Soft Tissue Interaction Detector

As previously stated, a method to detect interactions with the soft tissue inside the abdomen is needed to avoid fulcrum point estimation errors when different forces and torques are exerted on the surgical instrument. To solve this problem, a method based on the surgeons' gestures is proposed. This method divides each gesture into different states and, depending on these states, the interaction with soft tissue is detected. For this purpose, the Hidden Markov Model (HMM) has demonstrated high flexibility for modelling and recognition of surgeon gestures performed by manipulators (Bauzano et al., 2014). This probabilistic model is explained in Appendix B.

3.5.1 Gesture Encoding by HMM

By considering a gesture as a movement of the surgical instruments inside the abdomen to achieve an objective, e.g., pushing up the liver, cutting tissue or tying a knot, the stage of the gesture can be used to determine whether the instrument is touching the soft tissue. For example, if a surgeon is going to push down the liver, the instrument will not be in contact with the liver until the surgeon moves the instrument towards it. Hence, a gesture can be divided into several states based on the interaction of the instrument tip with the soft tissue during its reproduction. To detect the current state of the gesture that is being performed, a model that encodes each gesture and provides the most likely state for each instant in time is needed. The discrete Hidden Markov Model (HMM) has demonstrated high flexibility for modelling and recognizing surgeon gestures performed by manipulators (Bauzano et al., 2014). Therefore, a gesture library $\Omega = \{\lambda_1, \lambda_2, \dots, \lambda_p\}$ has been created, where each gesture λ_i is encoded into an HMM (Rabiner, 1989) with parameters as follows:

$$\lambda_i = (Q, A_i, B_i, \pi_i, E) \tag{3.17}$$

where Q represents the *hidden* states; A_i denotes the state transition distribution matrix, i.e., probabilistic relationship between the states; E is the set of discrete observations; B_i is the observation distribution probability matrix, which stores the likelihood between each discrete observation and each state; and π_i is the initial state distribution vector.

The considered gestures have been divided into a set of five *hidden* states $Q = \{q_1, q_2, \dots, q_5\}$ that provide the best relationship between the forces exerted during the execution of the gesture. Hence, each state represents the following information:

- q_1: The beginning of the gesture, when there is no interaction with tissue.

- q_2: The interval between the initial interaction with the tissue and the maximum exerted force.
- q_3: The maximum exerted force interval.
- q_4: The interval between the maximum exerted force and the end of the interaction.
- q_5: Starts when there is no interaction with the tissue and determines that the gesture has finished.

In summary, the transition between states one and two signifies the start of the tissue interaction, and the transition between states four and five signifies the end of the tissue interaction. Thus, during the reproduction of a gesture, a sequence of hidden states $\hat{Q}(k) = q(1), q(2), \dots, q(j), \dots, q(k); q(j) \in Q$ is generated. Figure 3-15 shows an example of a pushing down gesture, with the interaction forces and the instrument tip velocity represented during the state sequence evolution. Thus, the first and second rows represent the vertical and lateral forces exerted throughout the instrument, respectively, as referred to in frame $\{H\}$, and the third row depicts the Z-axis velocity at the tip of the instrument, as referred to in frame $\{O\}$.

The state transition distribution matrix A_i, whose dimensions depends on the number of states and have been fixed to five, is calculated for each gesture. This matrix provides the probabilistic relationship between the states, which configures the HMM topology. Thus, element a_{mn} of A_i represents the probability of jumping from $q_m(k)$ to $q_n(k+1)$, as indicated in (3.18), where k represents the current instant in time.

$$a_{mn} = P(q_n(k+1) | q_m(k)) \qquad (3.18)$$

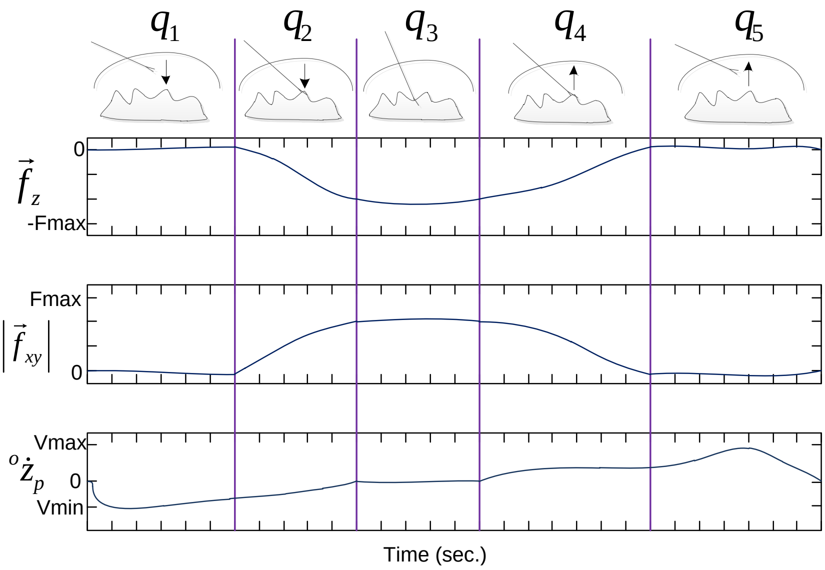


Figure 3-15. Example of the relation between the measurements and the HMM states. $|\overrightarrow{f_z}|$ and $|\overrightarrow{f_{xy}}|$ are the longitudinal and transversal exerted forces on the instrument, respectively; ${}^O\dot{z}_P$ is the longitudinal velocity of the instrument; and $q_1, q_2, \dots, q_5$ are the five hidden states that have been considered in the HMM

To allow for the fact that that a gesture could be incomplete, a modified left-right topology has been used to allow jumping between states during the execution of the gesture. Figure 3-16 presents the defined topology, where each arrow defines that the probability of jumping between states is greater than zero.

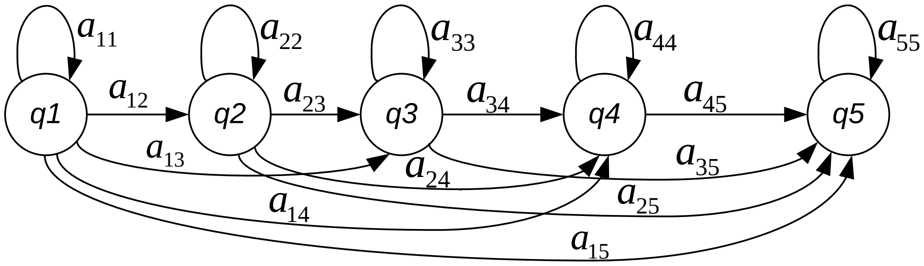


Figure 3-16. Modified left-right HMM topology

The defined HMM is based on a set of discrete observations $E = \{e_1, e_2, \dots, e_r\}$ that are used to establish a probabilistic relationship between each observation and state, as will be explained later. However, the used measurement tuple (3.14) is based on continuous values. Therefore, a method to translate continuous measurements into a set of discrete observations is needed. To solve this issue, the use of Vector Quantization (VQ) is proposed. This technique has been widely used to obtain a set of discrete symbols for speech (Rabiner et al., 1983) and gesture (Mitra et al., 2007) recognition based on HMMs. It creates a partition of the measurement feature space using the k-means algorithm (Appendix C). Thus, during the reproduction of each gesture, a sequence of measurements $\hat{\phi}(k) = \phi(1), \phi(2), \dots, \phi(k)$ is obtained. It is used to create a partition of the measurement feature space and to obtain E, which represents the set of r discrete observations that cover the measurement feature space. Hence, $\hat{\phi}$ can be translated into an observation sequence $\hat{E}(k) = e(1), e(2), \dots, e(j), \dots, e(k); e(j) \in E$, which replaces each measurement tuple by its corresponding observation. Once E is obtained, the probabilistic relationship between each observation and state is defined by the observation probability distribution matrix B_i, whose element b_{mn} represents the probability of measuring the observation $e_m(k) \in E$ in the state $q_n(k) \in Q$ as follows:

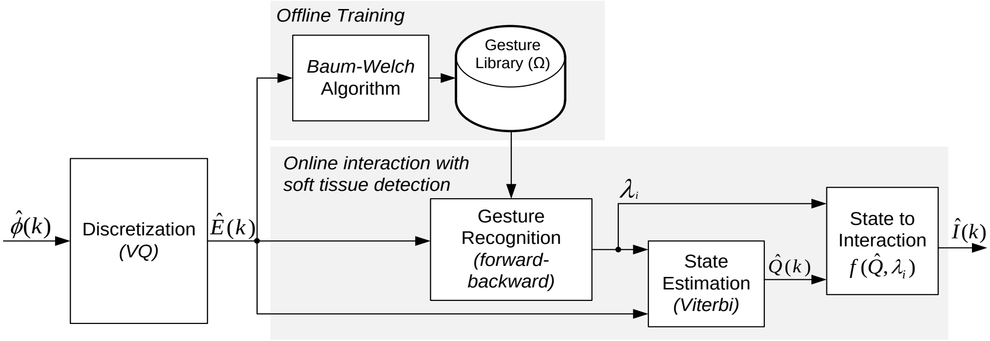
$$b_{mn} = P(e_m(k) | q_n(k)) \qquad (3.19)$$

The dimension of this matrix is $r\ x\ 5$, and its value can be estimated during the training stage.

Finally, the initial state distribution π_i represents the probability of starting for each state. It is defined by a five-element vector that is estimated during the training stage. Owing to the predefined HMM topology, $\pi_i = [1,0,0,0,0]$, which indicates that the HMM always starts at q_1.

3.5.2 Interaction with Soft Tissue Training and Recognition

Once the method of encoding each gesture into the HMM is defined, the proposed training and recognition block diagram can be obtained, as illustrated in Figure 3-17. First, the obtained measurement sequence $\hat{\phi}(k)$ is discretized into an observation sequence $\hat{E}(k)$ by VQ, as previously explained. It is used in the next two procedures: offline training, which is performed during the training stage to obtain the gesture library Ω, and online interaction with soft tissue detection, which recognizes the gesture λ_i that is being performed and identifies the state sequence of the recognized gesture $\hat{Q}(k)$ to detect whether there is an interaction with the soft tissue $\hat{I}(k)$.



Offline Training
Baum-Welch Algorithm
Gesture Library (Ω)
Online interaction with soft tissue detection
$\hat{\phi}(k)$
Discretization (VQ)
$\hat{E}(k)$
Gesture Recognition (forward-backward)
λ_i
State Estimation (Viterbi)
$\hat{Q}(k)$
State to Interaction $f(\hat{Q},\lambda_i)$
$\hat{I}(k)$

Figure 3-17. Block diagram of the interaction with the tissue detector.

During the offline training stage, the parameters of each gesture (HMM) have to be estimated through a training procedure. Thus, each gesture is reproduced n times, and the obtained training observation sequences $\hat{E}_1(k), \hat{E}_2(k), \dots, \hat{E}_n(k)$ with their corresponding previously known hidden state sequences $\hat{Q}_1(k), \hat{Q}_2(k), \dots, \hat{Q}_n(k)$ are used to train each HMM using the Baum-Welch algorithm. This algorithm provides the HMM parameters A_i, B_i, and π_i using an iterative procedure based on the training observation sequences and the hidden state sequence to maximize $P\left(\hat{E}(k)\middle|\lambda_i\right)$, which is the probability that an observation sequence $\hat{E}(k)$ belongs to the HMM of the performed gesture λ_i. Using this algorithm to train each gesture, the gesture library Ω can be obtained.

Once each gesture is trained and stored in Ω, this gesture library can be used to recognize the gesture that is being performed and the corresponding state. Thus, the gesture recognition block (Figure 3-17) calculates the most likely gesture λ_i using the *forward-backward* algorithm (Rabiner, 1989). This algorithm is based on dynamic programming and calculates $P\left(\hat{E}(k)\middle|\lambda_i\right)$, which can be used to select the most likelihood performed gesture i as follows:

$$i = \arg\max_{1 \le i \le p}\left[P(\hat{E}(k)|\lambda_i)\right] \qquad (3.20)$$

Once the gesture is obtained, the sequence of the most likely *hidden* states $\hat{Q}(k) = q(1), q(2), \dots, q(j), \dots, q(k); q(j) \in Q$ is estimated using the *Viterbi* algorithm (Viterbi, 1967). This algorithm provides the most likely sequence of hidden states for a given HMM λ_i and an observation sequence $\hat{E}(k)$. The last state $q(k)$ represents the current state during the gesture reproduction.

Finally, the interaction sequence $\hat{I}(k) = i(1), i(2), \dots, i(j), \dots, i(k); i_j \epsilon \{yes, no\}$ is obtained by the function $\hat{I}(k) = f\left(\hat{Q}(k), \boldsymbol{\lambda_i}\right)$, which establishes a relationship between each state $q(j)$ and the interaction with soft tissue, taking into account the recognized gesture λ_i.

Because this method can be executed in real time, its computation complexity is important for maintaining the minimum sample time. Thus, there are three blocks whose time complexities are analysed. The complexity of the VQ is $O_{VQ}(r)$ because this algorithm consists only of searching for the nearest observable feature element e_j. The complexity of the forward-backward algorithm is $O_{FB}(r \cdot u^2 \cdot p)$, and that of the Viterbi algorithm is $O_V(r \cdot u^2)$ (Rabiner, 1989), where r is the size of the set of observable features, u is the number of states and p is the number of trained gestures. Therefore, the computational complexity O_D of the soft tissue interaction detector can be given as follows:

$$O_D = O_{VQ} + O_{FB} + O_V \tag{3.21}$$

For example, for the case of an observation codebook of 512 elements (r = 512), 4 trained gestures (p = 4) and 5 states of the HMM (u = 5), a total of 64,512 computational units are executed during the sampling period. Considering that each computational unit takes no longer than 10 ns (a typical RISC 20 MHz microcontroller computes one instruction in 0.5 ns), a 0.65 ms minimum sample time may be used. Therefore, this computation can be executed in real time.

3.6 Summary

This chapter proposes a navigation method for generic kinematic structures that handle surgical instruments introduced through a multiport trocar to perform SPAS. This navigation method is based on a parallel force-position Jacobian-based control scheme whose objective is to minimize the forces exerted on the abdomen. This navigation method implements a virtual RCM that is calculated online using the information provided by two F/T sensors placed on the manipulator end effector. However, when this type of sensor is placed on the manipulator end effector, the provided measurements could be distorted when there are several interaction forces throughout the instrument, e.g., the forces exerted during the interaction of instruments with the soft tissue inside the abdomen. To solve this issue, a soft tissue interaction detector is proposed. It is based on surgeon gestures that have been divided into internal states and encoded by the HMM. Using this detector, each gesture is trained into a HMM, and once the gesture is reproduced, the current internal state is calculated, which provides information on the interaction of the instrument with the soft tissue.

Finally, sensory information from the manipulators and the F/T sensors can be used to assist the surgeon through haptic guidance, as explained in the next chapter.

4 Learning from Demonstration for Haptic Guidance

4.1 Introduction

In surgical teleoperation, surgeons need to receive rich sensory information from the robotic system to perform a surgery optimally. Although there has been considerable research in this field, surgeons currently do not have all of the required information during teleoperation. This lack of information causes the surgeon’s learning process to be slow and ineffective. To compensate for this shortcoming, haptic guidance has been demonstrated as a promising method to improve performance in generic teleoperated tasks, which indicates a reduction in the learning process that is achieved by robotic assistance. Conventionally, haptic guidance is implemented though *virtual fixtures,* which use a virtual spring to guide the operator to follow a prescribed reference trajectory or avoid forbidden regions. Currently, this method is position-based, i.e., it relies on an accurate reference position or trajectory, which is obtained offline using images or marks on the patient’s body. However, the location of forbidden regions or defined trajectories can be affected by measurement errors. To mitigate this problem, (Oosterhout et al., 2015) suggested combining force feedback with guidance forces, to enable the operator to perform a task despite inaccurate guidance. Therefore, it could be assumed that a guidance system that would use force information directly and correct the inaccuracy of the guidance would require less effort from the operator.

This chapter explores the use of the Learning from Demonstration approach (LfD) to generate haptic guidance references that have been previously trained. Currently, this approach is widely used in robotics for generating temporary continuous trajectories based on manipulator position (Calinon et al., 2010; Jakel et al., 2010) or contact force measurements (Chowriappa et al., 2013, Rozo et al., 2013). However, straightforward implementation of LfD-based haptic guidance into a surgical teleoperation system has not yet been achieved. The key difficulty is the use of sensory information to generate appropriate haptic guidance references that have been previously trained and modelled. The proposed LfD approach in this chapter uses only the exerted force/torque measurements on the surgical instrument to provide the guidance force references to the master device that is handled by the surgeon. Therefore, this study focuses on the description of a new LfD-based method to derive guidance reference trajectories for haptic teleoperation in surgical robotics.

This chapter explains the proposed method, which uses sensory information to generate a model of a manoeuvre to guide the surgeon to perform it during teleoperation. Thus, the surgeon will receive only guidance forces instead of direct force feedback from the instrument interaction. The proposed method is divided into two stages. The first stage consists of training a model of the manoeuvre from previous surgeon demonstrations, as explained in Section 4.2.1. The second stage consists of providing haptic guidance to the surgeon during the reproduction of the manoeuvre, as explained in Section 4.2.2. Finally, a summary is provided in Section 4.3.

4.2 Method

The proposed method provides position-based haptic guidance to assist in the execution of manoeuvres of surgical instruments that are handled by slave manipulators and teleoperated by master haptic devices. For example, in NOTES, it would be difficult to introduce an instrument through a natural orifice while exerting minimal forces on the patient’s body. Another example is the necessity of pulling the gallbladder with a constant force to perform a bile duct resection during a cholecystectomy. Although these manoeuvres are easy when the instruments are directly handled by surgeons because they can feel the forces on their own hands, the lack of sensory information in teleoperation causes certain manoeuvres to be difficult. Therefore, the surgeon’s master console is commonly equipped with haptic devices that are used for two purposes: to send the position reference of the instruments to the slave manipulators, and to provide direct force feedback from the instruments to the surgeon. However, this force feedback could be replaced by haptic guidance to assist the surgeon during the execution of the manoeuvres.

The proposed method divides a manoeuvre into several gestures that are executed by the surgeon. For example, during gallbladder pulling, the instrument has to first reach the gallbladder and hold it. Then, it has to pull it with a constant force that keeps the gallbladder tense without exerting high forces to avoid harming the patient. These gestures are then trained and reproduced in the following manner.

1. Training (offline): the surgeon demonstrates each gesture several times in a simulated environment. The demonstrations can be performed using different training platforms, e.g., kinaesthetic movement of a manipulator, direct teleoperation or direct handling of a surgical instrument provided with sensors. During the u demonstration of the i gesture, a training sequence of k elements $\hat{\xi}_{iu}(k) = \xi(1), \dots, \xi(j), \dots, \xi(k)$ is generated. The training sequence is composed of $\xi(j) = [\vec{f}(j), \vec{\tau}(j), \vec{h}(j)]; 1 \leq j \leq k,$ where $\vec{f}(j) = (f_x, f_y, f_z)$ and $\vec{\tau}(j) = (\tau_x, \tau_y, \tau_z)$ represent the instrument's exerted forces and torques (interaction measurements), respectively. Assume that the objective of each gesture is to move the instrument tip towards a goal position defined as $\vec{p}(k)$, where specific $\vec{f}(k)$ forces and torques $\vec{\tau}(k)$ are exerted, and $\vec{h}(j) = (h_x, h_y, h_z)$ is defined as the ideal guidance reference. $\vec{h}(j)$ is calculated using the difference between the obtained position at each instant in time $\vec{p}(j)$ and the goal position $\vec{p}(k)$ as an elastic force that attracts the instrument tip towards the mentioned goal position, as explained in Section 4.2.1.1. Having obtained the training sequences, the objective of the training stage is to build a probabilistic model ρ_i through a Gaussian Mixture Model (GMM), which stores the relationship between the interaction measurements with the ideal guidance references at each instant j in time. Finally, each gesture model is then evaluated to analyse how it fits with the performed demonstrations, and it is then stored into a gesture library that will be used during the reproduction stage.
2. Reproduction (online): During teleoperation the surgeon performs a manoeuvre using the navigation method proposed in Chapter 3, and the interaction measurements $\vec{f}(j)$ and $\vec{\tau}(j)$, which are obtained online during the teleoperation, are used to detect the gesture that is being performed and retrieve its trained model ρ_i. Then, Gaussian Mixture Regression (GMR) is used to derive the haptic guidance reference $\vec{g}(j) = (g_x, g_y, g_z)$ from the previously trained gesture model ρ_i using the current interaction measurements. Next, the obtained haptic guidance reference $\vec{g}(j)$ is transmitted through the haptic device to the surgeon to assist him during teleoperation. Thus, the haptic device generates a homogeneous transformation matrix $\{P\}$, which contains the referenced position and the orientation of the instrument tip that is sent to the navigation system, as previously illustrated in Figure 3-14.

Both stages and their interaction with the surgeon and patient are presented in Figure 4-1. During the first stage, a training platform is used to obtain the sensory information from surgeon demonstrations, i.e., kinaesthetic movements, teleoperation, sensorized instruments, etc. Then, during the reproduction stage, the surgeon handles a haptic device that provides haptic guidance to the surgeon, whose movements generates the instrument tip position reference $\{P\}$ to the described navigation system in Chapter 3, which interacts with the patient's abdomen.

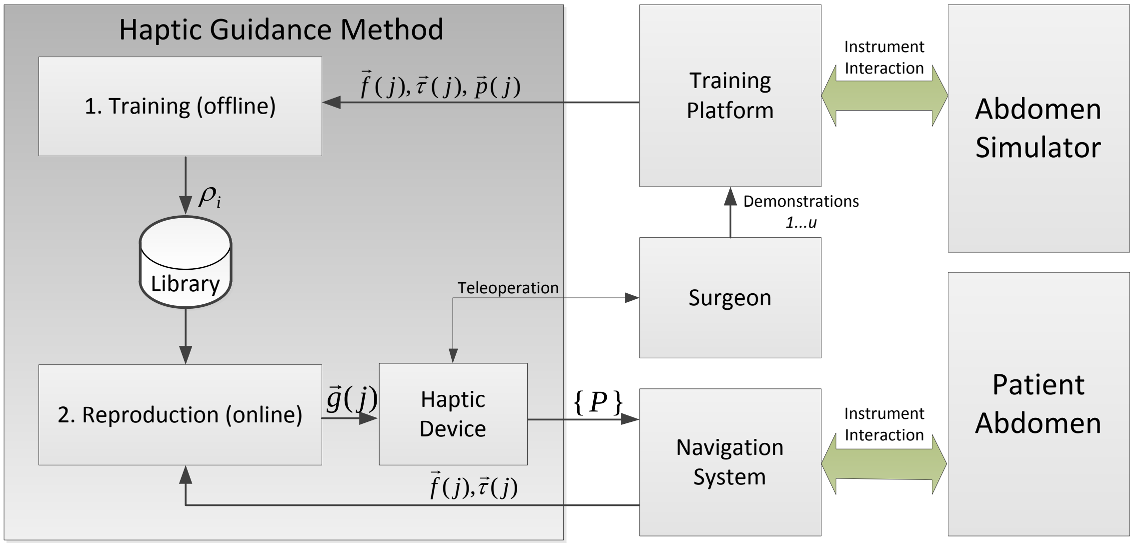


Figure 4-1. Proposed learning from demonstration for haptic guidance approach

4.2.1 Training

The primary objective of the training stage is to obtain a gesture library that contains a model of each ρ_i gesture belonging to a surgical manoeuvre. This stage is presented in Figure 4-2, where a training platform is used to perform the demonstrations of each gesture by obtaining the sensory information sequences that will be used to train the gesture, as explained below.

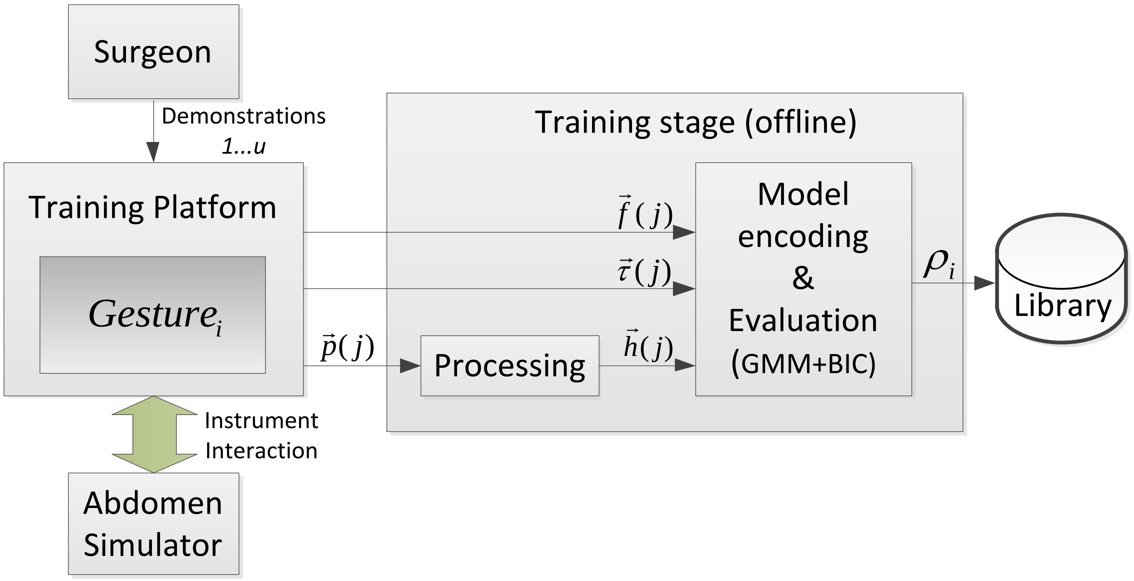


Figure 4-2. Training stage diagram. Each gesture is independently trained and stored into the library.

4.2.1.1 Processing

During the demonstration of each gesture, the position of the instrument tip $\vec{p}(j)$ is obtained at each instant j in time, as previously stated. Then, this information is used to calculate the ideal guidance reference $\vec{h}(j)$, which represents the elastic attraction forces of the instrument tip towards the goal position at each instant in time during the demonstrations. Figure 4-3 presents a simple 2D example of the use of haptic guidance to pull the gallbladder. During the training stage, the demonstrations place the instrument tip in a goal position $\vec{p}(k)$, which pulls the gallbladder with a specific force $\vec{f}(k)$, as indicated in Figure 4-3.a. Therefore, to obtain the ideal guidance references $\vec{h}(j)$ for each instant j in time during the demonstration, the Cartesian position of the instrument tip is used. Thus, the difference between the goal position $\vec{p}(k)$ and the obtained position at the j instant in time $\vec{p}(j)$ can be used to calculate the ideal guidance reference $\vec{h}(j)$, as indicated in Figure 4-3.b, where $\vec{h}(j)$ is calculated by (4.1) using a spring constant K_H that is experimentally obtained to scale the haptic guidance forces transmitted to the surgeon.

$$\vec{h}(j) = K_H \cdot \left(\vec{p}(k) - \vec{p}(j)\right) \qquad (4.1)$$

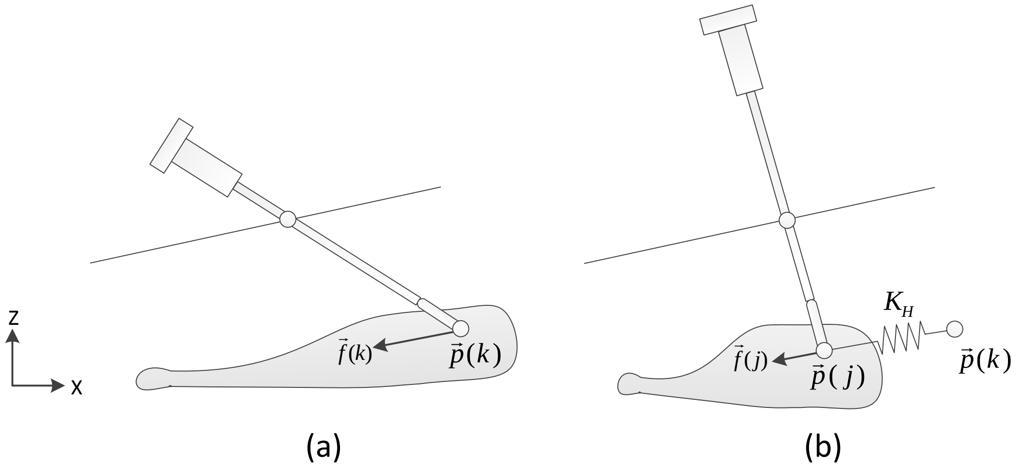


Figure 4-3. Ideal haptic guidance reference generation example for pulling the gallbladder. (a) Final position: the gallbladder is pulled with a specific force $\vec{f}(k)$; and (b) Intermediate position of the instrument: the spring provides the ideal guidance reference to reach the final position

4.2.1.2 Model Encoding and Evaluation

As previously defined, the training sequences $\hat{\xi}_{iu}(k)$ are composed of the interaction measurements ($\vec{f}(j)$ and $\vec{\tau}(j)$) and the ideal guidance reference $\vec{h}(j)$. It can be observed that the dimension of ξ is $D = 9$ elements if all of the interaction measurements and ideal guidance references are used. However, the fit of the GMM to the training sequences can be improved by removing elements that do not provide relevant information. For example, in the case of the gallbladder pulling gesture (Figure 4-3), the torques can be removed because they do not provide any useful information. Therefore, the tuple dimension would be reduced to $D = 6$ elements.

Figure 4-4 illustrates an example of 4 demonstrations (coloured lines) in the X-axis, for the 2D gallbladder pulling gesture (Figure 4-3), where the training sequences $\hat{\xi}_{1u}(k); 1 \leq u \leq 4$ are shown. It should be noted that the Y and Z axes can be represented in the same way. In this figure, the abscissa and the ordinate axes represent the component X of the surgeon exerted forces f_x and the ideal guidance references h_x, respectively. When the measured force exerted by the instrument is zero, the ideal guidance references have a maximum value h_{max} to guide the surgeon to the goal position of the instrument tip. When the forces exerted by the instrument tip increase towards the final force $f_x(k)$, the ideal guidance references decrease as the instrument tip reaches the goal position.

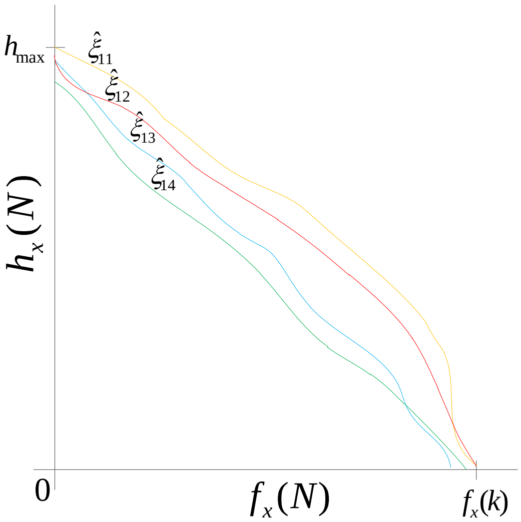


Figure 4-4. Four demonstrations of the gallbladder pulling gesture in the X-axis. Each $\hat{\xi}_{1i}$ represents the relationship between the exerted forces f_x and the estimated ideal guidance references h_x for each demonstration.

Once the training sequences $\hat{\xi}_{iu}$ are obtained for each i gesture, they are used to encode a Gaussian Mixture Model (GMM) ρ_i that will provide the most likely haptic guidance trajectory for each gesture based on previous demonstrations. A GMM is a probabilistic model that assumes that the training sequences $\hat{\xi}_{1u}$ can be included in a set of N Gaussians distributions, where each Gaussian covers a part of the training sequences. Thus, a GMM can be defined as $\rho = \{\pi_n, \mu_n, \Sigma_n\}_{n=1}^{N}$, whose parameters can be given as follows:

- Number N of Gaussians: This is an important parameter; if it is too low, the GMM will not fit with the training sequences, and if it is too high, two cases could occur: high processing time requirements during the real-time GMR owing to the high number of Gaussians in the GMM function and/or a low improvement in the fitness of the GMM with respect to the training sequences as N is increased. Therefore, to adjust this parameter, the Bayesian Information Criterion (BIC) has been used to evaluate how a GMM with different number of Gaussians N fits the training sequences.
- Prior probabilities π_n: This represents the weight of each Gaussian with respect to the demonstrations, i.e., if a Gaussian n covers more elements of the training sequences compared with another one, its prior probability will be higher.
- Means μ_n: This represents the centroid of each Gaussian of the GMM.
- Covariance matrices Σ_n: This defines the amplitude of each Gaussian n.

The model encoding consists in adjusting the parameters of the Gaussians so that they fit the training sequences. It is solved using the Expectation-Maximization (EM) algorithm (Bilmes, 1998). The algorithm is based on an iterative method that is able to approximate the Gaussians to the training sequences, thus maximizing the likelihood of the training sequences belonging to the GMM ρ_i. It is implemented as a function EM (4.2), as explained in Appendix D.1. The input parameters are all of the training sequences $\hat{\xi}_{i1}, \dots, \hat{\xi}_{iU}$, where U is the number of demonstrations, and N is the number of Gaussians.

$$\rho = EM(\hat{\xi}_{i1}, \dots, \hat{\xi}_{iU}, N) \qquad (4.2)$$

Using this function, the relationship between the interaction measurements and the ideal guidance references of each gesture are encoded into a GMM ρ_i. Figure 4-5 depicts how a GMM composed of $N = 3$ Gaussians (represented as ellipses) is adjusted to the training sequence using the EM algorithm for the gallbladder pulling gesture example, whose training sequences are represented by the X-axis in Figure 4-4. Figure 4-5.a depicts the initial configuration of each i Gaussian ($i = 1, \dots, N$) at the first iteration, which can be represented by the centre μ_i^0 (red cross) and the covariance Σ_i^0 (ellipse) placed in a random position. During the execution of the EM algorithm, the Gaussians are adjusted to the demonstrations, as indicated in Figure 4-5.b for the j iteration. Finally, the EM algorithm fits the best configuration at iteration m; thus, the Gaussians are adjusted to the demonstrations, as indicated in Figure 4-5.c.

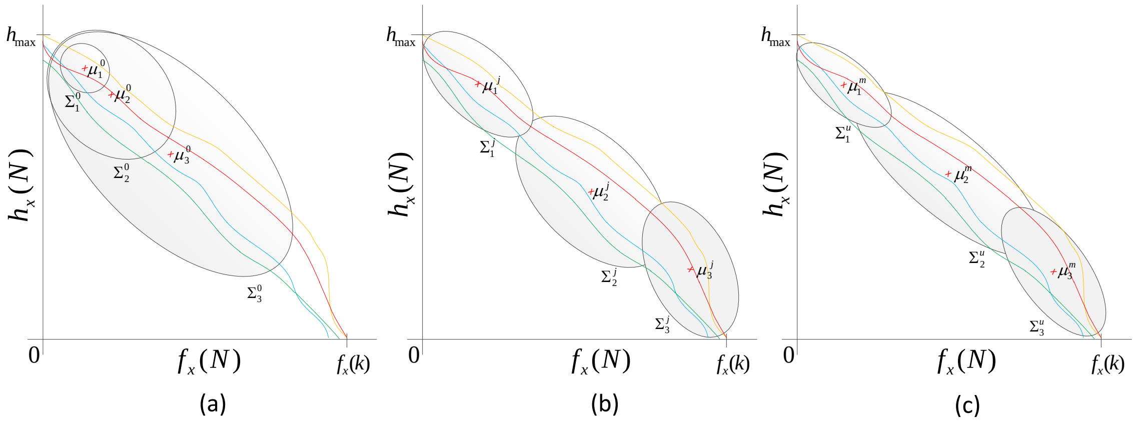


Figure 4-5. EM Training procedure for a GMM of $N = 3$ Gaussians: μ_i represents the centres and Σ_i represents the covariances of the GMM. (a) First EM iteration; (b) intermediate EM iteration; and (c) Final trained GMM

As previously shown, the EM algorithm can be used to encode a gesture into a GMM. However, the number of Gaussians is also an important parameter of the model: if it is too low, the GMM will not sufficiently fit the training sequence. Hence, Figure 4-6 illustrates an example of three encoded GMMs for the gallbladder pulling gesture, as previously illustrated in Figure 4-4, in which three number of Gaussians have been taken into account ($N = 1,3,5$). It can be observed from the figure that the fitness of the GMMs improves according to the number of Gaussians, demonstrating the best fitness for $N = 5$ (Figure 4-6.c). However, if the number of Gaussians is too high, the computer processing time of the GMR function would be excessively long, thus violating the real-time constraint, as explained in Appendix D.2.

Therefore, the number of Gaussians must be selected based on the real-time constraint and by evaluating how well the GMM fits the demonstrations. Therefore, the Bayesian Information Criterion (BIC) (Schwarz, 1978) has been used to evaluate the fitness of the GMM. This method provides a score for different estimated GMMs, which can be calculated as follows:

$$BIC = -\mathcal{L}\left(\hat{\xi}_{i1}, \dots, \hat{\xi}_{iU}\right) + \frac{n_p}{2}\log(k) \tag{4.3}$$

where $\mathcal{L}\left(\hat{\xi}_{i1}, \dots, \hat{\xi}_{iU}\right)$ represents the log-likelihood of the training sequences; $\hat{\xi}_{i1}, \dots, \hat{\xi}_{iU}$ belongs to the encoded GMM ρ, as explained in Appendix D.1; and $n_p = (N-1) + N(D + (1/2)\, D(D+1))$, where N is the number of Gaussians in the GMM and D is the dimension of ξ, as explained above. The first part of the equation denotes how well the sequence has been trained, and the second part penalizes the score based on the value of D. Thus, a lower score results in a better model. This criterion can be used for two purposes: to select the optimal number of Gaussians and to form a comparison between the different dimensions of ξ.

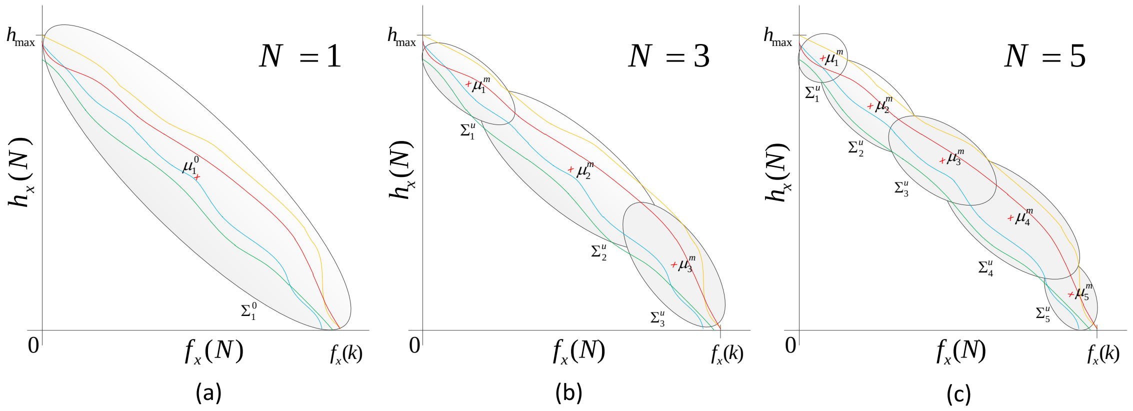


Figure 4-6. Three trained GMMs with different number of Gaussians: $N = 1, 3, 5$

Finally, once the gesture is encoded into the GMM that fits best with the demonstrations, it is stored in the gestures library, which will be used to provide haptic guidance references during the real-time reproduction of the manoeuvre.

4.2.2 Reproduction

When the gesture library has been obtained for a manoeuvre, a method is needed to retrieve the haptic guidance reference from the trained GMMs. The proposed method is presented in Figure 4-7. First, the model ρ_i of the gesture being performed has to be selected from the library using the interaction measurements $(\vec{f}(j)$ and $\vec{\tau}(j))$ obtained at the instant j in time, which is provided by the navigation system. The gesture can be identified using different methods, e.g., analysing sensory information or using the HMM, as previously explained in

Section 3.5. Then, Gaussian Mixture Regression (GMR) is used to generate the haptic guidance references $\vec{g}(j) = (g_x, g_y, g_z)$ from ρ_i based on the interaction measurements. The haptic guidance forces are then provided to the surgeon through the haptic device, which assists the surgeon's movement, thus providing the instrument tip pose $\{P\}$ that is transmitted to the navigation system described in Chapter 3.

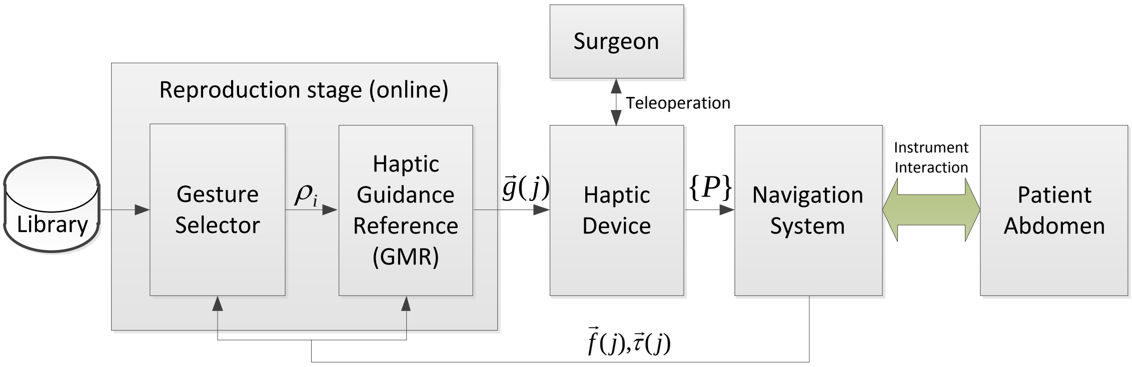


Figure 4-7. Haptic guidance reference generation diagram used during the real-time teleoperation

As previously explained, the GMM ρ_i of the gesture that is being performed is selected, and the objective is to generate the most likely haptic guidance reference $\vec{g}(j)$ for the interaction measurements provided by the navigation system. For this purpose, a GMR function has been defined in (4.4) and explained in Appendix D.2. In the case of the gallbladder pulling gesture example (Figure 4-3), $\vec{\tau}$ was removed to improve the GMM fitness, as described in Section 4.2.1.

$$\vec{g}(j) = GMR(\rho_i, \vec{f}(j), \vec{\tau}(j)) \qquad (4.4)$$

Figure 4-8 presents the obtained haptic guidance trajectory along the X-axis (red line) for the gallbladder pulling gesture example that was encoded into a GMM (Figure 4-5). As previously indicated, GMR has been used to generate the trajectory that follows the most likely path through the Gaussians, which best fits the demonstrations that were initially performed to encode the gesture. Depending on the interaction force f_x, a haptic guidance reference g_x will be provided to the surgeon to guide him to move the instrument tip to a goal position, where the gallbladder is pulled with a previously trained force $f_x(k)$. In fact, the haptic guidance force is close to h_{max} when the instrument tip interaction force f_x is low, and the haptic guidance force decreases as the gallbladder is pulled with the trained force $f_x(k)$. It should be noted that forces f_x outside the generated trajectory, i.e., lower than zero or higher than $f_x(k)$, indicate that the surgeon is not performing the encoded gesture; therefore, the GMR will not provide any haptic guidance reference $\vec{g}$.

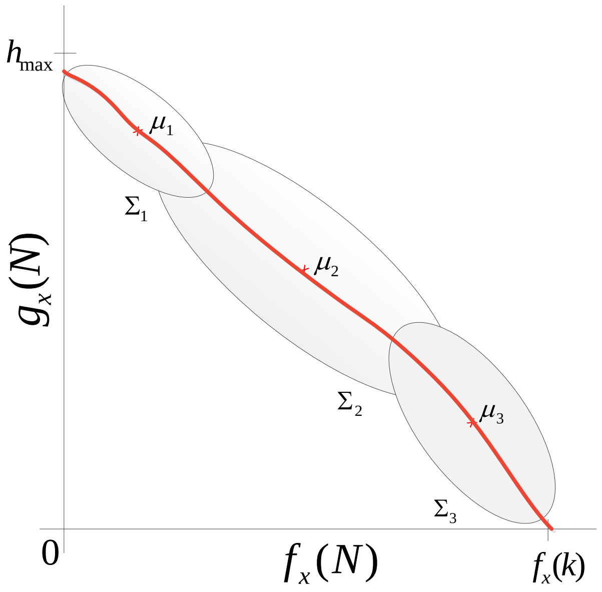


Figure 4-8. Haptic guidance trajectory obtained from GMR for the gallbladder pulling gesture

4.3 Summary

This chapter proposes a new method for haptic guidance that assists a surgeon during the reproduction of a manoeuvre that has been previously trained by expert demonstrations. The primary advantage of the proposed method is that it does not use previously fixed stored information from the patient, e.g., processed images taken before the surgery, marks on the body, etc., to provide haptic guidance through *virtual fixtures*. By contrast, it uses interaction measurements obtained in real-time during teleoperation to provide guidance forces to assist the surgeon. Moreover, these guidance forces are learned using the LfD methodology, which allows the system to be trained by experts using kinaesthetic teaching or directly teleoperating the robot. For this process, a manoeuvre is divided into several simple gestures that are trained separately. Each gesture is encoded into a GMM that is stored in a gesture library, which will be used during the reproduction of the manoeuvre. Hence, different haptic guidance references are generated depending on the gesture that is being performed taking into account that different gestures could generate opposing guidance forces owing to the characteristics of the manoeuvre. Once the gesture library has been obtained, it is used during the reproduction stage. First, the gesture that is being performed is selected. Then, the haptic guidance references are generated using the GMR.

This method has been implemented in a generic teleoperation system to solve the peg-in-hole insertion task, which is more complex than the example used in this chapter. The experimental results are presented in Section 5.3.

5 Implementation and Experiments

5.1 Introduction

In previous chapters, the theoretical aspects of two approaches that comprise the proposed smart navigation method for SPAS have been described. In particular, Chapter 3 proposes a navigation method that improves the estimation of the fulcrum point using a soft tissue interaction detector and a Kalman filter measurement fusion method, and that also minimizes the forces exerted by the instrument on the abdomen using a parallel force-position control scheme. Additionally, Chapter 4 detailed an LfD-based approach that was integrated in the proposed navigation system to assist the surgeon during the execution of a task using haptic guidance.

This chapter describes how these approaches have been implemented in two robotic platforms to obtain experimental results that validate both approaches. Specifically, the navigation method for SPAS has been implemented in the CISOBOT platform, which was developed by the Surgical Robotics Team from the Universidad de Malaga; and the haptic guidance approach has been implemented in the LWR Taskboard Workcell from the Telerobotics and Haptics Laboratory at ESA-ESTEC. It should be noted that only a preliminary study of the haptic guidance approach has been conducted, i.e., solving a complex task: the peg-in-hole insertion, which demonstrates that the proposed approach is able to solve similar surgical tasks, such as inserting an instrument through the trocar or pulling the gallbladder during a cholecystectomy by applying a previously trained force.

This chapter describes the experimental results of the proposed smart navigation method. Four experiments, which validate the proposed parallel force-position control scheme, gesture recognizer and soft tissue interaction detector, are presented in Section 5.2. Section 5.3 describes how the method for the LfD-based haptic guidance has been used to solve a trained generic task, i.e., the peg-in-hole insertion, and its experimental results are provided in this section. Finally, a summary of the experimental results are described in Section 5.4.

5.2 Navigation Method for SPAS

This section describes the experiments performed to validate the proposed navigation method. The primary objectives of the performed experiments are explained, and the results of four experiments, which validate the navigation method and the soft tissue interaction detector, are presented.

5.2.1 Objectives

The primary objective of the performed experiments is to validate the navigation method proposed in Section 3. This objective can be divided into four sub-objectives as follows:

- Validation of the inverse Jacobian position control scheme. The goal is to perform an experiment that validates the proposed inverse Jacobian position control scheme, as described in Section 3.4, thus demonstrating its performance in a real manipulator.
- Comparison of the parallel force-position control scheme with the fulcrum point estimation. An experiment is conducted to compare how the parallel force-position control scheme and the fulcrum point estimation work together using sensory information from two manipulators whose instruments are inserted through the same trocar.
- Gesture detector performance analysis. An analysis of the gesture detector is needed for the interaction with the soft tissue detector. Therefore, an experiment is conducted to demonstrate how the different gestures are detected.
- Validation of the soft tissue interaction detection algorithm. An experiment to validate the soft tissue interaction was performed by measuring the delay between the instant when the interaction starts and finishes and the instant when it is detected by the proposed algorithm.

5.2.2 Parallel Force-Position Control Scheme Validation

This section describes the experiments that have been conducted to validate the proposed parallel force-position control scheme, which has been implemented in the CISOBOT platform, as described in Appendix E. Using the control scheme, both manipulators are teleoperated using haptic devices.

The first experiment validates the implementation of the position control scheme (defined in Section 3.4) without force feedback to demonstrate the accuracy of the manipulator when it is moving around a predefined fulcrum point using the proposed control scheme. Then, the fulcrum point estimation and force feedback are included in the control scheme. Using both manipulators, a second experiment is conducted to validate that the abdominal forces decrease and the fulcrum point estimation improves.

5.2.2.1 Experiment 1: Position Control Scheme

The objective of this experiment is to demonstrate the use of the proposed position control scheme based on the fulcrum point constraint during the movement of the instrument tip. In this experiment, two movements were performed: the first consisted of a predefined circular movement of the instrument tip based on the fulcrum point constraint, and the second consisted of teleoperation of the instrument by an operator. To demonstrate the feasibility of the proposed control scheme, the position error of the instrument tip was analysed during the movement.

As mentioned before, the first movement was defined as a 100 mm circular trajectory in a plane starting from $^{O}p_{P} = [450\ 50 - 200]$ mm, where the fulcrum point position has been fixed to $^{O}p_{F} = [500\ 0\ 0]$, as shown in Figure 5-1. This movement was performed in 15 seconds.

The instrument tip error during the movement is depicted in Figure 5-2. In this figure, the first row represents the referenced trajectory (blue line) and the followed position of the instrument tip (red line) during the execution of the movement in the three axes. The second row indicates the instrument tip position error during the movement. As shown, the peak error of the instrument tip position was approximately 1.5 mm in the X and Y-axes, and the error at the end of the movement was close to zero in all of the axes.

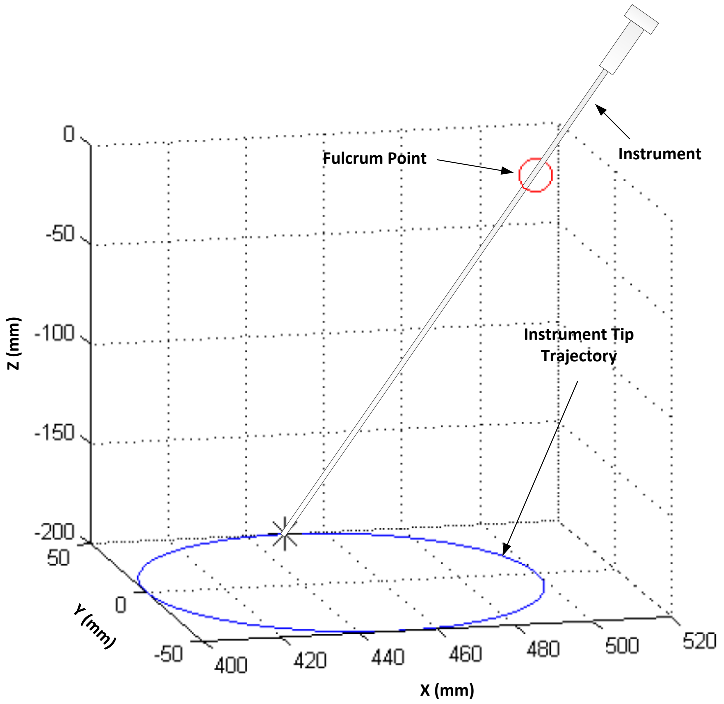


Figure 5-1. Circular trajectory used to validate the position control scheme

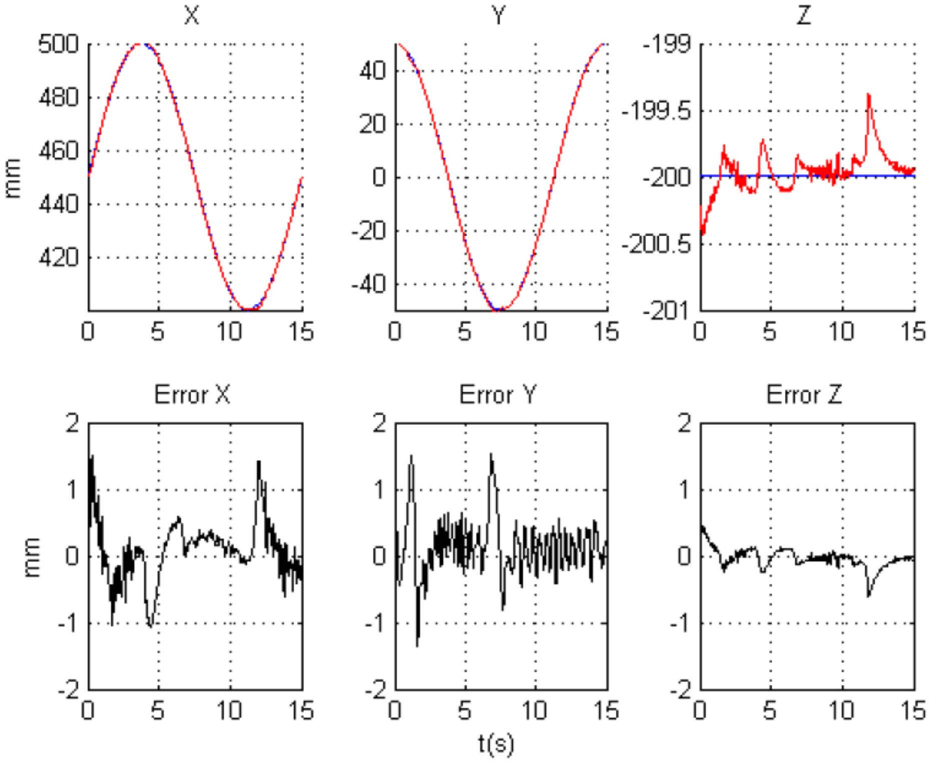


Figure 5-2. Instrument tip error during the circular movement

The second movement was performed by teleoperating a manipulator with a haptic device (described in Appendix D.4). The goal of this movement was to validate the performance of the control scheme and the manipulator when it was teleoperated, which involved faster movements than the obtained from a predefined trajectory. Thus, a human operator handled the haptic device and performed a movement based on a fixed fulcrum point as a virtual RCM. In this experiment, the instrument tip position error was also analysed. Figure 5-3 presents the performed teleoperated trajectory and the position of the fulcrum point.

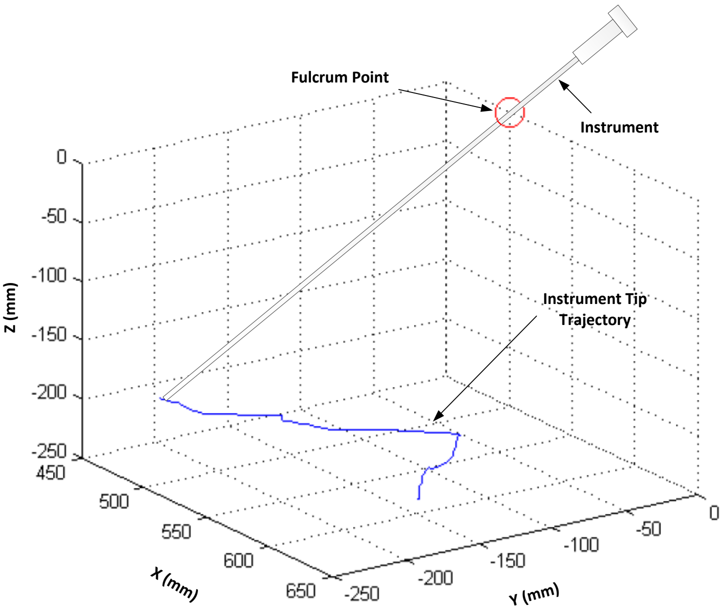


Figure 5-3. Obtained trajectory from teleoperation using a haptic device

Figure 5-4 shows the Cartesian instrument tip position reference during the movement (first row, blue line) and the followed movement of the manipulator (red line). The second row shows the error between the position reference provided by the haptic device that is handled by the operator and the followed manipulator position, which is obtained from the joint sensors. During this movement, the maximum peak error was approximately 3 mm, which can be considered normal because it occurs when the instrument tip is quickly changed, as indicated in the performed trajectory shown in the first row. The error at the end of the movement was between 0.5 mm and 1.5 mm, which is also acceptable.

As demonstrated, the inverse Jacobian-based position control scheme (presented in Section 3.4) has been validated when implemented into the CISOBOT platform, thus allowing the manipulators to be teleoperated based on a fulcrum point constraint that is translated into a virtual RCM. Therefore, the validation of the entire navigation method taking into account both manipulators, the parallel force-position control scheme and the fulcrum point estimation method is detailed in the next experiment.

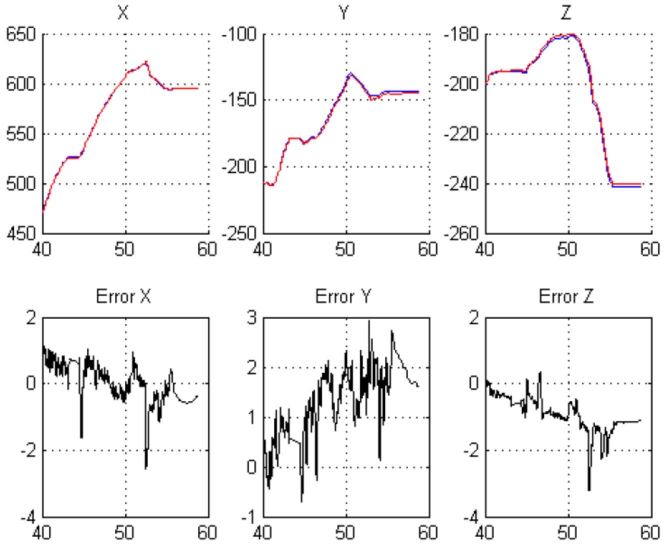


Figure 5-4. Instrument tip error during the teleoperated movement

5.2.2.2 Experiment 2: Navigation Method

The objective of this experiment is to evaluate the fulcrum point estimator and the parallel force-position control algorithm using both manipulators. This experiment starts with both manipulators handling the SPAS instruments, which were manually inserted through the multiport trocar. Then, both manipulators performed a pre-programmed trajectory that consisted of a rotation of $\pi/6$ rad about the Y-axis from the fulcrum point in 1.5 seconds. Figure 5-5 depicts the initial and final positions of both manipulators. The distance covered by both instrument tips is 35.8 mm. As shown, each manipulator performed this rotation in the opposite direction. To avoid inaccurate fulcrum point estimations due to F/T sensor errors when the exerted forces are low, a dead zone was defined between -2 N and 2 N, which indicates that if the measured forces are in this interval, they are not used to estimate the fulcrum point, and the previous estimation is used. Four trials were performed to compare the performance of the proposed navigation method when the Kalman filter fusion method and/or the force control are enabled or disabled.

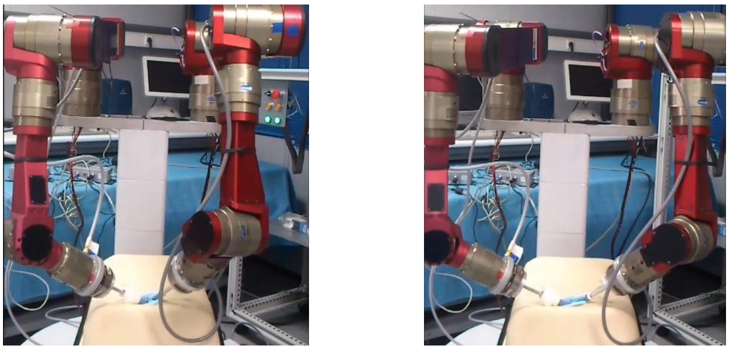

(a) (b)

Figure 5-5. Pre-programmed trajectory to validate the proposed navigation method: (a) initial position; and (b) final position

During the reproduction of the movements, the exerted abdominal forces were analysed to demonstrate that the force control minimizes them when it is enabled. Furthermore, the fulcrum point estimation error was analysed to demonstrate how the Kalman filter-based fusion method improved its estimation, and the instrument tip position error was analysed to demonstrate the accuracy of the performed movements.

The first trial was conducted with both the force feedback control and the Kalman filter fusion method disabled. Figure 5-6 depicts the measured forces (first and second rows) and the estimation errors (third and fourth rows) for both manipulators, where the peak forces are 6.5 N, which coincides with the final exerted forces. Furthermore, these forces increase during the movement owing to the accumulation of the estimation errors. The estimation error changes during the movement. As can be observed, it reaches a peak error of 70 mm and 35 mm at the end of the movement.

The second trial was conducted with the Kalman filter fusion method enabled and the force feedback control disabled. As shown in Figure 5-7, the maximum peak force and the final exerted force are 4.8 N. Moreover, the forces increase during the movement. The fulcrum point estimation error reaches a peak of 30 mm and 21.6 mm at the end of the movement. Therefore, by comparing this plot with the previous one, the exerted forces and estimation error are reduced when the Kalman filter fusion method is enabled; however, the forces increase during the movement in the same way as in the previous trial.

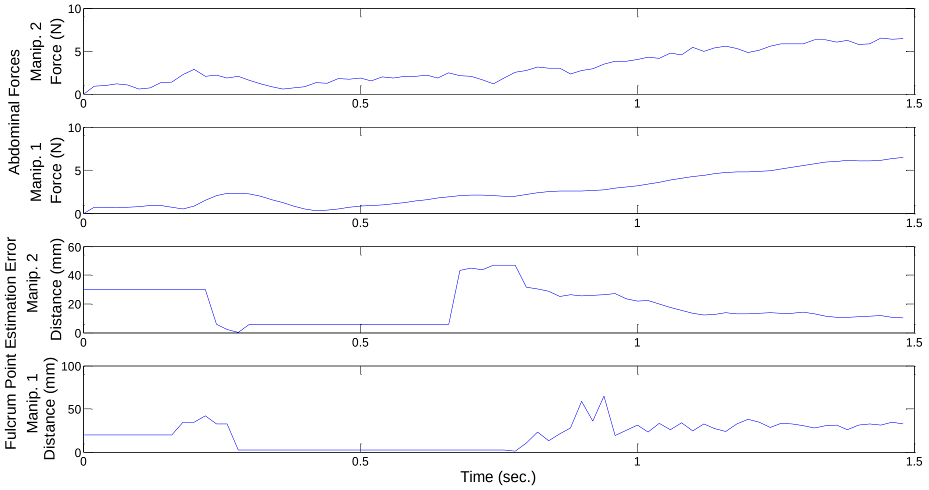


Figure 5-6. First trial: measured forces and estimation errors when both the Kalman filter and the force feedback control were disabled

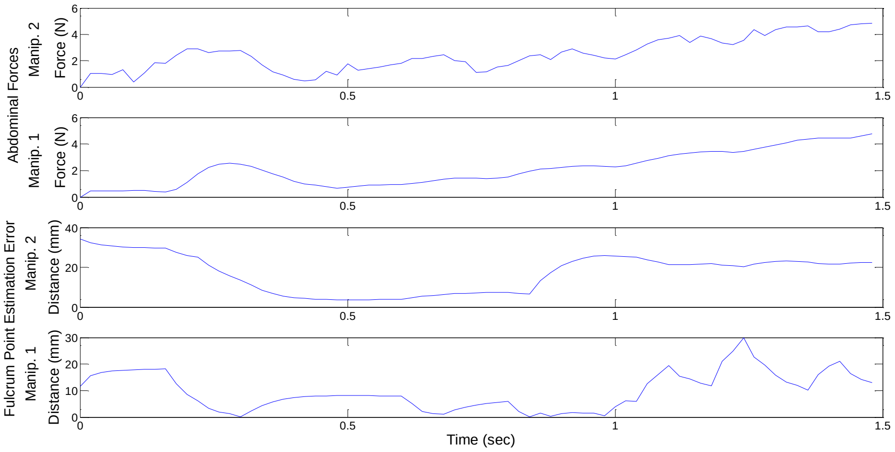


Figure 5-7. Second trial: measured forces and estimation errors when only the Kalman filter fusion method is enabled

The third trial, where only the force feedback control was enabled, is depicted in Figure 5-8. In this movement, the peak exerted force is 3.87 N, and the final exerted force is 1.5 N. In contrast to the previous trials, the forces decrease during the movement owing to the compensation of the force feedback control. The estimation error is similar to the first movement where the Kalman filter was disabled, i.e., approximately 36 mm at the end of the movement. It should be noted that the error remains nearly constant during the movement owing to the defined dead zone.

The fourth trial was conducted with both the force feedback control and the Kalman filter fusion method enabled. Figure 5-9 presents the results. It can be observed that the peak exerted force is 3.18 N and the final exerted force is 1.5 N. The behaviour of the forces during the movement is similar to that of the previous movement because of the force feedback control, and the fulcrum point estimation improves during the entire movement because of the use of the Kalman filter based fusion method, thus reaching 6.5 mm at the end of the movement.

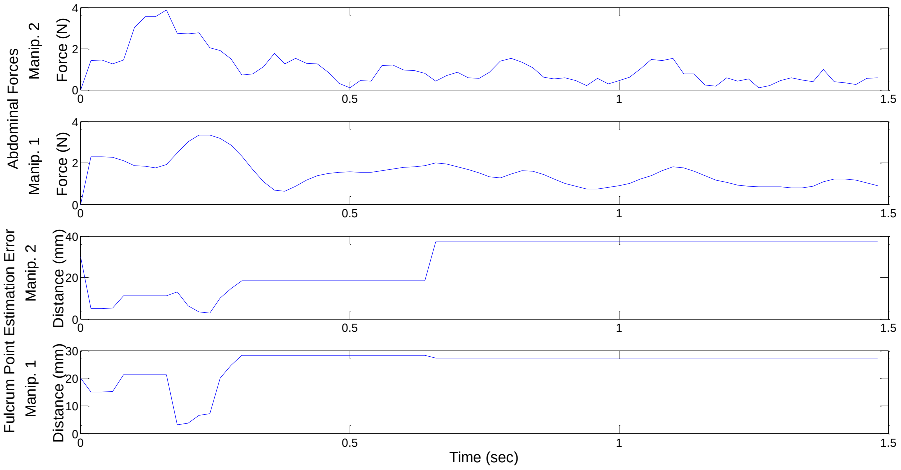


Figure 5-8. Third trial: measured forces and estimation errors when only the force feedback is enabled

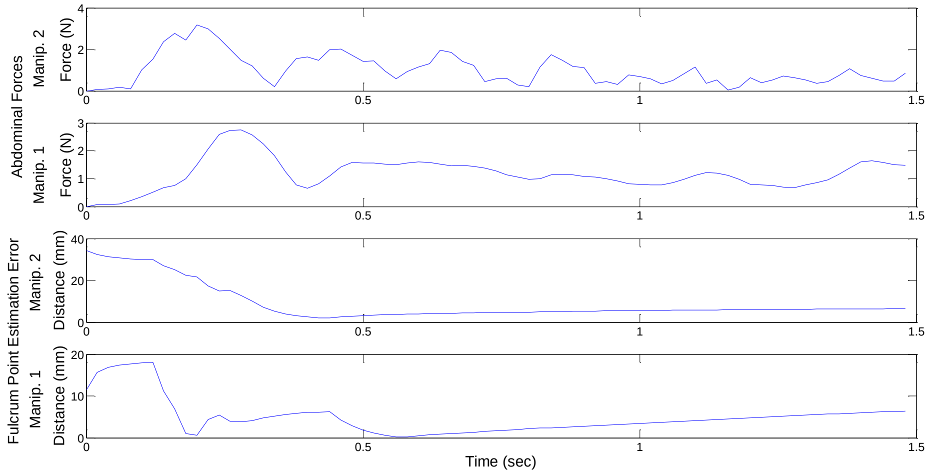


Figure 5-9. Fourth trial: measured forces and estimation errors when both the force feedback and the Kalman filter fusion method are enabled

Finally, Figure 5-10 presents a comparison between the instrument tip reference and followed trajectory for each movement. The module of the error between them for each considered case is presented in the last row. As indicated, during the movement, the error when the force control was enabled and the Kalman filter based estimation was disabled was higher than that of the case when both were disabled. This result is due to the force compensations, which move the instrument tip. At the end of the movements, both errors are similar (5.4 mm) because of the accumulation of the estimation errors during the movement. In the case of the force control being disabled and the Kalman filter based estimation being enabled, the error was reduced to 2.5 mm. Finally, the error was reduced to 0.2 mm when both actions were enabled.

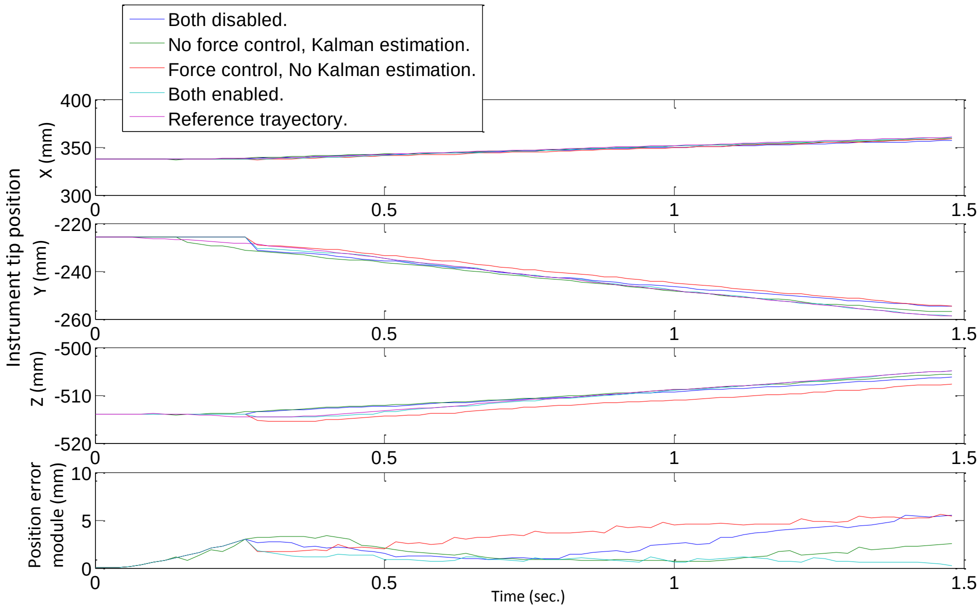


Figure 5-10. Referenced versus followed trajectory for each movement. The module of the position error is also represented.

In summary, Table 5.1 presents the results of these trials. As explained, the best results are obtained when both the force feedback control and the Kalman filter fusion estimation are enabled, i.e., trial four. However, as is indicated in trial three, when only the force control is enabled, the abdominal peak and steady state forces are reduced even with a large fulcrum point estimation error. Moreover, if only the Kalman filter estimation is enabled (trial two), the fulcrum point estimation error and the exerted forces are higher than during trials three and four owing to the accumulation of the error during the movement.

Table 5.1. Navigation Method Experimental Results

Trial	**Description**					
	Force Feedback Control	**Kalman Estimation**	**Peak Force**	**Steady State Force**	**Fulcrum Point Est. Error**	**Instrument Tip position Error**
1	Disabled	Disabled	6.5 N	6.5 N	35 mm	5.4 mm
2	Disabled	Enabled	4.84 N	4.84 N	21.6 mm	2.5 mm
3	Enabled	Disabled	3.87 N	1.5 N	36 mm	5.3 mm
4	Enabled	Enabled	3.18 N	1.5 N	6.5 mm	0.2 mm

This experiment demonstrates how the proposed navigation method works. In particular, the Kalman filter measurement fusion method improves the fulcrum point estimation when both manipulators handle instruments that are inserted through the same trocar, and the parallel force-position control scheme reduces the forces exerted on the abdomen. Finally, the instrument tip position error is reduced to 0.2 mm, which is considered low enough to perform teleoperated surgeries.

5.2.3 Soft Tissue Interaction Detection

Two experiments have been performed to demonstrate the feasibility of the soft tissue interaction detector. The first experiment demonstrates the experimental results of the gesture recognition, and the second experiment shows the results of the soft tissue interaction detection once the gesture is recognized. For these experiments, four gestures were trained, as indicated in Figure 5-11. The gestures are divided into two subsets. The first one is composed of the longitudinal insertion and extraction movements into the abdomen: nail (Figure 5-11.a) and pull (Figure 5-11.b); and the second one represents the vertical movements of the instrument tip: push up (Figure 5-11.c) and down (Figure 5-11.d). During the first subset of movements, there are low lateral forces $\left|\overrightarrow{f_{xy}}\right|$ and high vertical forces $\overrightarrow{f_z}$ on the instrument tip. Conversely, in the second subset of movements, the lateral forces are high and the vertical forces are low. Moreover, it should be noted that during the pull and push up movements, the tissue is picked up and moved back to its initial position.

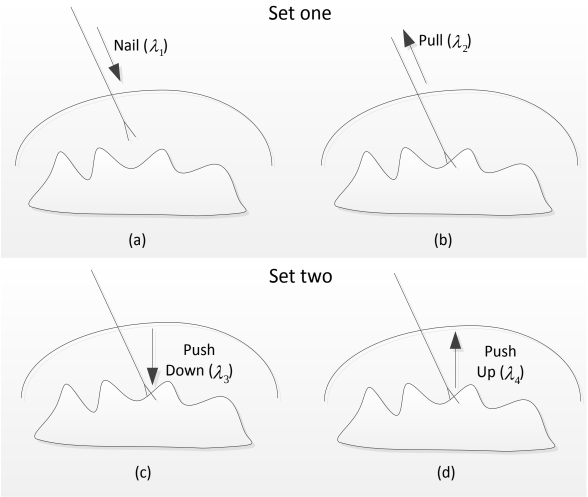


Figure 5-11. Four trained gestures. Nail: insertion of the instrument until it touches soft tissue; Pull: extraction of the instrument while it holds the tissue; and Push up and down: vertical movement of the instrument while it holds the tissue

To evaluate the proposed interaction with the tissue detector, an HMM was trained for each gesture in Figure 5-11. Therefore, a set of gestures $\Omega = \{\lambda_1, \dots, \lambda_4\}$ was obtained. The size of the observable feature set was fixed to $r = 512$ symbols, which was gathered by the VQ using the measurement tuple defined in (3.14).

To conduct this experiment, 80 movements were performed for each gesture. Half of the movements were used for the HMM training and the rest for testing the detector. These movements were executed using the teleoperation subsystem and a patient simulator with artificial tissue, as depicted in Figure E-1.a.

5.2.3.1 Experiment 1: Gesture Detection

The objective of this experiment was to demonstrate that each trained gesture is correctly recognized in most cases. The experiment was conducted using 40 samples for training and 40 for validating the gesture recognizer. The statistical results of the last 40 samples were obtained to demonstrate that most of gestures were correctly recognized.

Table 5.2 presents the obtained results. During the reproduction of these movements, each one was recognized, except for the nail gesture, where three samples were not correctly recognized. The primary reason for most of the movements being correctly recognized as gestures is related to the sensory information that was used to perform the recognition. As explained above, the different magnitudes of forces, torques and velocities are measured during the reproduction of each gesture, and they are detected by the HMM.

Table 5.2. Gesture Recognition Results

Gesture	Success	Failure	Percent
Nail	37	3	92.5%
Push down	40	0	100%
Pull	40	0	100%
Push up	40	0	100%

5.2.3.2 Experiment 2: Hidden States Estimation

As regards to the estimation of the *hidden* states, which are used to obtain the interaction with the soft tissue inside the abdomen, an experiment was conducted to validate the interaction with the soft tissue detector. The transition between states one and two, which represents the start of the interaction, and four and five, which represents the end of the interaction, were analysed.

The entire diagram in Figure 3-14 was then implemented, and the 40 samples obtained for testing were used. The delay between the instant in time when the interaction with soft tissue occured and when it was detected was analysed to demonstrate the performance of the soft tissue interaction detector. Figure 5-12, Figure 5-13, Figure 5-14 and Figure 5-15 show an example of a reproduction of each gesture and how the proposed interaction with the soft tissue detector works. In these figures, the first four rows represent the obtained measurements from a manipulator and an F/T sensor, and the fifth row shows the performed movement of the instrument tip in the Z-axis (blue line) and the position of the tissue (green line). The estimated state at each instant is shown in the sixth row; and the log-likelihood of each gesture $\ell(\lambda_i)$ during the movement is represented in the seventh row, where each line represents the log-likelihood of each gesture. It should be noted that a likelihood close to zero, which implies that the log-likelihood tends to minus infinity, is not represented in this plot ($\ell(\lambda_i) < -10^{-308}$), as can be observed in the figures when the coloured lines are not drawn. The transition between state one and two represents the start of the interaction with the tissue, and the transition between four and five represents the end of the interaction. As indicated, during the reproduction of the nail gesture (Figure 5-12), when the interaction with the soft tissue occurs, the vertical exerted forces on the instrument $\left|\overrightarrow{f_z}\right|$ between seconds 8 and 10 on the plot are negative and higher than the horizontal exerted forces $\left|\overrightarrow{f_{xy}}\right|$. The same result occurs for the pull gesture (Figure 5-13), but in this case, $\left|\overrightarrow{f_z}\right|$ is positive (for the interval between seconds 6 and 13). Conversely, when the push down and up gestures are reproduced (Figure 5-14 and Figure 5-15, respectively), $\left|\overrightarrow{f_{xy}}\right|$ is higher than or similar to $\left|\overrightarrow{f_z}\right|$, as indicated in the corresponding plots: seconds 7 and 10 for the push down gesture and seconds 2 and 7 for the push up one. Therefore, by training each gesture into an HMM using this information, as previously stated, the gesture λ_i that is being performed and its corresponding *hidden* state q_j are detected in real-time during the execution of the gesture, as shown in rows six and seven.

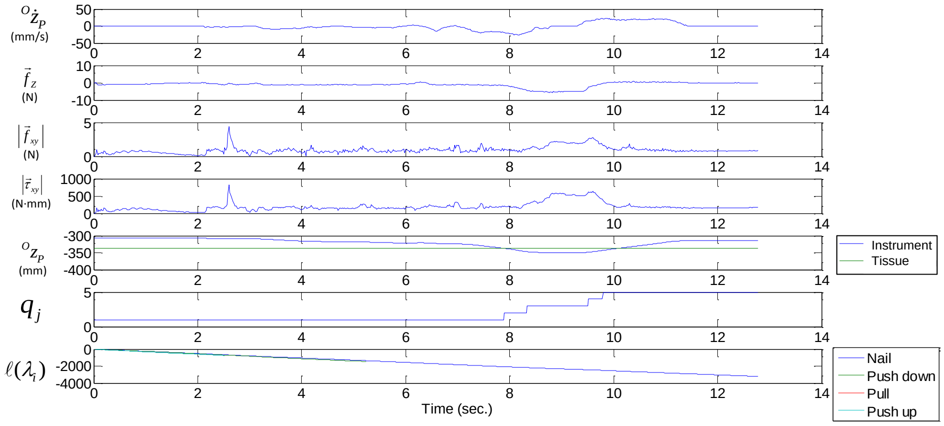


Figure 5-12. Reproduction of the nail gesture

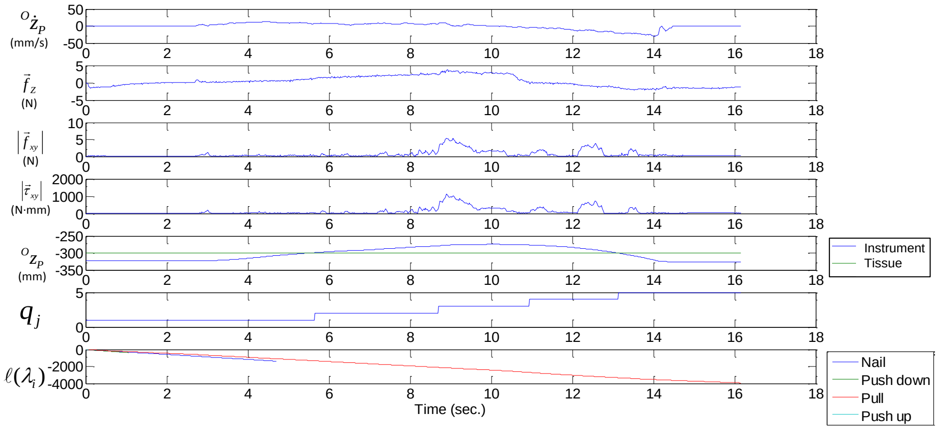


Figure 5-13. Reproduction of the pull gesture

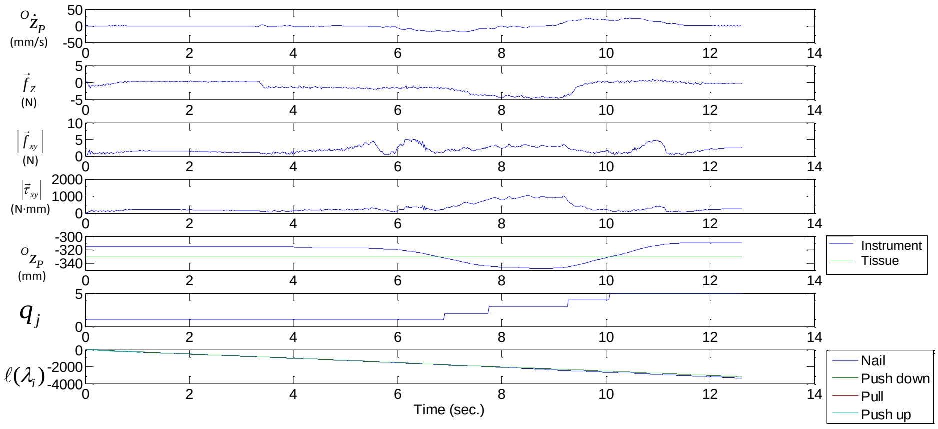


Figure 5-14. Reproduction of the push down gesture

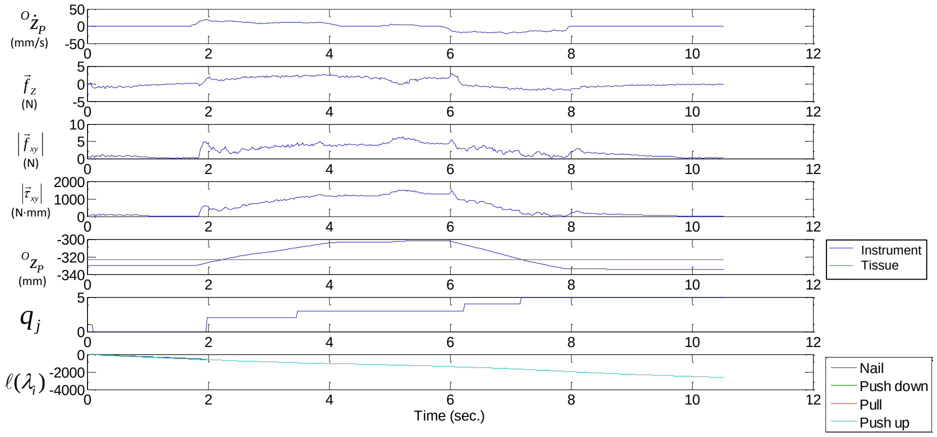


Figure 5-15. Reproduction of the push up gesture

Figure 5-16 illustrates the distribution of the delays between the instants when the interaction with the soft tissue started and ended and the instants when they were detected, i.e., transition between states one and two and transition between states four and five, respectively. As indicated, the delay at the beginning of the interaction is lower than the delay at the end of the interaction, and the push down movement has the maximum delay when the interaction starts and ends. It should be noted that it is important to detect when the interaction starts because it is the moment when the fulcrum point estimator stops. Moreover, the average delay is between 100 ms and 200 ms depending on the gesture, which is considered sufficient to not affect the fulcrum point estimation, which is performed using a least square estimator that has been adjusted to take this delay into account.

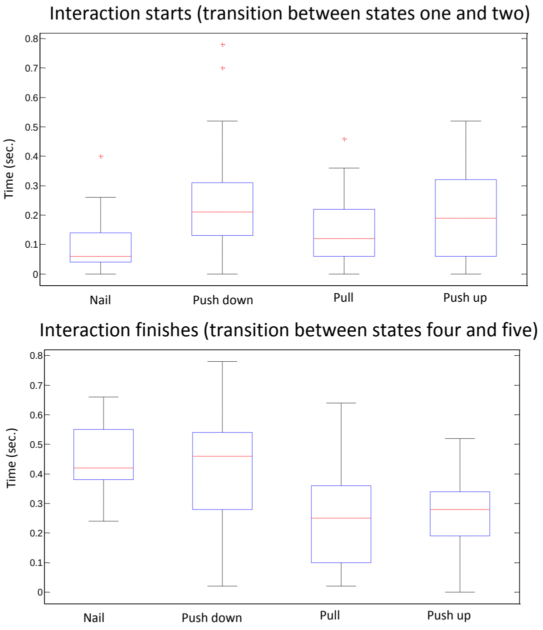


Figure 5-16. Delay distribution between the real interaction and the estimated one. This figure shows the difference between the different gestures and delays when the interaction starts and ends.

5.3 LfD for Haptic Guidance

This section describes a preliminary study that was conducted to experimentally validate the proposed LfD approach for haptic guidance. As stated in the previous chapter, it consists in providing guidance forces that have been previously trained from expert demonstrations. This approach was implemented in the LWR Taskboard Workcell (Appendix F) located at ESA-ESTEC. It was used to solve a generic and complex task: the peg-in-hole insertion, which is a de facto benchmark test for robotics assembly (Unger et al., 2001). Moreover, despite this task being trivial when performed manually, it has proven to be relatively challenging when performed by a robot (both teleoperated and automatically). Thus, the primary objectives, which are the focus of the performed experiments, are described in Section 5.3.1. Then, the specific peg-in-hole insertion task is tackled in Section 5.3.2. Finally, the results of the performed experiments are presented in Sections 5.3.3 and 5.3.4 for the training and reproduction stages, respectively.

5.3.1 Objectives

The primary objective of these experiments is to validate the proposed LfD approach. Because this approach is divided into two stages: training and reproduction, the primary objectives of these experiments can be given as follows:

- Training stage validation. To solve a task, it has to be previously trained. Therefore, an experiment is defined to demonstrate how well the proposed approach is trained to solve a generic task: peg-in-hole insertion.
- Haptic guidance accuracy. Once the training has been conducted, an experiment is needed to test how the guidance is performed by the proposed approach.

5.3.2 Peg-in-hole Task

To demonstrate the feasibility of the proposed LfD approach, the peg-in-hole insertion task has been selected. This task, despite being trivial when performed manually, has proven to be relatively challenging by the use of a robot (both teleoperated and autonomously) (Chhatpar and Branicky, 2001).

First, the task was divided into two gestures, as illustrated in Figure 5-17, which depend on the interactions between the peg and the hole during the task. At the beginning, the operator attempts to position the peg at the entrance to the hole. In this situation, lateral contacts with the hole are performed, and the objective is to guide the operator to place the peg tip in the centre of the hole. This gesture is represented in Figure 5-17.a. As shown, the hole surface is pushed with a lateral force $\overrightarrow{F_m}$. Because the peg is rigid, a reaction force of the same magnitude is transmitted to the base of the peg as $\vec{f}$. Furthermore, a small torque $\vec{\tau}$ is generated on the peg. In this case, the peg tip has to be moved horizontally to coincide with the hole. On the other hand, if the peg is already at the entrance to the hole but is not correctly oriented (Figure 5-17.b), the operator has to align the peg with the hole. In this situation, if the peg is 'pushed' down with a force $\overrightarrow{F_m}$, vertical and opposite lateral forces $\vec{f}$ arise because the peg is locked in the hole, and torques $\vec{\tau}$ arise in the opposite direction because of the lever effect. Thus, the peg may be rotated to align it with the hole.

To summarize, forces and torques of different magnitudes and directions are expected depending on the type of contact with the hole, which also require different movements to facilitate the insertion. Thus, the proposed task has been divided into two gestures: (a) *surface contact* and (b) *lever effect*.

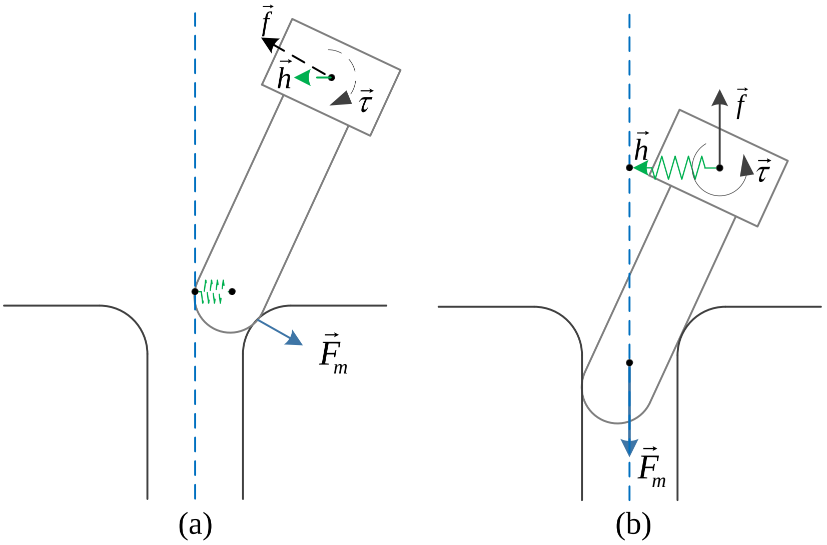


Figure 5-17. Peg-in-hole identified gestures: (a) Lateral movement, where the peg is touching the surface of the hole surface with lateral forces; and (b) Push down movement, which generates a lever effect due to the incorrect orientation of the peg. The blue arrows represent the manipulator exerted forces, whereas the green springs and arrows represent the ideal guidance forces.

To simplify the gesture selector (Figure 4-7), vertical forces exerted on the peg base f_z have been used to define a threshold that selects the gesture that is being performed. During the reproduction of the surface contact gesture, the force f_z is significantly lower than that during the lever effect gesture.

Second, to obtain the tuple ξ, an analysis of both gestures has been conducted to reduce the tuple dimension D. In the case of surface contact (Figure 5-17.a), only the lateral force vector can be used because the torques do not provide any relevant information during the reproduction of this gesture. Moreover, f_z can be removed because it is extremely low during the movement, and it does not provide any relevant information to generate the haptic guidance references. Thus, ξ_{sc} can be defined in the same way, as represented in (D.13), for the surface contact gesture as follows:

$$\xi_{sc} = \left[\xi_{sc}^{i}; \xi_{sc}^{o}\right] = \left[f_x, f_y; h_x, h_y, h_z\right] \tag{5.1}$$

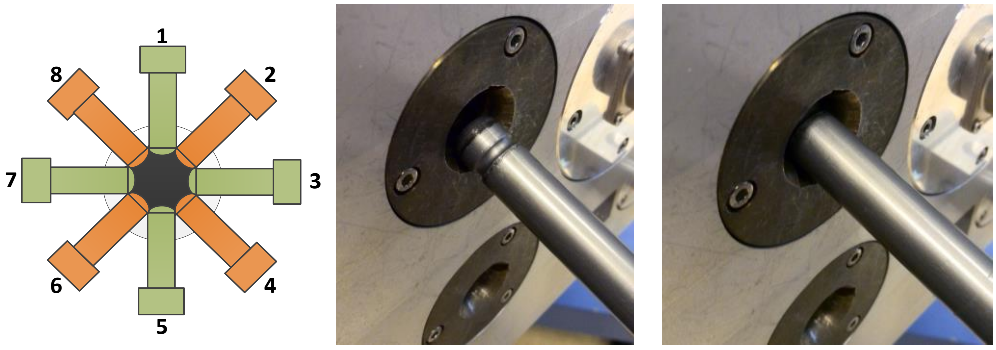
In the case of the lever effect (Figure 5-17.b), only the torque vector can be used because the forces do not provide any relevant information in this gesture. Furthermore, the component τ_z can be removed because it represents the measured torque of a longitudinal rotation of the peg, which does not occur during the reproduction of the task. Therefore, ξ_{le} can be defined as follows:

$$\xi_{le} = \left[\xi_{le}^{i}; \xi_{le}^{o}\right] = \left[\mu_x, \mu_y; h_x, h_y, h_z\right] \tag{5.2}$$

Once we have defined these tuples, both training sequences can be obtained from multiple demonstrations using a robotic teleoperation platform.

5.3.3 Training the Task

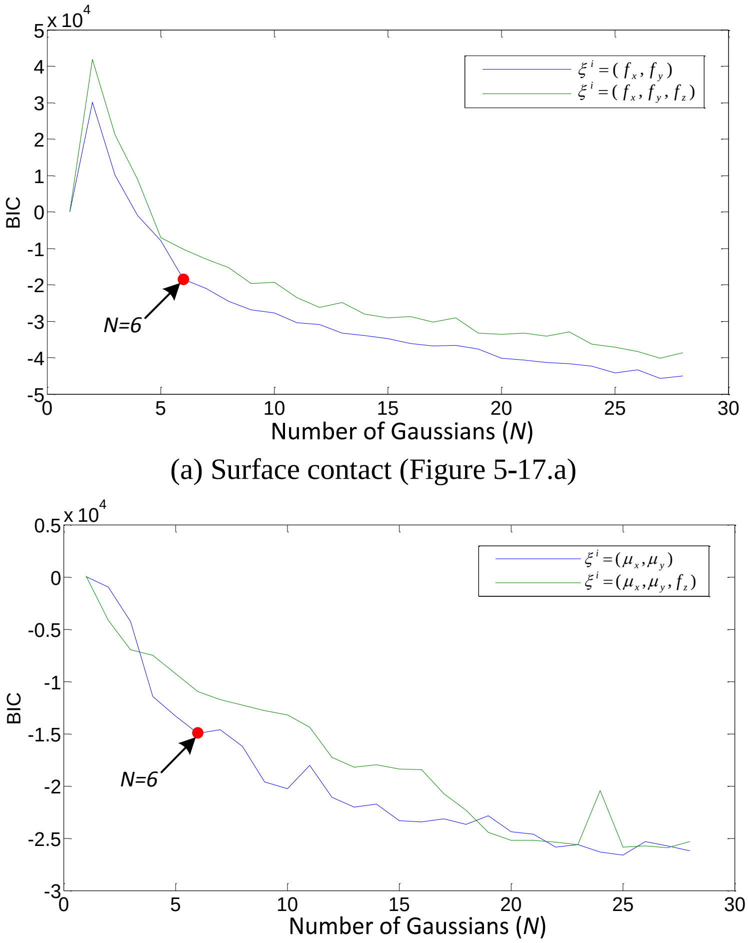
To obtain the training sequences for both gestures, the LWR was manually guided in kinaesthetic mode to demonstrate both gestures, which start from eight 45º displaced initial positions. Figure 5-18.a presents the front view of the hole and the defined initial positions that cover a circle. Figure 5-18.b and Figure 5-18.c present the starting positions of the peg for the surface contact and the lever effect gesture, respectively. Six training movements were performed for each gesture and initial position, resulting in 6x2x8 = 96 insertions.



(a) (b) (c)

Figure 5-18. Initial training positions: (a) Eight different insertion positions to cover a circle over the hole; (b) Initial position of the surface contact gesture; and (c) Initial position of the lever effect gesture

These movements were grouped so that there was only one sensory information sequence for each gesture. These sequences were processed to obtain the training sequences that were used to encode both GMMs with a different number N of Gaussians, and they were evaluated through the BIC, as outlined in Section 4.2.1. Figure 5-19 shows a comparison of the different number of Gaussians and demonstrates how the BIC score is improved when the dimension D of ξ is reduced. In this figure, the blue line represents a tuple ξ^i of two dimensions instead of three, as denoted by the green line. As expected, a lower dimension of ξ^i improves the score. Specifically, a similar score is obtained using six Gaussians and two dimensions or eleven Gaussians and three dimensions, as indicated in Figure 5-19.a. Furthermore, as depicted in both plots, the BIC score improves as more Gaussians are used. However, as stated in Section 4.2.2, the GMR processing time increases linearly with the number of Gaussians. Owing to the processing time constraint of the system (1 ms), several tests were performed, and the maximum number of Gaussians that the main computer was able to execute in real time was $N = 6$.

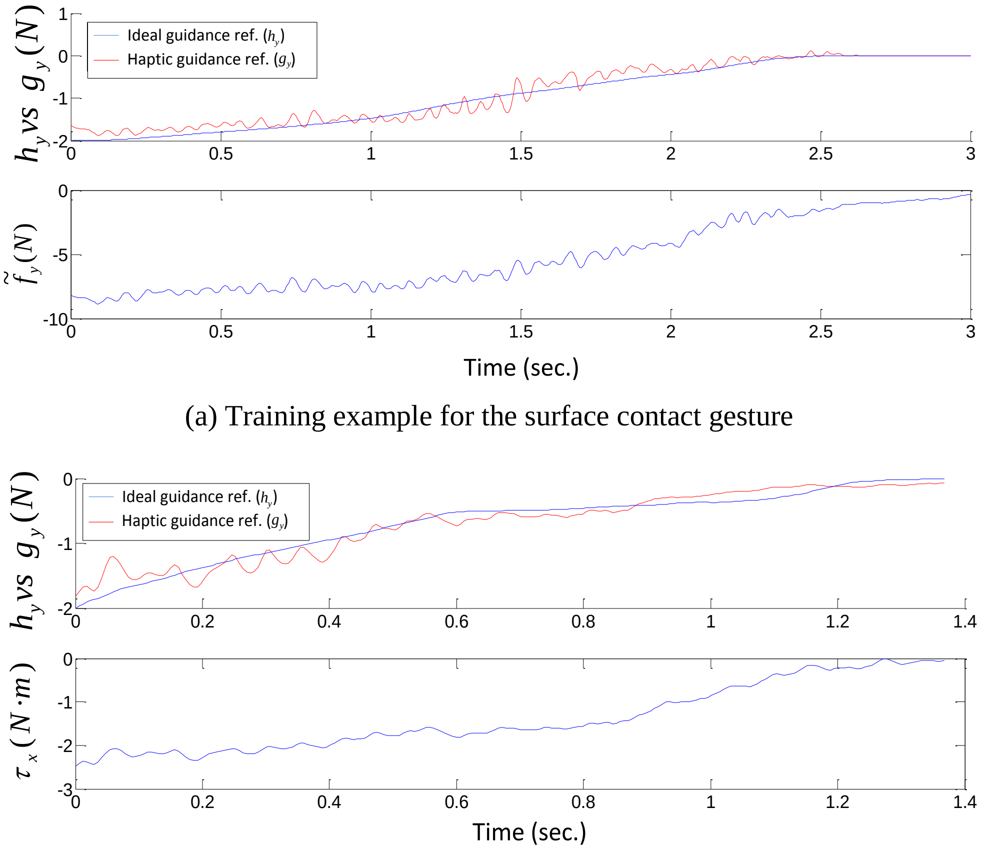


(a) Surface contact (Figure 5-17.a)

(b) Lever effect gesture (Figure 5-17.b)

Figure 5-19. BIC scores for different gestures using different tuples in the input sequences. Lower values denote a better model fit.

Once the best encoded GMMs were selected, they were stored into the gesture library, which was validated offline. For this purpose, $M = 8x2 = 16$ new validation movements were conducted in kinaesthetic mode, one for each initial position and gesture. These new movements were used to calculate the differences between the processed ideal guidance references and the obtained haptic guidance references through the GMR. Figure 5-20 depicts an example of both gestures starting at position 5 in Figure 5-18.a. The first row plots the ideal guidance force references $\vec{h}$, which are obtained by kinaesthetic teaching from (4.1) (blue line), and the output guidance force references $\vec{g}$, which are obtained from (4.4) using the measured forces and torques during the movement (red line) for the Y-axis. As indicated, the generated references follow the performed trajectory during the kinaesthetic training. The small oscillations during the movement are due to F/T sensor noise and the friction effects between the peg and the hole surface during the kinaesthetic movement of the manipulator. The measured forces and torques, f_y and τ_x, during the movement are shown in row two of each plot. They are used to generate the haptic guidance references g_y.



(a) Training example for the surface contact gesture

(b) Training example for the lever effect gesture

Figure 5-20. Example of training for each GMM, where a comparison between the ideal guidance references with the haptic guidance references is performed. These movements have been performed starting from position 5 in Figure 5-18.a.

Finally, the Root-Mean-Square (RMS) error was calculated using (5.3) for all of the validation movements, where $\vec{g}_i(j)$ and $\vec{h}_i(j)$ represent the haptic and ideal guidance references, respectively, for the movement i at an instant j in time, and k_i is the number of elements within the validation sequence.

$$e_{rms} = \sqrt{\frac{\sum_{i=1}^{M} \sum_{j=1}^{k_i} \left(\vec{g}_i(j) - \vec{h}_i(j)\right)^2}{\sum_{i=1}^{M} k_i}} \qquad (5.3)$$

Figure 5-21 illustrates the RMS error for each axis of the guidance references. It can be observed that the RMS error is lower than 0.28 N, which is achieved during the lever effect gesture.

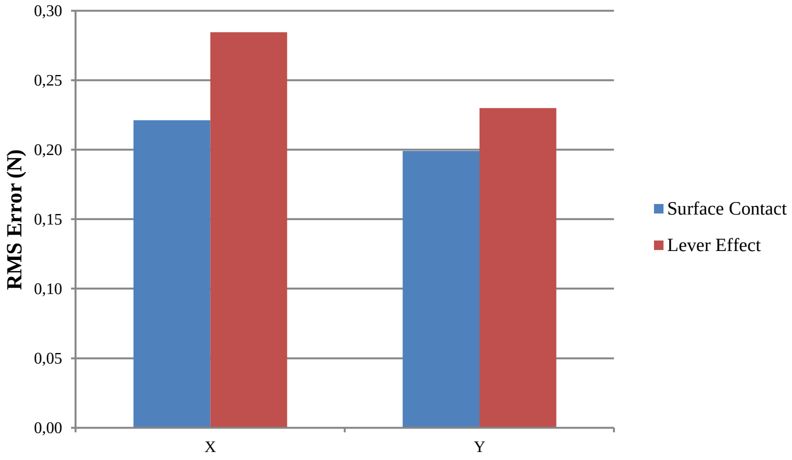
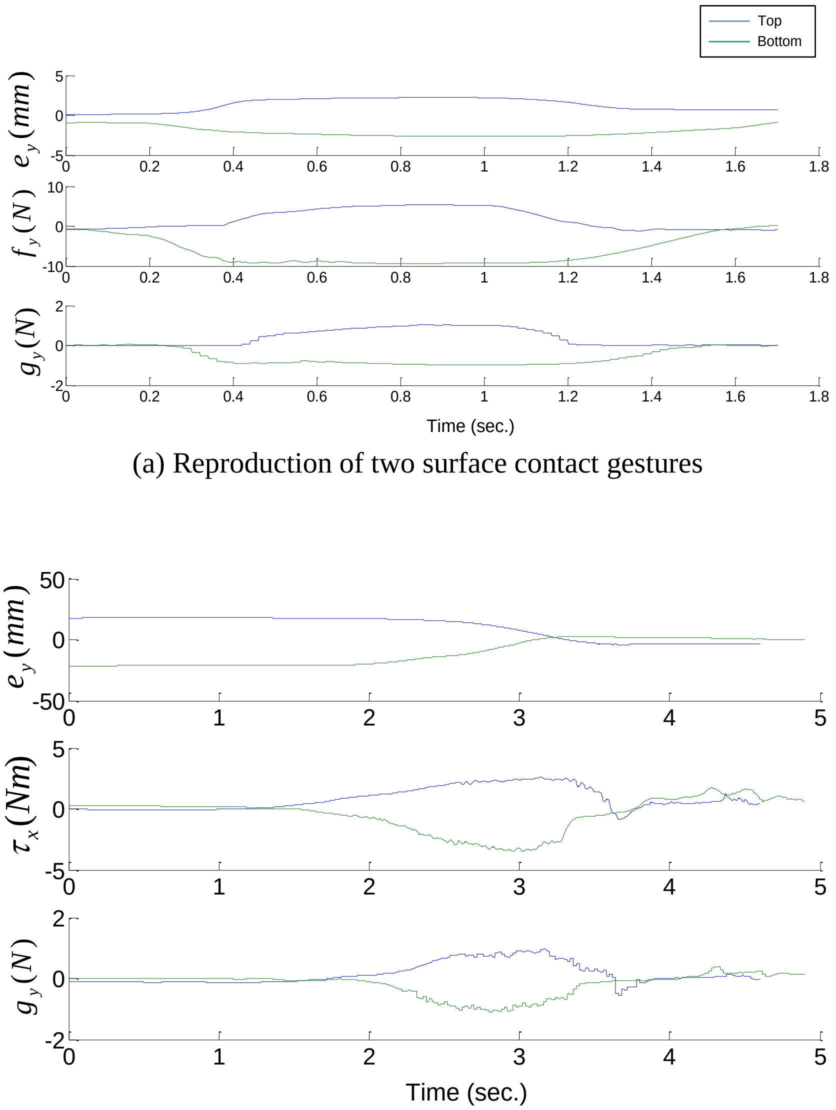


Figure 5-21. RMS Error between the ideal guidance force references and the forces obtained by the GMR

5.3.4 Reproducing the Task

Finally, a teleoperated peg-in-hole insertion using haptic guidance was performed to validate the proposed methodology. For this purpose, the Kuka LWR slave manipulator was configured in compliance mode with a Cartesian stiffness of $k_{trans} = 500$ Nm/rad and $k_{rot} = 50$ Nm/rad for translations and rotations, respectively. Moreover, the vertical forces threshold, which is used to select the gesture that is being performed, was fixed at 5 N. The operator used the Sigma.7 haptic master device to insert the peg into the hole on the taskboard, thus receiving haptic guidance forces. To keep the focus primarily on the actions of the guidance system, the measured force feedback loop was disconnected, and the operator had to rely only on the aid of the GMR-obtained haptic guidance references.

Using this configuration, each gesture was teleoperated starting from positions 1 and 5 in Figure 5-18.a, i.e., vertical movements to insert the peg. Figure 5-22 depicts the measured information, where only the implied axes are presented. The surface contact gesture results are depicted in Figure 5-22.a. The error in the Y-axis e_y indicates the difference between the current position of the manipulator end effector and the correct one to place the peg tip in the hole, and f_y represents the vertical interaction forces, which are measured by the F/T sensor and used to provide haptic guidance references g_y that reduce the error during the teleoperation. A similar result occurs in the case of the lever effect, as illustrated in Figure 5-22.b, where, e_y represents the error between the current position of the end effector and the correct position that aligns the peg with the hole. In this gesture, the torque τ_x is the interaction measurement that is used to generate the vertical haptic guidance references g_y to align the peg with the hole, thus reducing the mentioned error, as illustrated in the figure.



(a) Reproduction of two surface contact gestures

(b) Reproduction of two lever effect gestures

Figure 5-22. Example of peg-in-hole insertions from positions 1 and 5 in Figure 5-18.a. The first one has been performed starting at the top of the hole (green line) and the second one at the bottom of the hole (blue line).

5.4 Summary

In this chapter, the proposed theoretical approaches previously presented have been validated. Thus, two different teleoperation systems have been used. The first one, which was developed by the Universidad de Málaga, has been used to validate the navigation method with soft tissue interaction detection for SPAS. This approach was validated using a patient simulator and artificial tissue. The second teleoperation system, which was developed by ESA-ESTEC, has been used to validate the LfD for haptic guidance approach by solving a generic and complex task: peg-in-hole insertion.

The navigation method experimental results have been presented in Section 5.2.2, and the soft tissue interaction detector results have been provided in Section 5.2.3. As shown in Table 5.1, the abdominal peak forces, abdominal steady state forces and fulcrum point estimation errors are reduced by the proposed control scheme. Furthermore, the proposed methodology for soft tissue interaction detection has demonstrated successful results for detecting different gestures (Table 5.2), and the mean delays for detecting when the interaction starts and ends are approximately 150 ms and 300 ms, respectively, in the performed experiment, as indicated in Figure 5-16. Although there are differences between when the interaction starts and ends, it is more important to detect when the interaction starts because the fulcrum point estimation stops at that moment.

Additionally, Section 5.3 has confirmed the feasibility of using Learning from Demonstration to construct force/torque measurement-based haptic guidance to aid a teleoperated generic task, such as peg-in-hole insertion. In fact, the RMS error between the ideal guidance force reference, which was obtained during training, and the haptic guidance force reference, which was calculated using GMR, was minimal (less than 0.28 N) and independent of the direction. We can conclude that training the GMM on demonstrations of peg insertions from 8 different initial orientations was sufficient to provide uniformly good coverage of the set of possible initial conditions. Therefore, the proposed LfD approach could be used to train a specific surgical task similar to the method that has been presented, e.g., instrument insertion through the trocar, pulling the gallbladder, etc.

6 Conclusions and Future Work

6.1 Conclusions

Minimally invasive surgery (MIS) has been a breakthrough with respect to traditional surgery. Although there are several MIS techniques, Single Port Access Surgery (SPAS) is on the cutting edge of MIS. It consists in introducing all of the instruments and the endoscope through a multiport trocar, thereby reducing the number of scars to one. Therefore, the hospital stay is reduced, and the cosmetic results are improved; the scar is concealed in the umbilicus. However, this technique entails some drawbacks for surgeons, i.e., they need to improve their ability to handle special instruments because they are inserted through the same trocar, which means that they are crossed and could collide, i.e., “sword fighting”. Hence, the use of teleoperated robotic platforms emerges as an advantage to perform this type of surgery. The use of these platforms would help the surgeons to handle the instruments and avoid such problems. The da Vinci surgical system has been successfully adapted for SPAS (Kroh et al., 2011). Additionally, several platforms are being developed for research (described in Section 2.2.2). One of the issues that arise when using these platforms is related to the constrained movement of the instruments when they are introduced through a trocar. The da Vinci surgical system solves this issue with a large kinematic structure, which uses a mechanical Remote Centre of Motion (RCM) that coincides with the fulcrum point (insertion point). Other authors have solved this issue using a virtual RCM, which is estimated using the measured interaction forces along the instrument (Section 3.2). This method can be implemented in generic and lightweight kinematic structures, which is useful for SPAS because they would reduce collisions between the manipulators.

This thesis proposes a smart navigation method for SPAS that uses the measured forces along two instruments to estimate the fulcrum point, thus improving its accuracy by a measurement fusion method that takes advantage of the fact that both instruments are inserted through the same trocar. This navigation method was implemented in the CISOBOT platform, which was developed by the Universidad de Málaga and whose experimental results have demonstrated that this method improves fulcrum point estimation. Thus, the fulcrum point estimation error was reduced when the measurement fusion method was used. The estimated fulcrum point was used to perform movements using a parallel force-position Jacobian-based control scheme that takes the fulcrum point as a virtual RCM and reduces the forces exerted on the patient's abdomen. As has been demonstrated in the experimental results, the maximum error during the reproduction of different trajectories based on a fixed RCM was low for a teleoperated movement. Furthermore, the abdominal forces exerted during the reproduction of the instrument's movements were reduced for a predefined trajectory that was performed with and without force feedback.

As explained in Section 3.3, there are two interaction forces along the instrument. The first one is related to the abdominal interaction, which can be used to estimate the fulcrum point and minimize the exerted forces on the abdomen. The second one addresses the soft tissue interaction forces inside the abdomen, which could be used to provide haptic feedback to the surgeon. However, the estimation of the fulcrum point can be performed when there are only abdominal interaction forces along the instrument. Thus, a soft tissue interaction detector based on the Hidden Markov Model has been trained to detect four generic gestures: nail, pull, push up and push down. These gestures have been divided in five states, which provide information on the interaction with soft tissue. The proposed method first detects the gesture that is being performed during the reproduction of a movement and then estimates the current state, thus providing information on the interaction with soft tissue (Figure 3-17). To validate this method, it was included in the proposed navigation method for SPAS and implemented into the CISOBOT platform. Eighty movements were performed for each gesture; forty movements were used for training, and forty were used for testing the detector. Two experiments were performed using these movements. The first experiment was used to validate the gesture recognition. Each movement was recognized as the correct gesture, except for the nail gesture. Three movements were not correctly recognized when this gesture was evaluated (Table 5.2). The second experiment consisted of validating the state estimation once the gesture was recognized, which was useful to detect the interaction of the instrument with soft tissue. To do this, the delay between the instant when the interaction with tissue started and ended and the instant when they were detected was measured, i.e., the transition between states one and two and the transition between states four and five, respectively (Figure 3-15). As indicated in Figure 5-16, the average delay was acceptable to perform the fulcrum point estimation without errors.

Having defined a navigation method for SPAS, the surgeon's experience would be improved by adding a haptic guidance system that could assist the surgeon during the teleoperation of a robotic system. Thus, a Learning from Demonstration approach for haptic guidance has been proposed in Chapter 4. The goal of this approach is to use previously trained gestures from expert demonstration to provide haptic guidance force references that depend on the robot's sensory information. A Gaussian Mixture Model was used to encode relationships between sensory information and ideal guidance references obtained during the training stage. Once each gesture was encoded into a GMM, the Gaussian Mixture Regression was used to generate force references during the reproduction of the gesture. The proposed approach has been used to solve the peg-in-hole insertion task, which is a complex task when it is performed using a teleoperated robotic system. Similarly, the proposed approach could be used to solve specific surgical tasks, such as instrument insertion through the trocar, holding the gallbladder with a previously trained force, etc.

The LWR Taskboard Workcell from the Telerobotics and Haptics Laboratory at the European Space Agency was used to perform a preliminary study of the proposed approach, i.e., solving the peg-in-hole insertion task by haptic guidance. For this purpose, the task was divided into two gestures (Figure 5-17): the contact surface gesture, which was used to place the peg tip on the hole; and the lever effect gesture, which aligned the peg with the hole. Each gesture was trained from kinaesthetic demonstrations starting from different initial positions (Figure 5-18) and encoded into two GMMs. Then, new validation movements were used to measure the RMS error between the trained movements and the output generated by GMR, as indicated in Figure 5-21. This error was acceptable for using the trained models in haptic guidance. Finally, several peg-in-hole insertions were performed using the teleoperation platform. The results shown in Figure 5-22 demonstrate that the error decreases during guided teleoperation.

In summary, this thesis proposes a smart navigation method for SPAS that takes into account the fulcrum point constraint using abdominal interaction forces and torques along the instrument. Because there are also forces on the instrument tip when the instrument interacts with the soft tissue inside the abdomen, a soft tissue interaction detector has been included in the navigation method to detect the occurrence of interactions with soft tissue. Moreover, the surgeon's experience may be improved by the use of a haptic guidance approach. Therefore, an

LfD approach has been proposed and used to solve a generic and complex task, i.e., peg-in-hole insertion, which is similar to specific surgical tasks.

6.2 Future Work

The research performed in this thesis could be advanced in different ways, as described below.

1. **Isolate interaction forces throughout the instrument**

It would be desirable to isolate the interaction forces throughout the surgical instrument by obtaining the abdominal interaction forces to estimate the fulcrum point and using the soft tissue interaction forces on the instrument tip to provide haptic feedback. Thus, the proposed interaction with the soft tissue detector would be modified to provide this information.

2. **Take into account articulated instrument tips in the proposed navigation method**

The proposed navigation method has been tested with rigid instruments; however, in SPAS, the instruments typically have articulated tips. Therefore, the proposed navigation method would be expanded to address this issue.

3. **Collisions avoidance during the teleoperation of manipulators in SPAS**

Because the surgical instruments are inserted through the same trocar, the external manipulators that handle these instruments could collide during teleoperation. Hence, the navigation method could be extended to take into account this issue. Moreover, redundant kinematic structures would be used to avoid this problem, thus extending the navigation method to provide the correct joint positions.

4. **Perform a user study to improve the proposed haptic guidance approach**

Although the proposed haptic guidance approach has been experimentally validated, a user study would be desirable to analyse the operator experience when using the haptic approach, as described in this thesis.

5. **Use of the haptic guidance approach to solve concrete surgical tasks**

As has been presented in this thesis, the proposed haptic guidance approach has been validated by solving the generic and complex peg-in-hole insertion task. However, it would be desirable to validate this approach for specific surgical tasks.

A. Position Control Scheme Stability Analysis

As stated in Section 3.4, a parallel force-position control is proposed. The position control scheme (Figure A-1) is based on the inverse Jacobian $J^{-1}(\theta)$ and the forward kinematic function ${}^{B}T_{P}(\theta)$. The use of these functions in the control loop causes the system to be nonlinear. Therefore, the stability of this control scheme needs to be demonstrated.

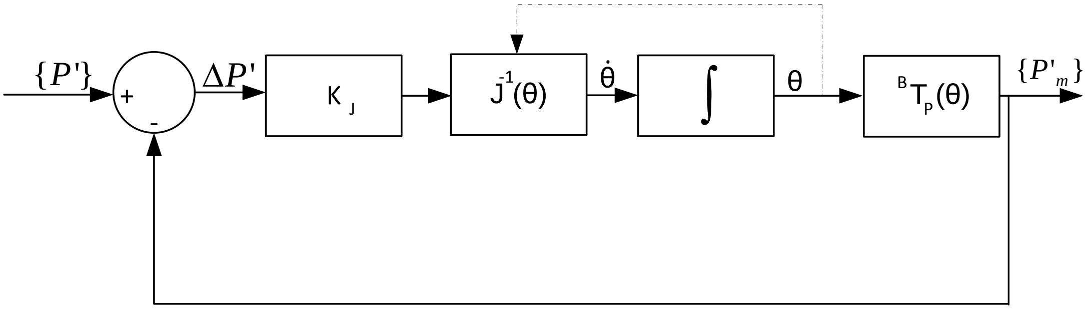


Figure A-1. Position control scheme

Thus, the direct Lyapunov method (Danwei and McClamroch, 1993) consists of defining a fictitious energy function that is used to demonstrate the system stability. Generally, this function represents the total energy of the system, which will be positive when the system is in a transitory status and null when the system reaches the steady state. Moreover, to ensure stability, the derivative of this function may be negative. This result indicates that the system loses energy until it reaches a steady state. To demonstrate the stability of the proposed control scheme, the selected candidate function can represent the kinetic energy of the Cartesian and rotational velocities $\Delta P'$ as follows:

$$V(\Delta P') = \frac{1}{2}(\Delta P')^2 \qquad \text{(A.1)}$$

To ensure stability using the direct Lyapunov method, this function must verify

$$\begin{gathered} \forall(\Delta P' \neq 0) \to V(\Delta P') > 0 \\ (\Delta P' = 0) \to V(\Delta P') = 0 \end{gathered} \qquad \text{(A.2)}$$

$$\begin{gathered} \forall(\Delta P' \neq 0) \to \dot{V}(\Delta P') < 0 \\ (\Delta P' = 0) \to \dot{V}(\Delta P') = 0 \end{gathered} \qquad \text{(A.3)}$$

Thus, the conditions in (A.2) are achieve by (A.1), and the conditions in (A.3) can be demonstrated as follows:

The first derivative of (A.1) is

$$\dot{V}(\Delta P') = \frac{d}{dt}\left(\frac{1}{2}(\Delta P')^2\right) = \Delta P' \cdot \Delta\dot{P}' \qquad \text{(A.4)}$$

Using Figure A-1, we can obtain

$$\Delta P' = P' - P'_m \qquad \text{(A.5)}$$

which implies

$$\Delta\dot{P}' = \dot{P}' - \dot{P}'_m \qquad \text{(A.6)}$$

Because the sample time of the position control is considerably lower than that of the force control, $\dot{P}' \approx 0$, and based on Figure A-1, equation (A.7) can be obtained as follows:

$$\Delta\dot{P}' = -\,{}^{B}\dot{T}_P(\theta) \qquad \text{(A.7)}$$

Because of the definition of the forward kinematic function and the Jacobian, the following expression can be given:

$${}^{B}\dot{T}_P(\theta) = J(\theta) \cdot \dot{\theta} \qquad \text{(A.8)}$$

By substituting in (A.7) and considering

$$\dot{\theta} = J^{-1}(\theta) \cdot K_j \cdot \Delta P' \qquad \text{(A.9)}$$

(A.10) can be obtained

$$\Delta\dot{P}' = -K_j \cdot \Delta P' \tag{A.10}$$

Therefore, the expression that achieves (A.3) can be obtained from (A.4) and (A.10) as follows:

$$\dot{V}(\Delta P') = -K_j \cdot (\Delta P')^2 \tag{A.11}$$

B. Hidden Markov Model

B.1. Introduction

A Hidden Markov Model (HMM) is a probabilistic approach that is used to model stochastic processes. A process is modelled as a sequence of *hidden* states that have a probabilistic relationship with the measured *observable* information from the process. This information can be discrete or continuous, clear or noisy, etc. Therefore, the HMM is a useful tool for recognizing patterns that have been previously encoded in an HMM as well as the *hidden* state of a sequence of *observable* measurements. In practice, this model has been used in multiple fields such as voice recognition, computational biology, and telecommunications. Therefore, this appendix presents the theoretical aspects of this model that have been used in this thesis by describing an HMM and its parameters (B.2), as well as the algorithms used (B.3 to B.5).

B.2. Description of an HMM

An HMM represents a stochastic process. It is composed of a finite number of u states, where each state (q_i) represents a basic action with observable features (e_i). Additionally, there is a probabilistic relationship between the states, which indicates the probability of jumping from one to another state. This relationship defines the HMM topology, and it is represented by the state transition distribution matrix A. The probabilistic relationship between the observable features (e_i) and each state (q_i) is represented by the observation symbol probability distribution matrix B. Therefore, an HMM can be represented as follows:

$$\lambda = (Q, E, A, B, \pi) \qquad \text{(B.1)}$$

where its parameters are

- $Q = \{q_1, q_2, \dots, q_u\}$ is the set of *hidden* states, where u is the number of states.
- $E = \{e_1, e_2, \dots, e_r\}$ is the discrete observable feature set, where r is the number of observable features in each state.
- A=$\begin{bmatrix} a_{11} & \cdots & a_{1u} \\ \vdots & a_{ij} & \vdots \\ a_{u1} & \cdots & a_{uu} \end{bmatrix}$ where $a_{ij} = P\big(q_j(k+1)\big|q_i(k)\big)$ is the probability of being at q_i at the sample time k and jumping to q_j at the next sample time $k+1$. It comprises the state transition distribution matrix.
- B=$\begin{bmatrix} b_{11} & \cdots & b_{1u} \\ \vdots & b_{ij} & \vdots \\ b_{r1} & \cdots & b_{ru} \end{bmatrix}$ where $b_{ij} = P(e_i(k)|q_j(k))$ is the probability of observing e_i and being at q_j at the same sample time k, which comprises the observation distribution probability matrix.
- $\pi = [\pi_1, \dots, \pi_u]$ where $\pi_i = P(q_i(1))$ denotes the initial state distribution vector.

An example of a HMM topology using this configuration is presented in Figure B-1, where each node represent a state q_i; arrows between the nodes represent the elements of the state transition distribution matrix; and arrows between the nodes and an observable feature e_i represent the elements of the observation distribution probability matrix.

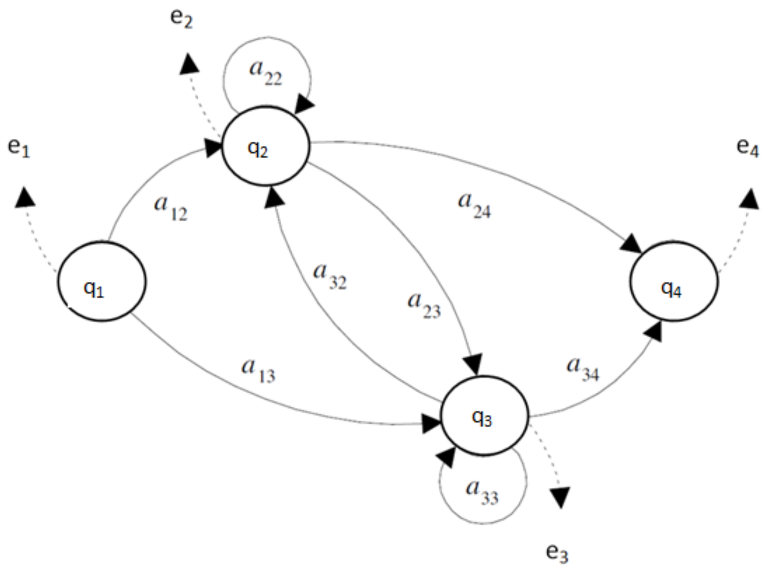


Figure B-1. Example of a four-state HMM

Once an HMM and its parameters have been defined, there are three key issues that have been solved and are useful in real world applications (Rabiner, 1989):

1) *Evaluation*: Determine the probability that an observed sequence of features belongs to an HMM. This is solved using the forward-backward algorithm, which has been used in this thesis to detect the gesture that is being performed.

2) *Decoding*: Given an HMM and an observation sequence of features, generate the states sequence. This is solved using the Viterbi algorithm. Hence, a *hidden* state sequence based on the observation features sequence can be obtained, where each state provides information about the stage of the gesture that is being performed and whether soft tissue interaction occurs.

3) *Training*: Adjust an HMM to maximize the probability that several observed sequences of features belong to that HMM. The Baum-Welch algorithm is used to solve this issue. It has been used in this thesis to generate an HMM for each trained gesture.

These problems are solved using the following algorithms.

B.3. Forward-Backward Algorithm

The evaluation problem consists of calculating $P(\hat{E}(k)|\lambda)$, where $\hat{E}(k) = e(1), e(2), \dots, e(j), \dots, e(k); e(j)\epsilon E$ is an observation sequence. The easiest method for calculating it is to enumerate every state sequence of the k dimension, which is equal to the dimension of the observation sequence $\hat{E}(k)$. Thus, a state sequence can be considered as follows:

$$\hat{Q}(k) = q(1), q(2), \dots, q(j), \dots, q(k); q(j)\epsilon Q \tag{B.2}$$

where $q(1)$ is the initial state. Therefore, the probability of the observation sequence $\hat{E}(k)$ matches with the state sequence $\hat{Q}(k)$ for a given HMM λ, which can be calculated as follows:

$$P(\widehat{E}(k)|\widehat{Q}(k),\lambda) = \prod_{j=1}^{k} P(e(j)|q(j),\lambda) \tag{B.3}$$

Taking into account that each observation is independent, this likelihood can be calculated as follows:

$$P(\widehat{E}(k)|\widehat{Q}(k),\lambda) = b_{q(1),e(1)} \cdot b_{q(2),e(2)} \cdot \dots \cdot b_{q(k),e(k)} \tag{B.4}$$

Additionally, the probability of a state sequence belonging to an HMM can be represented as follows:

$$P(\widehat{Q}(k)|\lambda) = \pi_{q(1)} \cdot a_{q(1)q(2)} \cdot \dots \cdot a_{q(k-1)q(k)} \qquad \text{(B.5)}$$

Therefore, the probability of an observation sequence and a state sequence belonging to a HMM together can be expressed as the product of (B.4) and (B.5) as follows:

$$P(\widehat{E}(k), \widehat{Q}(k)|\lambda) = P(\widehat{E}(k)|\widehat{Q}(k), \lambda) \cdot P(\widehat{Q}(k)|\lambda) \qquad \text{(B.6)}$$

Using this equation to calculate each sequence of possible states for a given observation sequence and HMM, (B.7) can be obtained.

$$P(\widehat{E}(k)|\lambda) = \sum_{\forall Q} P(\widehat{E}(k)|\widehat{Q}(k), \lambda) \cdot P(\widehat{Q}(k)|\lambda) \qquad \text{(B.7)}$$

This proposed method requires a computer complexity of $(2 \cdot k^2 \cdot u)$. To improve it, the *forward-backward* algorithm was proposed. This algorithm is based on dynamic programming to efficiently compute the required probability. It uses two intermediate variables: the forward variable $\alpha_j(i)$, which denotes the probability of the partial sequence of observations $e(1), e(2), \dots, e(j)$ and the state q_i for a HMM λ; and the backward variable $\beta_j(i)$, which is the probability of the partial sequence of observations from j+1 until the end of the sequence for a given state q_i and an HMM λ. Both equations are represented as follows:

$$\alpha_j(i) = P(e(1), e(2), \dots, e(j), q(j) = q_i|\lambda) \qquad \text{(B.8)}$$

$$\beta_j(i) = P(e(j+1), e(j+2), \dots, e(k), q(j) = q_i|\lambda) \qquad \text{(B.9)}$$

Forward variable calculation

The forward variable $\alpha_j(i)$ calculation can be performed by induction following these steps: initialization (B.10), induction (B.11) and conclusion (B.12).

$$\alpha_1(i) = \pi_i b_i\big(e(1)\big)\ 1 \le i \le u \qquad \text{(B.10)}$$

$$\alpha_{j+1}(i) = \left[\sum_{n=1}^{u} \alpha_t(n) a_{ni}\right] \cdot b_{ie(j+1)}\ 1 \le i \le u\ 1 \le j \le k-1 \qquad \text{(B.11)}$$

$$P(\widehat{E}(k)|\lambda) = \sum_{i=1}^{u} \alpha_k(i) \qquad \text{(B.12)}$$

Backward variable calculation

The calculation of $\beta_j(i)$ can be performed similar to that calculated previously: initialization (B.13), induction (B.14) and conclusion (B.15):

$$\beta_1(i) = 1\ 1 \leq i \leq u \tag{B.13}$$

$$\beta_j(i) = \sum_{n=1}^{u} a_{in} \cdot b_{ne(j+1)} \cdot \beta_{j+1}(n)\ j = k-1, k-2, \dots, 1 \tag{B.14}$$

$$P(\widehat{E}(k)|\lambda) = \sum_{i=1}^{u} \beta_k(i) \tag{B.15}$$

In both cases, the computer complexity is $u^2 \cdot l$.

B.4. Viterbi Algorithm

There is no unique optimized sequence of states $\hat{Q}(k)$ for an observation sequence $\hat{E}(k)$ and an HMM λ (decoding problem). The Viterbi algorithm solves this issue by dynamic programming, i.e., it finds an optimized state sequence for a given observation sequence and an HMM. To find this sequence, $\delta_j(i)$ is defined as the most likelihood sequence of states for an observation sequence in the instant j, considering only the first j observations and ending in the state q_i (B.16).

$$\delta_j(i) = \max_{q(1),q(2),\dots,q(j-1)} P(\,q(1), q(2), \dots, q(j) = i, e(1), e(2), \dots, e(j)|\lambda) \tag{B.16}$$

By mathematical induction, (B.17) can be obtained as follows:

$$\delta_{j+1}(i) = \left[\max_{n=1,\dots,u} \delta_j(i) a_{ni}\right] \cdot b_{ie(j+1)}\ j = 1, \dots, k-1 \tag{B.17}$$

Taking this equation into account, the Viterbi algorithm consists of the following steps, where ψ_j is a vector that stores the parameters that maximize (B.17).

- Initialization:

$$\delta_1(i) = \pi_i b_{ie(1)} \; 1 \leq i \leq u \tag{B.18}$$

$$\psi_1(i) = 0 \; 1 \leq i \leq u \tag{B.19}$$

- Recursion:

$$\delta_j(i) = \max_{1\leq n\leq u}\left[\delta_{j+1}(n)a_{ni}\right] \cdot b_{ie(j)} \; 2 \leq j \leq k \; 1 \leq i \leq u \tag{B.20}$$

$$\psi_j(i) = \arg\max_{1\leq n\leq u}\left[\delta_{j+1}(n)a_{ni}\right] \; 2 \leq j \leq k \; 1 \leq i \leq u \tag{B.21}$$

- Termination:

$$q_k^* = \arg\max_{1\leq i\leq u}[\delta_k(i)] \tag{B.22}$$

- Backward state sequence:

$$q_j^* = \psi_{j+1}\left(q_{j+1}^*\right) \; j = k-1, k-2, \dots, 1 \tag{B.23}$$

B.5. Baum-Welch Algorithm

This algorithm is used to train an HMM by estimating its parameters (A, B, π) to maximize $P(\hat{E}(k)|\lambda)$ using a previously obtained observation sequence $\hat{E}(k)$ during the training stage. There is no known method that calculates the HMM parameters using an analytical method. Therefore, the Baum-Welch algorithm is used to estimate HMM parameters by taking into account that $P(\hat{E}(k)|\lambda)$ is locally maximized using an iterative method, which means that the initial parameters are needed. To perform the re-estimation of these parameters, an intermediate variable $\varphi_j(m,n)$ is defined in (B.24), which represents the likelihood of being in state *m* at the instant time *j* and being in state *n* in the instant *j+1* given a HMM λ and an observation sequence $\hat{E}(k)$.

$$\varphi_j(m,n) = P(q_j = q_m, q_{j+1} = q_n|\hat{E}(k), \lambda) \tag{B.24}$$

This equation can be represented by Figure B-2.

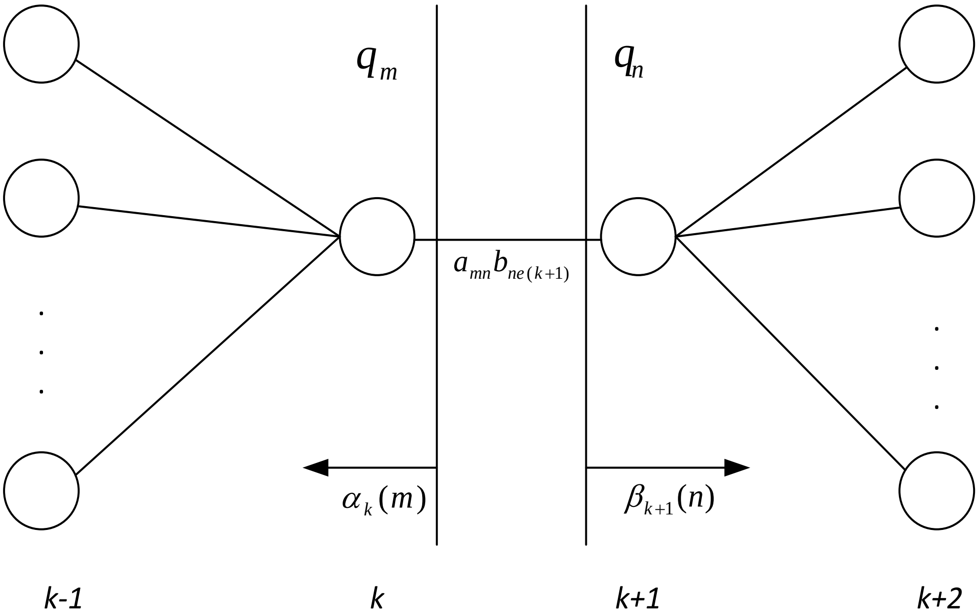


Figure B-2. Required sequence of operations when the HMM is in the *m* state in the instant *k* and state *n* in *k+1*

Using the forward $\alpha_j(n)$ and backward $\beta_j(n)$ variables, equation (B.24) can be defined as follows:

$$\varphi_j(m,n) = \frac{\alpha_j(m)a_{mn}b_{ne(k+1)}\beta_j(n)}{P(\hat{E}(k)|\lambda)} = \frac{\alpha_j(m)a_{mn}b_{ne(k+1)}\beta_j(n)}{\sum_{m=1}^{u}\sum_{n=1}^{u}\alpha_j(m)a_{mn}b_{ne(k+1)}\beta_{j+1}(n)} \qquad \text{(B.25)}$$

By considering that $\gamma_j(n)$ is the probability of being in state *n* at the instant in time *j*, $\gamma_j(n)$ can be calculated using the observation sequence and the HMM as follows:

$$\gamma_j(n) = \sum_{m=1}^{u} \varphi_j(n,m) \qquad \text{(B.26)}$$

By adding $\gamma_j(n)$ for each instant in time *k*, as stated in (B.27), the number of times the state q_n is visited during the observation sequence is obtained, which denotes the number of transitions from the state q_n.

$$\sum_{j=1}^{k-1} \gamma_j(n) \qquad \text{(B.27)}$$

Similarly, the sum of $\varphi_j(m,n)$ in each instant j (B.28) can be interpreted as the number of transitions from q_m to q_n.

$$\sum_{j=1}^{k-1} \varphi_j(m,n) \tag{B.28}$$

Using these equations, an HMM parameter re-estimation method can be iteratively obtained, as presented in (B.29), (B.30) and (B.31):

$$\bar{\pi}_n = \gamma_1(n) \tag{B.29}$$

$$\bar{a}_{mn} = \frac{\sum_{j=1}^{k-1} \varphi_j(m,n)}{\sum_{j=1}^{k-1} \gamma_j(n)} \tag{B.30}$$

$$\bar{b}_{mn} = \frac{\sum_{\substack{j=1 \\ e(j)=e(n)}}^{k} \gamma_j(m)}{\sum_{j=1}^{k-1} \gamma_j(m)} \tag{B.31}$$

It should be noted that this re-estimation method is limited to the stochastic constraints for the HMM parameters as follows:

$$\sum_{n=1}^{u} \bar{\pi}_n = 1 \tag{B.32}$$

$$\sum_{n=1}^{u} \bar{a}_{mn} = 1\ 1 \leq m \leq u \tag{B.33}$$

$$\sum_{n=1}^{r} \bar{b}_{mn} = 1\ 1 \leq m \leq u \tag{B.34}$$

C. Vector Quantization

As stated in Appendix B, when HMMs are used, a discrete observable features set is required. Thus, when sensory information (continuous) is used to encode and recognize gestures, a quantization technique, which translates continuously measured information into discrete symbols (clustering), is needed. Vector Quantization (VQ) is a technique for signal processing that was originally used for data compression; it has also been used to obtain a set of discrete symbols for speech recognition (Rabiner et al., 1983) and gesture recognition (Mitra and Acharya, 2007) using HMM.

Assuming a training sequence $\hat{\vartheta}= \vartheta_1, \vartheta_1, \dots, \vartheta_n$, which is a good representation of the performed movements during the reproduction of a gesture, VQ determines the optimum set of discrete observable feature symbols $E = \{e_1, e_2, \dots, e_r\}$, which accomplish that for a given ϑ_i, it can be replaced by the closest observable feature point e_j, thus minimizing the average distortion over the entire training sequence as (C.1), where $d(\vartheta_i, \mu_j)$ denotes the Euclidean distance between ϑ_i and the centre μ_j of the cluster e_j.

$$\overline{D} = \frac{1}{n}\sum_{i=1}^{n} \min_{1\le j\le r} d(\vartheta_i, \mu_j) \qquad \text{(C.1)}$$

Therefore, equation (C.2) needs to be solved. In this equation, c_j represents the set of datapoints that belong to cluster e_j.

$$\arg\min_{c} \sum_{j=1}^{r} \sum_{\vartheta\in c_j} d(\vartheta, \mu_j) = \arg\min_{c} \sum_{j=1}^{r} \sum_{\vartheta\in c_j} \left\|\vartheta, \mu_j\right\|_2^2 \qquad \text{(C.2)}$$

The solution of this equation is not trivial in terms of computer complexity (NP-hard); therefore, the k-means algorithm (Lloyd, 1982), also known as Lloyd's algorithm, can determine a local minimum in a heuristic way by following these steps:

1. Initialize the centres of the feature symbol set μ_j:

$$\mu_j = some\ value, j = 1, \dots, r \qquad \text{(C.3)}$$

2. Include the data points into the nearest cluster c_j of each feature symbol:

$$c_j = \{i: d(\vartheta_i, \mu_j) \leq d(\vartheta_i, \mu_l), l \neq j, i = 1, \dots, n\} \qquad \text{(C.4)}$$

3. Set the centre of each symbol μ_j to the mean of all datapoints assigned to each cluster c_j:

$$\mu_j = \frac{1}{|c_j|} \sum_{i \in c_j} \vartheta_i \, , \forall j \qquad \text{(C.5)}$$

   In C.5, $|c_j|$ represents the number of elements in c_j.

4. Repeat steps 2 and 3 until convergence.

Once the centres of each symbol have been found, partitioning of a plane can be performed using a Voronoi diagram, which represents the regions based on the distance to specific points. Figure C-1 presents a graphical example using a Voronoi diagram of the training sequence to accomplish (C.2) with $r = 5$.

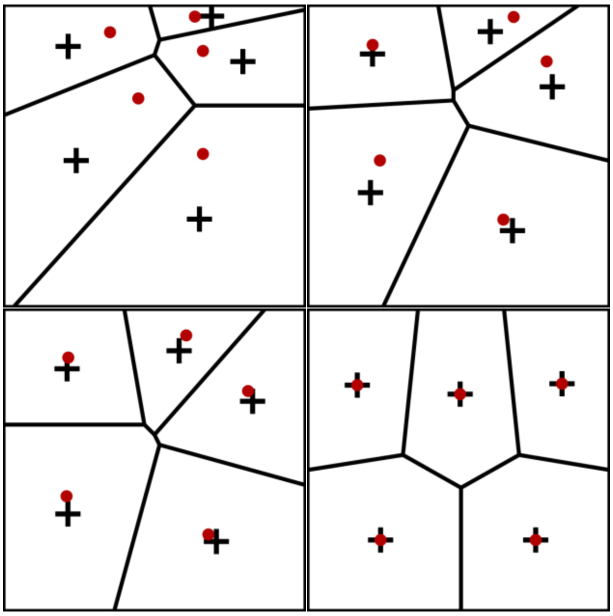


Figure C-1. Example of the evolution of the K-means algorithm using a Voronoi diagram

Finally, once the region that represents each feature symbol has been obtained, any point can be assigned to a feature symbol, which will be used to represent an observable feature in an HMM.

D. Learning from Demonstration Functions

This appendix defines the two functions (4.2) and (4.4) defined in Chapter 4. The first one is used to encode a GMM from previous demonstrations through the Expectation-Maximization algorithm, and the second one is used to generate the haptic guidance references $\vec{g}$ using Gaussian Mixture Regression (GMR).

D.1. Expectation-Maximization Algorithm for GMM

A GMM is defined as $\rho = \{\pi_n, \mu_n, \Sigma_n\}_{n=1}^{N}$, where N is the number of Gaussians, π_n is the prior probability, μ_n is the mean and Σ_n is the covariance matrix of each Gaussian n. The probability that a tuple ξ, belongs to ρ can be given as follows:

$$\mathcal{P}(\xi) = \sum_{n=1}^{N} \mathcal{P}(n)\mathcal{P}(\xi|n) \qquad \text{(D.1)}$$

where $\mathcal{P}(n)$ is the prior probability (D.2); and $\mathcal{P}(\xi|n)$ is a conditional probability density function, whose parameters are defined in (D.3).

$$\mathcal{P}(n) = \pi_n \qquad \text{(D.2)}$$

$$\mathcal{P}(\xi|n) = \mathcal{N}(\xi;\mu_n,\Sigma_n) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_n|}}e^{-\frac{1}{2}((\xi-\mu_n)^T\Sigma_n{}^{-1}(\xi-\mu_n))} \qquad \text{(D.3)}$$

Using these equations, the log-likelihood of a training sequence $\hat{\xi}(k)$, which appends all of the training sequences $\hat{\xi}_{1\ldots u}$, belongs to ρ and can be obtained as follows:

$$\mathcal{L}\big(\hat{\xi}(k)\big) = \sum_{j=1}^{k}\log\big(\mathcal{P}(\xi(j))\big) \qquad \text{(D.4)}$$

where each $\mathcal{P}(\xi(j))$ is obtained from (D.1). Therefore, the objective is to find a set of GMM parameters that maximize $\mathcal{L}\big(\hat{\xi}(k)\big)$. Thus, the parameter estimation function can be defined as follows:

$$\rho = EM(\hat{\xi}(k),N) \qquad \text{(D.5)}$$

This function uses the EM algorithm to adjust a GMM composed of N Gaussians to the training sequence $\hat{\xi}(k)$. This algorithm starts with each Gaussian initialized with random parameters $\{\pi_n^0,\mu_n^0,\Sigma_n^0\}$ that can be adjusted to the training sequence $\hat{\xi}(k)$ iteratively as follows:

- Expectation step:

$$\mathcal{P}(n|\xi(j))^u = \frac{\pi_n^u\mathcal{N}(\xi(j);\mu_n^u,\Sigma_n^u)}{\sum_{i=1}^{N}\pi_i^u\mathcal{N}\big(\xi(j);\mu_i^u,\Sigma_i^u\big)} \qquad \text{(D.6)}$$

$$E_n^u = \sum_{j=1}^{k}\mathcal{P}(n|\xi(j)) \qquad \text{(D.7)}$$

- Maximization step:

$$\pi_n^{u+1} = \frac{E_n^u}{k} \qquad \text{(D.8)}$$

$$\mu_n^{u+1} = \frac{\sum_{j=1}^{k}\mathcal{P}(n|\xi(j))^u\xi(j)}{E_n^u} \qquad \text{(D.9)}$$

$$\Sigma_n^{u+1} = \frac{\sum_{j=1}^{k}\mathcal{P}(n|\xi(j))^u(\xi(j)-\mu_n^{u+1})(\xi(j)-\mu_n^{u+1})^T}{E_n^u} \qquad \text{(D.10)}$$

In these equations, $\mathcal{P}(n|\xi(j))$ is defined as the posterior probability that is obtained from the Bayes theorem as (D.11), E_n is the sum of posterior likelihoods that has been used to simplify the notation, and *u* represents the iteration number.

$$\mathcal{P}(n|\xi(j)) = \frac{\mathcal{P}(n)\mathcal{P}(\xi(j)|n)}{\sum_{i=1}^{N}\mathcal{P}(i)\mathcal{P}(\xi(j)|i)} \tag{D.11}$$

The iteration ends when the difference in log-likelihoods (D.4) between the iterations is less than a predefined threshold C:

$$\frac{\mathcal{L}\left(\hat{\xi}(k)\right)^{u+1}}{\mathcal{L}\left(\hat{\xi}(k)\right)^{u}} < C \tag{D.12}$$

D.2. Gaussian Mixture Regression

The Gaussian Mixture Regression is used to calculate the most likely values for the output variables of a tuple ξ for an encoded GMM by specifying the values of the input variables. Thus, ξ can be defined as follows:

$$\xi = (\xi^{i}, \xi^{o}) \tag{D.13}$$

where ξ^{i} represents the input variables, whose information can be retrieved from the sensory information; and ξ^{o} represents the output variables, which are used to generate an output that depends on the encoded GMM ρ and the sensory information ξ^{i}. Thus, the GMR function can be defined as follows:

$$\xi^{o} = GMR(\rho, \xi^{i}) \tag{D.14}$$

which has been used in Section 4.2.2 to calculate the haptic guidance references $\xi^{o} = \vec{g}$ from the encoded ρ and the obtained interaction measurements $\xi^{i} = \left(\vec{f}, \vec{\tau}\right)$.

To obtain the GMR of ρ, the parameters of a Gaussian n that belongs to ρ can be represented as follows:

$$\mu_n = \begin{bmatrix} \mu_n^{i} \\ \mu_n^{o} \end{bmatrix},\ \Sigma_n = \begin{bmatrix} \Sigma_n^{i} & \Sigma_n^{io} \\ \Sigma_n^{oi} & \Sigma_n^{o} \end{bmatrix} \tag{D.15}$$

Using this representation, the expected distribution of ξ^{o} for a given input variable ξ^{i} and a Gaussian distribution n can be defined as follows:

$$P(\xi^{o}|\xi^{i}, n) \sim \mathcal{N}(\mu'_n, \Sigma'_n) \tag{D.16}$$

where

$$\mu'_n = \mu_n^o + \Sigma_n^{oi}\left(\Sigma_n^i\right)^{-1}(\xi^i - \mu_n^i) \tag{D.17}$$

$$\Sigma'_n = \Sigma_n^o - \Sigma_n^{oi}\left(\Sigma_n^i\right)^{-1}\Sigma_n^{io} \tag{D.18}$$

By considering the GMM as N Gaussian distributions, the expected distribution of ξ^o for a given ξ^i can be estimated as follows:

$$P(\xi^o|\xi^i) \sim \sum_{n=1}^{N} P\left(n\middle|\xi^i\right)\mathcal{N}(\mu'_n, \Sigma'_n) \tag{D.19}$$

where $P\left(n\middle|\xi^i\right)$ can be given as the probability of an observed input belonging to the Gaussian distribution n as follows:

$$P\left(n\middle|\xi^i\right) = \frac{P(n)P(\xi^i|n)}{\sum_{j=1}^{N} P(j)P(\xi^i|j)} = \frac{\pi_n\mathcal{N}\left(\xi^i; \mu_n^i, \Sigma_n^i\right)}{\sum_{j=1}^{N}\pi_j\mathcal{N}\left(\xi^i; \mu_j^i, \Sigma_j^i\right)} \tag{D.20}$$

Therefore, the conditional expectation ξ^o can be estimated through the Gaussian distribution $\mathcal{N}\left(\mu'_n, \Sigma'_n\right)$ as follows:

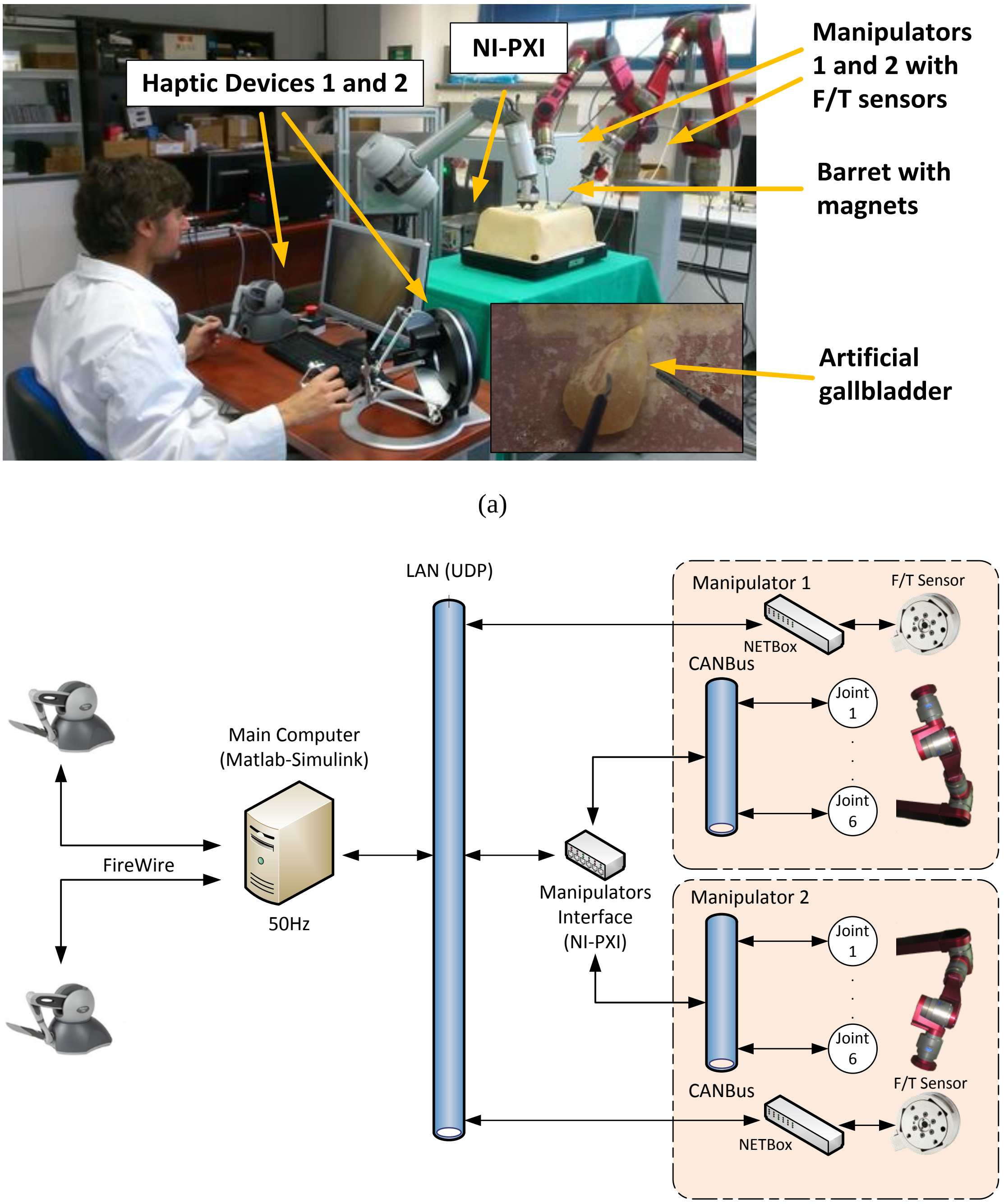
$$\xi^o = \sum_{n=1}^{N} P\left(n\middle|\xi^i\right)\mu'_n \tag{D.21}$$

Finally, it should be noted that equations (D.20) and (D.21) are used to estimate the haptic guidance references in real time, which indicates that the execution of these equations is limited to the time constraints of a real-time system. Hence, computer complexity could help to clarify the required processing time. After analysing these equations, there is one sum in (D.21) and another sum in the divisor of (D.20). Although (D.20) is calculated inside the sum in (D.21), its divisor can be previously calculated because it is independent of the sum in (D.21). Therefore, the processing time of the GMR function increases linearly depending on the number of Gaussians, which limits the maximum number of Gaussians that can be used depending on the available processing time.

E. The Cisobot Platform

The proposed navigation method has been implemented in the experimental robotic surgical platform developed by the Universidad de Málaga, as shown in Figure E-1.a. This platform is divided into two subsystems: the user handles two haptic devices that teleoperate two manipulators using the proposed navigation method (teleoperation subsystem); and a Barret manipulator is used to control a camera inside the patient's abdomen that is held by magnetic interaction. To simulate the patient's body, an abdominal simulator is used with a 4x7 cm artificial gallbladder inside. The primary objective of this platform is research into new cognitive methods and algorithms for robotic surgery.

The teleoperation subsystem, whose architecture is depicted in Figure E-1.b, has been used to validate the proposed navigation method. It is composed of a main computer, which executes the proposed navigation method (Matlab/Simulink), two haptic devices, which are used to provide position references to the manipulators and force feedback to the surgeon, and two customized manipulators and two multi-axial F/T sensors, which measure the exerted forces and torques throughout the instruments. These manipulators are controlled by real-time hardware (NI-PXI) that provides an interface between the main computer using UDP and each joint controller through CANbus. Each F/T sensor is connected to a *Netbox*, which is the interface that provides UDP and is used to send the measurements to the main computer. Finally, the haptic devices are connected to the main computer using a Firewire interface. All of these devices are described in more detail in the following subsections.



(a)

(b)

Figure E-1. CISOBOT Platform: (a) Image of the whole platform; (b) Teleoperation subsystem architecture

E.1. Robotnik Modular Arm

The CISOBOT platform is composed of two customized manipulators (Figure E-2) that are manufactured by Robotnik Automation S.L. Each manipulator has 6 DoF $(\theta_1, \dots, \theta_6)$, whose joints consist of a rotational axis in an RRR-RRR with a spherical wrist configuration. This allows a decoupled kinematic configuration in which the first three joints are used to place the end effector in a concrete position and the last three joints (spherical wrist) are used to configure the end effector orientation.

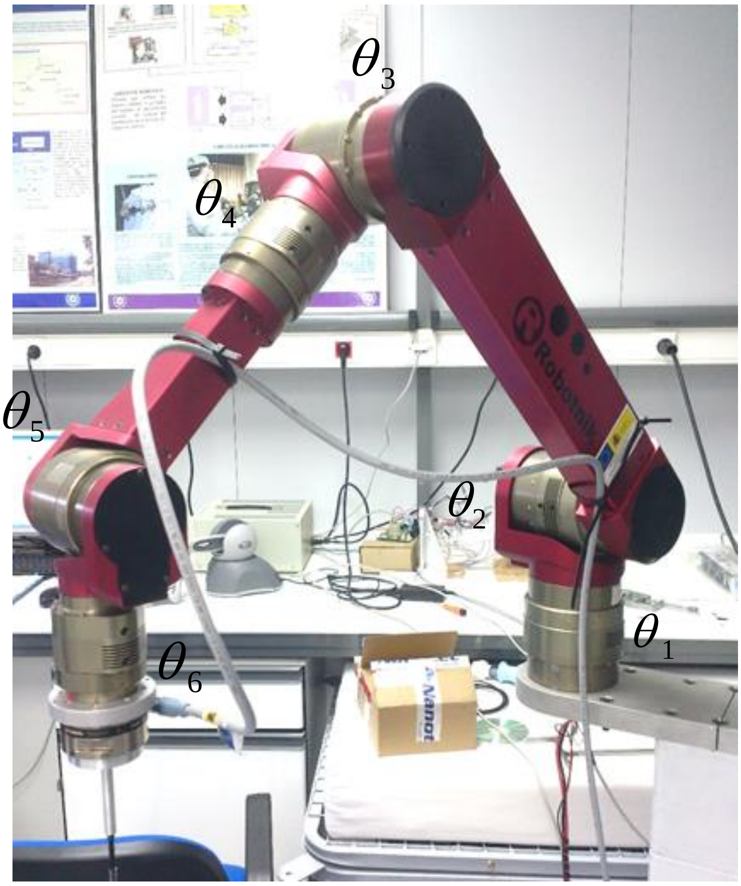


Figure E-2. Robotnik Modular Arm

The rotational joint actuators are comprised of PowerCube servomotors (Figure E-3) from Schunk Corp., whose dimensions depend on their location, i.e., torque requirements. Each PowerCube includes a position and velocity PID controller that is configured to receive position and/or velocity references, thus generating a velocity profile that takes these references into account. Therefore, each joint is responsible for accomplishing the referenced position. Moreover, each PowerCube provides sensory information, such as position, velocity and state of the joint. Communication with the actuators are performed by CANbus. This interface guarantees real-time communication and allows the use of a unique bus for each joint. Hence, a joint decoupled control can be performed, and the position and velocity references can be sent to each joint simultaneously.

Figure E-3. Schunk PowerCube actuator

Although this manipulator was conceived for industry, its features make it extremely versatile. It is lightweight (only 19kg), has a wide workspace owing to its kinematic configuration, and produces a rapid response. Thus, each PowerCube is able to reach a maximum speed between 57 degrees/sec. and 300 degrees/sec. depending on the actuator dimensions. Moreover, it needs only a DC 24 V power supply that can be directly provided from batteries, which is useful if it is going to be used in an operating room.

E.2. PXI Real-Time Hardware

PCI eXtension for Instrumentation is an open industry specification that was developed by National Instrument in 1998. This specification defines a rugged PC-based platform (Figure E-4), whose primary advantages are its modularity, high performance and low cost deployment of instrumentation and/or automation. These advantages allow the use of this platform for research and industrial purposes.

Figure E-4. PXI platform

In the CISOBOT platform, a PXI has been used as a manipulators interface, i.e., this system controls the communications between each joint of each manipulator and the main computer. Moreover, the use of this system allows standard UDP communications between this device and the main computer, which facilitates compatibility with the software used in the main computer (described in Section E.5). The PXI has been primarily used to provide position or velocity references and obtain the measured position and velocity from the manipulator joints.

The development of the manipulator interface has been performed using LabVIEW. This is a graphical environment that is used to develop software for PCs and specific NI hardware, such as the NI-PXI. Figure E-5 depicts an example of the application that has been developed to communicate between the main computer and the manipulators. This example retrieves the position and velocity of each joint using the CANbus interface and sends it to the main computer through the UDP.

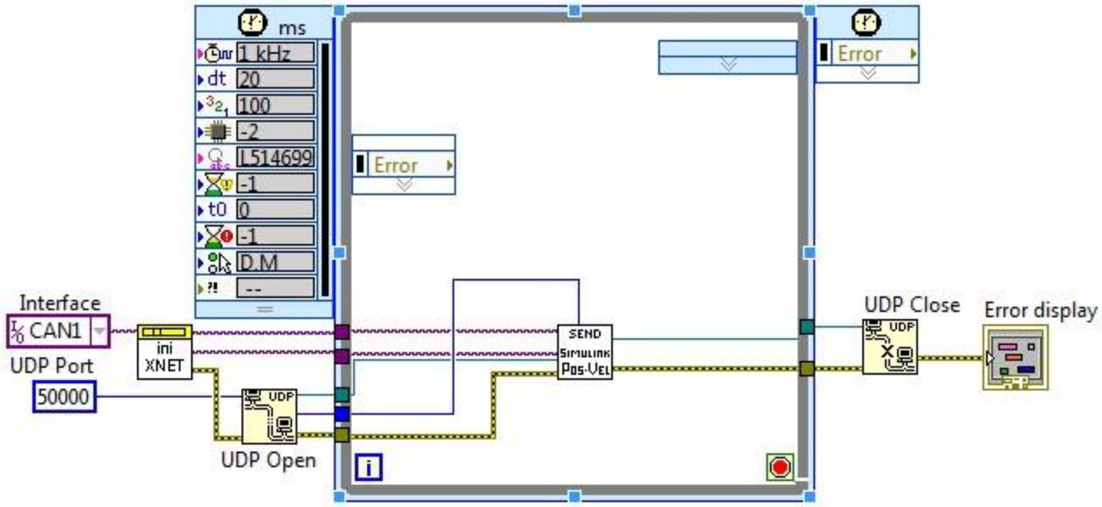


Figure E-5. LabVIEW-based application

E.3. Force-Torque Sensor

As stated in Chapter 3, a Force-Torque sensor is used to measure the interaction forces throughout the instrument. This sensor is placed between each manipulator end effector and the instrument. Thus, a F/T sensor from ATI Industrial Automation Ltd. has been used. In particular, the Net F/T Gamma SI 65-5 is a sensor that measures the forces and torques in the X, Y and Z-axes,thereby providing six measurements. The dimensions of this sensor are 33 mm height and 75 mm diameter. These dimensions allow the sensor to be placed on the manipulator end effector, and an instrument can be attached to it, as depicted in Figure E-6.

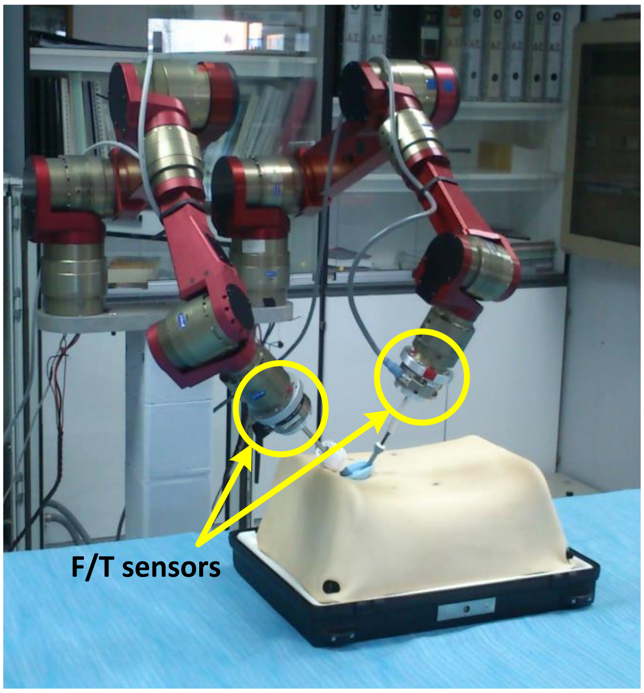


Figure E-6. F/T sensors placed between the manipulator end effector and the instrument

Each F/T sensor is connected to a NET Box device (Figure E-7), which reads analog signals from the F/T sensor and translates them into force and torque measurements that are sent through different communication interfaces. This feature is one of the primary advantages of this type of sensor, including UDP and CANbus interfaces. In the CISOBOT platform, the UDP interface has been used to connect each F/T sensor with the main computer.

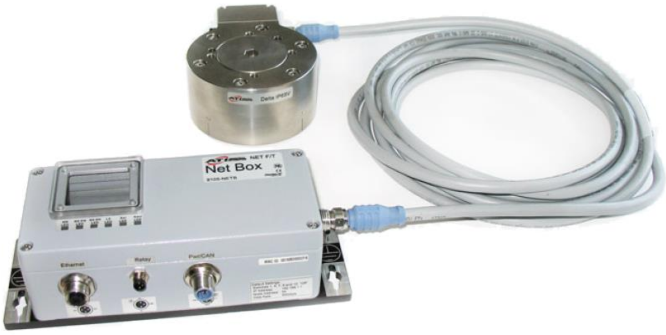


Figure E-7. ATI Net Box connected to a Gamma F/T sensor

E.4. Haptic Device

To teleoperate both manipulators and provide force feedback to the surgeon, two Phantom Omni devices (Figure E-8) from Sensable Inc. have been used. These devices are able to move in six DoF, and their kinematic configuration allows joint decoupling using a spherical wrist that provides the end effector orientation. Moreover, they have two switches that can be programmed for general purposes.

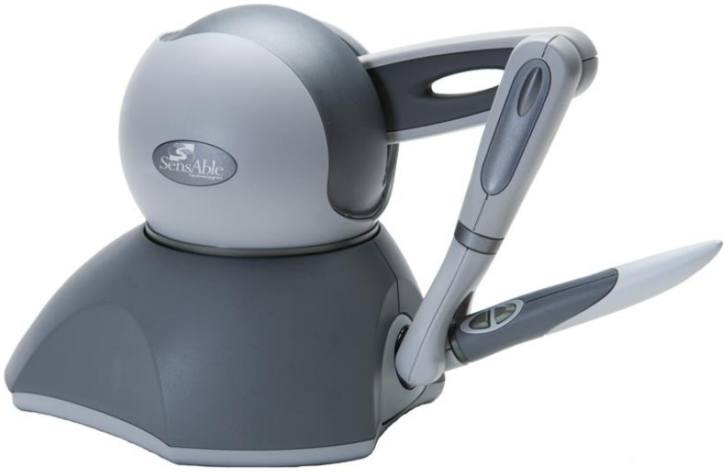


Figure E-8. Phantom Omni haptic device

The Phantom Omni is held by the operator using its end effector, whose shape is similar to a pencil and facilitates teleoperation. Plastic pieces cover its metallic internal structure, thus providing good stability during usage. Its kinematic configuration allows a workspace of 160x120x70 mm, and the end effector can be oriented in any position owing to its wrist configuration. The accuracy between adjacent points is approximately 0.055 mm, and it can provide forces on the three space axes, reaching a peak of 3.3 N and 0.88 N continuously.

Both devices are connected to the main computer using an IEEE1394 Firewire interface. To calibrate the devices, the manufacturer provides software that allows the parameters to be adjusted by calibrating each joint and performing a few tests. Additionally, applications can be developed using the OpenHaptics library, which has been used to develop a simple application that communicates with Simulink (UDP) to provide the position and orientation of the haptic device end effector, receive haptic forces from Simulink and send them to the haptic device.

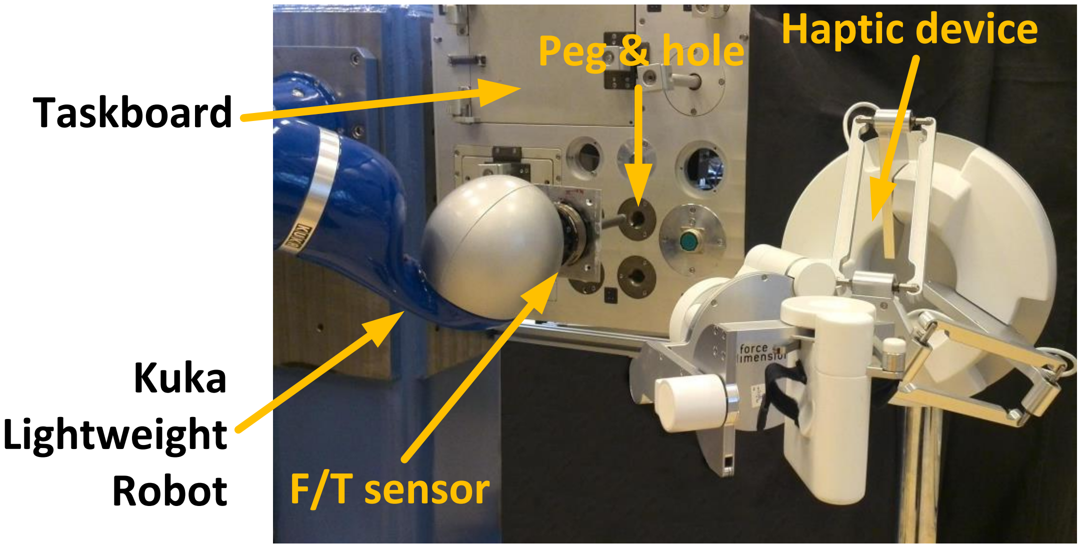
E.5. Main Computer

To deploy the defined control schemes for surgical navigation and perform several experiments, a generic PC was used, comprising an i5 microprocessor, 4GB RAM and Windows 7 OS. The defined control scheme was developed in Matlab-Simulink. Although Simulink was conceived as a simulation environment, it has been expanded towards direct system control by adding hardware support packages with internal tools that compile and build C/C++ applications from Simulink diagrams. Specifically, Real-Time Windows Target (RTWT) has been used to run control schemes on a generic PC. The use of these tools allows rapid development and deployment of control schemes owing to the use of different Matlab toolboxes compared to standard programming languages, such as C/C++, which are directly deployed into Windows or Linux OS. Furthermore, each part of the control scheme can be programmed into independent blocks that are connected to make up the entire control scheme.

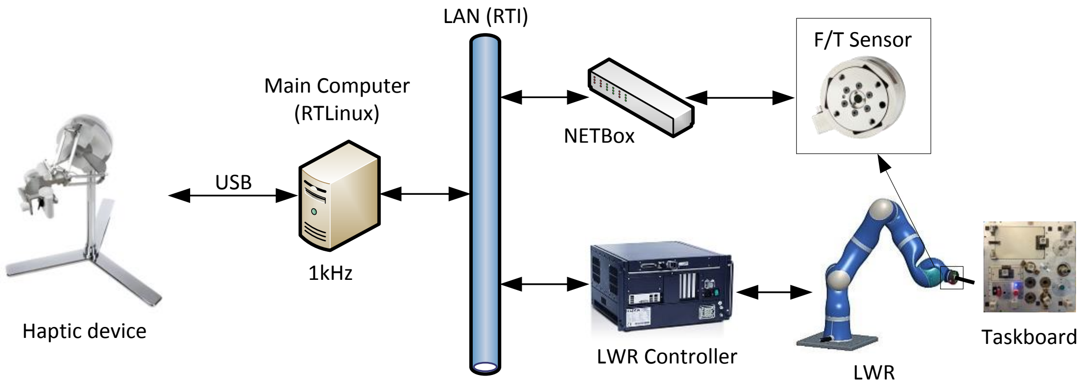
Therefore, the control scheme that was developed in Simulink and executed on the main computer controls and supervises the entire system. Owing to the limitations of the RTWT, all of the communications between devices were performed using a UDP interface. Thus, the main computer is connected to a LAN, where the F/T sensors and the NI-PXI are also connected. This main computer executes a simple C++ application that communicates with the haptic devices by Firewire and Simulink by UDP using the virtual local host network.

F. LWR Taskboard Workcell

The teleoperation system consists mainly of a slave KUKA Lightweight Robot (LWR) and a Sigma.7 haptic master device, as shown in Figure F-1.a. Figure F-1.b shows the software and hardware architecture of the teleoperation system. The taskboard, on the right of the figure, is used to solve generic tasks. It contains different holes, pegs and doors that can be used to validate different control algorithms. The interaction with the taskboard is conducted by a Kuka LWR, which is described below. An ATI F/T sensor (similar to that described in Section E.3) is attached to the manipulator end effector to measure the exerted forces and torques. Both the manipulator and the F/T sensor are connected to the main computer that executes the proposed LfD approach in real time. Additionally, this computer is connected to a Sigma.7 haptic device that is handled by the operator during the reproduction of the task, thus providing guidance forces to the operator. All of the subsystems, except the haptic device, which is connected directly through a USB, communicate over a one gigabyte LAN network with RTI Data Distribution Service middleware, reaching a 1 kHz sample time.



(a)



(b)

Figure F-1. LWR Taskboard Workcell: (a) Teleoperation setup; and (b) SW/HW architecture

F.1. Taskboard

The taskboard used (Figure F-2) is composed of different elements that allow a manipulator to interact with it to solve specific tasks. This taskboard can be used to solve opening door tasks, connector plugging, switching, etc. For the experiments described in this thesis, the peg-in-hole task was selected. This task consists of inserting a peg into a hole in the taskboard.

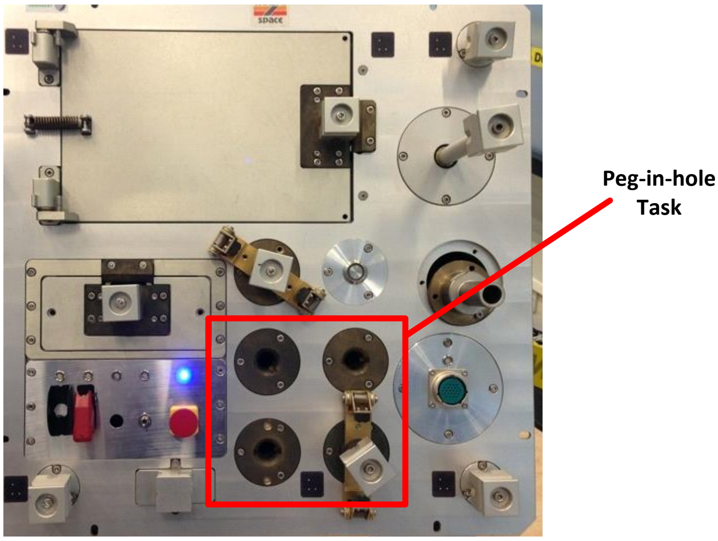


Figure F-2. Taskboard for haptic guidance experiments

F.2. Kuka LightWeight Robot

The manipulator used to perform the experiments was the KUKA LWR robot. It is a specific lightweight robot whose weight is only 22.9 kg. It has 7 DoF with a maximum payload on the end effector of 7 kg. This manipulator has two features that are extremely important for the performed experiments. The first is the ability to perform kinaesthetic movements, which means that an operator can take the end effector and move it freely in space. This feature was useful during the training stage because the operator could train the system to solve the peg-in-hole insertion task by moving the manipulator and using it to measure the position and velocity of the end effector during the movements. The second interesting feature is the active compliance mode allowed by this robot. During the reproduction stage, the robot was programmed with a Cartesian stiffness of $k_{trans} = 500$ Nm/rad and $k_{rot} = 500$ Nm/rad for translations and rotations, respectively. Hence, the exerted forces of the robot when it is interacting were softened. Furthermore, a multi-axial F/T sensor (described in Section E.3) was mounted on the manipulator end effector, and a 155 mm long titanium peg was rigidly placed on the F/T sensor.

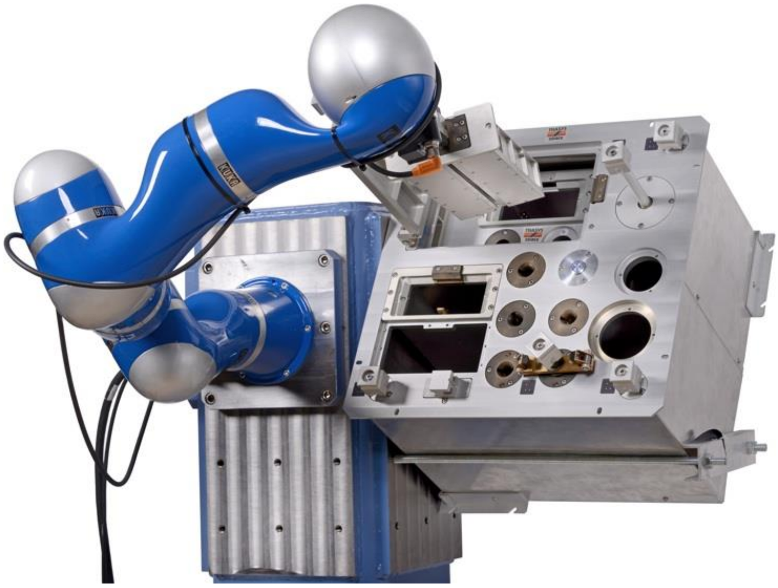


Figure F-3. Kuka Lightweight robot attached to the taskboard

F.3. Main Computer

The main computer is the centre of the architecture. It receives and store information from devices and sends references to the manipulator. This computer is a conventional PC with a 3 GHz i7 microprocessor. The operative system is a Linux distribution that is used to execute C/C++ code in real time by adding the Xenomai Real-time framework. This framework supplements the original Linux kernel with a real time one that works alongside it. Moreover, it includes a real-time operative system API to control the process executions.

The Simulink Coder was used to generate C/C++ code from Simulink diagrams, thus making the development procedure easier. Hence, two applications were developed. The first application was used to retrieve information from the robot and the F/T sensor to train the proposed GMM model using kinaesthetic movements, and the second application was used to teleoperate the robot using the Sigma.7 haptic device, thus providing haptic guidance.

Owing to the low computer complexity of the used algorithms and the features of the used computer and OS, a sample time of 1 ms was used during the experiments.

F.4. Sigma.7 Haptic Device

The Sigma.7 haptic device (Figure F-4) was originally conceived as part of the master console of the MiroSurge robotic system (described in Section 2.2.3). It is manufactured by Force Dimension in cooperation with the German Aerospace Center (DLR).

This device is a seven DoF fully actuated device that uses a 3 DoF parallel mechanism for translational motions, 3 DoF intersecting axis drives for rotational motions, and a grasping unit. This structure allows the kinematics and dynamics of the device to be decoupled (Tobergte et al., 2011). Its kinematic configuration allows a spherical workspace of approximately 120 mm in diameter, and the rotational wrist covers the entire human hand workspace. The haptic forces provide up to 20 N within the translational space and approximately 0.4 Nm of maximum torque from the rotational wrist. Furthermore, a force sensor is placed on the grasping unit that is able to measure up to 8 N. This device was connected to the main computer using a USB interface that is able to reach up to a 4 kHz sample time.

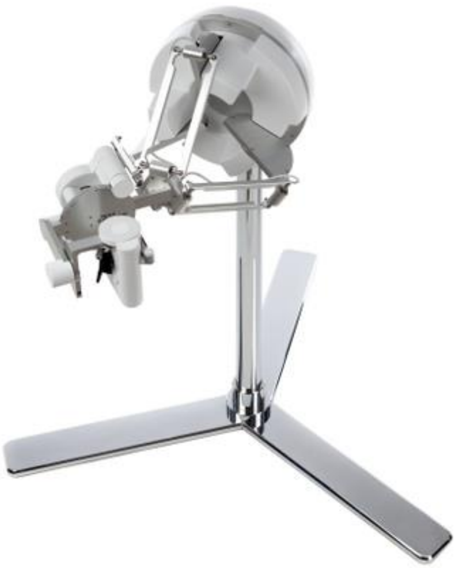


Figure F-4. Sigma.7 haptic device

Glossary of Terms

AOB: Active Observers

BIC: Bayesian Information Criterion

CIS: Computer Integrated Surgery

CISOBOT: Experimental surgical robotic platform developed by the Universidad de Málaga

DARPA: Defense Advanced Research Projects Agency

DLR: German Aerospace Centre

DoF: Degree of Freedoms

DTW: Dynamic Time Warping

EM Algorithm: Expectation-Maximization algorithm

ESA-ESTEC: European Space Agency – European Space Research and Technology Centre

FDA: US Food and Drug Administration

Fulcrum Point: Point at which the trocar is introduced

GMM: Gaussian Mixture Model

GMR: Gaussian Mixture Regression.

HALS: Hand Assisted Laparoscopic Surgery

Haptic Guidance: Use of forces to guide the operator during the reproduction of a task

HMM: Hidden Markov Model

LabVIEW: Programming language from National Instruments

LAN: Local Area Network

Laparotomy: Open surgery

LfD: Learning from Demonstration

LQR: Linear-Quadratic Regulator

LS: Laparoscopic Surgery

LWR Taskboard Workcell: Experimental generic task platform developed by ESA

LWR: Locally Weighted Regression

Machine Learning: Set of algorithms that are used to make predictions or decisions

MIS: Minimally Invasive Surgery

Multi-axial F/T sensor: A sensor that is able to measure forces and torques

Multiport Trocar: A trocar that is used to introduce all of the necessary instruments and laparoscopic camera through the abdomen

NI-PXI: Real-time hardware developed by National Instruments

NOTES: Natural Orifices Trans-luminal Surgery

PED: Prediction from Expert Demonstration

RCM: Remote Centre of Motion

RMS: Root Mean Square

ROS: Robotic Operating System

RTI: Real Time Innovations Inc.

RTWT: Matlab-Simulink Real Time Windows Target

SPAS: Single Port Access Surgery

Trocar: A sharp-pointed surgical object, used to introduce a surgical instrument into the abdomen

UDP: User Datagram Protocol

Virtual Fixtures: Haptic guidance methodology that uses predefined trajectories or forbidden regions to generate guidance forces

VQ: Vector Quantization

Notation

$\{i\} = {}^{j}T_i = \left({}^{j}\vec{p}_i,\ {}^{j}\sigma_i\right)$: reference frame that can be represented by the homogeneous transformation matrix ${}^{j}T_i$ composed of the Cartesian position ${}^{j}\vec{p}_i$ and orientation ${}^{j}\sigma_i$.

A: Initial position of the instrument

B: Final position of the instrument

$\{F'_A\}$: Estimated fulcrum point at the initial position A

$\{F'_B\}$: Estimated fulcrum point at the final position B

$\{F_A\}$: Real fulcrum point at the initial position A

$\{F_B\}$: Real fulcrum point at the final position B

$\vec{f}_{F_B}$: Abominal forces on the fulcrum point at the final position *B*

$\overrightarrow{r_F}$: Distance between $\{F_A\}$ and $\{F_B\}$

K_a: Skin elasticity constant

$\overrightarrow{f_{P_B}}$: Exerted forces on the instrument tip

$\overrightarrow{f_{H_B}}$: End effector exerted forces that can be measured by an F/T sensor

$\overrightarrow{\mu_{H_B}}$: End effector torque that can be measured by an F/T sensor

$\overrightarrow{f_r}$: Abdominal force reference equal to zero

V_{PI}: PI controller output

θ : Manipulator joint positions

$\dot{\theta}$: Manipulator joint velocities

K_J: Matrix that fixes the control scheme dynamics

ϕ_m: Sensory information provided by manipulator *m*

$\vec{f} = (f_x, f_y, f_z)$: Measured forces on instrument

$\vec{\tau} = (\tau_x, \tau_y, \tau_z)$: Measured torques on instrument

$\vec{p} = (p_x, p_y, p_z)$: Measured position on the instrument

Ω: Gesture library for the interaction with soft tissue detection

λ_i: Encoded gesture into an HMM

Q: Set of hidden states

q_i: *Hidden* state

$\hat{Q}(k)$: Sequence of *k* hidden states

A_i: State transition distribution matrix

E: Set of observable features obtained from Vector Quantization

$\hat{E}(k)$: Sequence of observation features

B_i: Observation probability distribution matrix

π_i: Initial state distribution vector

$\hat{I}(k)$: Sequence that indicates whether interaction occurs with soft tissue at each instant

$\hat{\xi}_{iu}(k)$: Haptic guidance training sequence u for the gesture *i*

$\xi(j) = (\xi^i; \xi^o) = \left[\vec{f}(j), \vec{\tau}(j); \vec{h}(j)\right]; 1 \leq j \leq k$: Training sequence tuple at the instant in time *j*

$\vec{h} = (h_x, h_y, h_z)$: Ideal guidance reference

$\vec{g} = (g_x, g_y, g_z)$: Haptic guidance reference

ρ_i: Encoded gesture *i* into a GMM

π: Prior probabilities of a Gaussian

μ_n: Centre of the *n* Gaussian of a GMM

Σ: Covariance of a Gaussian

N: Number of Gaussians into a GMM

D: Dimension of the tuple ξ

References

Abbott, Jake J., Panadda Marayong, and Allison M. Okamura. "Haptic virtual fixtures for robot-assisted manipulation." Robotics research. Springer Berlin Heidelberg, pp. 49-64, 2007.

Adili, A. “Robot-assisted orthopedic surgery”. Surgical Innovation, vol. 11(2), pp. 89-98, 2004.

Anderson B, Moore J. “Optimal Control: Linear Quadratic Methods”. Prentice-Hall, 1989.

Agachan F., Joo J.S., Sher M., Weiss E.G., Nogueras J.J., Wexner S.D. “Laparoscopic colorectal surgery. Do we get wafter?” Surgical Endoscopy, vol. 11 (4), pp. 331-5, 1997.

Aiple M., Schiele A. "Pushing the limits of the CyberGrasp™ for haptic rendering." IEEE International Conference on Robotics and Automation (ICRA), pp. 3541-3546, 2013.

Bardram, J.E. et al. “Phase recognition during surgical procedures using embedded and body-worn sensors”. IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 45-53, 2011.

Bauzano, E., Garcia-Morales, I., del Saz-Orozco, P., Fraile, J. C. and Muñoz, V. F. “A minimally invasive surgery robotic assistant for HALS–SILS techniques”, Computer methods and programs in biomedicine, vol. 112(2), pp. 272-283, 2013.

Bauzano, E.; Estebanez, B.; Garcia-Morales, I.; Munoz, V.F., "Collaborative Human–Robot System for HALS Suture Procedures" Systems Journal, IEEE , vol.PP, no.99, pp.1,10, 2014.

Bauzano, E. “Robot Asistente Semiautónomo de dos Brazos para Intervenciones de Cirugía Laparoscópica”, PhD Thesis, Universidad de Málaga, 2012.

Bassi E., Benzi F. et al. “Characterization of the Dynamical Model of a Force Sensor for Robot Manipulators”, Robot Motion and Control, Lecture Notes in Control and Information Sciences, vol. 396, pp 243-253, Springer-Verlag, 2009.

Berkelman, P. J., Whitcomb, L. L., Taylor, R. H., & Jensen, P. “A miniature microsurgical instrument tip force sensor for enhanced force feedback during robot-assisted manipulation”. IEEE Transactions on Robotics and Automation, vol. 19(5), pp. 917-921, 2003.

Billard, A, Calinon, S, and Dillmann, R. “Learning from Human Demonstration”. Handbook of Robotics: MIT Press, 2013.

Bilmes, J. A. “A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models”. International Computer Science Institute, vol. 4(510), p. 126, 1998.

Boessenkool, H., Abbink, D. A., Heemskerk, C. J., van der Helm, F. C., & Wildenbeest, J. G. “A task-specific analysis of the benefit of haptic shared control during telemanipulation”. IEEE Transactions on Haptics, , vol. 6(1), pp. 2-12, 2013.

Bowyer, S. A., Davies, B. L., & Rodriguez y Baena, F. “Active constraints/virtual fixtures: A survey”. IEEE Transactions on, Robotics, vol. 30(1), pp. 138-157, 2014.

Brenna D. Argall, Eric L., Sauser and Billard A.G. "Tactile Guidance for Policy Adaptation". Foundations and Trends in Robotics: vol. 1(2), pp 79-133, 2011.

Brown, J. D., Rosen, J., Chang, L., Sinanan, M. N., & Hannaford, B. “Quantifying surgeon grasping mechanics in laparoscopy using the blue DRAGON system”. Studies in health technology and informatics, vol. 98, 34-36, 2004.

Butner, S. E., & Ghodoussi, M. “A real-time system for tele-surgery”. 21st International Conference on in Distributed Computing Systems, pp. 236-243, 2001.

Calinon S. "Robot programming by demonstration." Springer handbook of robotics. Springer Berlin Heidelberg. 1371-1394, 2008.

Calinon, S. "Robot programming by demonstration: A probabilistic Approach" EPFL Press, 2009.

Calinon, S., D'halluin, F., Sauser, E. L., Caldwell, D. G., & Billard, A. G. “Learning and reproduction of gestures by imitation”. IEEE Robotics & Automation Magazine, vol. 17(2), pp. 44-54, 2010.

Casals, A., Basomba J., Bergés E., Frigola M. and Amat J. “Sistema Integrado de Posicionado, Visulización y Acotamiento de Áreas de Trabajo en Cirugía Ortopédica”, ROBOT 2011, pp. 367-372, 2011.

Chhatpar, S. R., & Branicky, M. S. “Search strategies for peg-in-hole assemblies with position uncertainty”. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), vol. 3, pp. 1465-1470, 2001.

Chowriappa, A., Wirz, R., Ashammagari, A. R., & Seo, Y. W. "Prediction from expert demonstrations for safe tele-surgery", International Journal of Automation and Computing, vol. 10(6), pp. 487-497, 2013.

Cortesao, R., Zarrad, W., Poignet, P., Company, O., & Dombre, E. "Haptic Control Design for Robotic-Assisted Minimally Invasive Surgery", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 454-459, 2006.

Danwei Wang and N. Hamis McClamroch, "Position and Force Control for Constrained Manipulator Motion: Lyapunov's Direct Method", IEEE Transactions on Robotics and Automation, vol. 9(3), pp. 308-313, 1993.

Ding, J., Xu, K., Goldman, R., Allen, P., Fowler, D., & Simaan, N. "Design, simulation and evaluation of kinematic alternatives for insertable robotic effectors platforms in single port access surgery". IEEE International Conference on Robotics and Automation (ICRA), pp. 1053-1058, 2010.

Eskicorapci S.Y., Teber D., Schulze M., Ates M., Stock C., Rassweiler J.J. "Laparoscopic radical nephrectomy: the new gold standard surgical treatment for localized renal cell carcinoma". The Scientific World Journal, vol. 7, pp. 825-836, 2007.

Estebanez, B. "Diseño e Implantación de un Sistema Multimodal para un Asistente Robótico", PhD Thesis, Universidad de Málaga, 2013.

Evensen G. and Van Leeuwen P.J. "An ensemble Kalman smoother for nonlinear dynamics". Monthly Weather Review, vol. 128(6), pp. 1852-1867, 2000.

Fischer J.E. and Bland K.I. "Mastery on Surgery", Lippincott Williams & Wilkins, vol. 1, 2007.

Fitchinger, G., D. Stoianovici, y R.H. Taylor. "The surgical CAD/CAM paradigm and an implementation for robotically-assisted percutaneuos local therapy", 30th Applied Imagery Pattern Recognition Workshop, pp. 3-8, 2001.

Garcia P., Rosen J., Kapoor C., Noakes M., Elbert G., Treat M., Ganous T., Hanson M., Manak J., Hasser C., Rohler D., Satava R.. "Trauma Pod: a semi-automated telerobotic surgical system". The International Journal of Medical Robotics and Computer Assisted Surgery, vol. 5(2), pp. 136-146, 2009.

Gan, Q.; Harris, C.J. "Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion" IEEE Transactions on Aerospace and Electronic Systems, vol.37(1), pp. 273-279, 2001.

Gomes, P. “Surgical robotics: Reviewing the past, analysing the present, imagining the future”. Robotics and Computer-Integrated Manufacturing, vol. 27(2), pp. 261-266, 2011.

Gu, Y., Thobbi, A., & Sheng, W. “Human-robot collaborative manipulation through imitation and reinforcement learning”. IEEE International Conference on Information and Automation (ICIA), pp. 151-156, 2011.

Guthart, G.S., y J.K. Salisbury. “The Intuitive Telesurgery System: Overview and Application”. IEEE International Conference on Robotics and Automation (ICRA), pp. 618-621, 2000.

Hagn U., Onietschke R. K, Tobergte A., et al. “DLR MiroSurge: a versatile system for research in endoscopic telesurgery”, International Journal of Computer Assisted Radiology and Surgery, vol. 5, pp. 183-193, 2010.

Hagn U., Nickl M., Jörg S., Passig G., Bahls T., Nothhelfer A., Hacker F., Le-Tien L., Albu-Schäffer A., Konietschke R., Grebenstein M., Warpup R., Haslinger R., Frommberger M., Hirzinger G. “The DLR MIRO—a versatile lightweight robot for surgical applications”. Industrial Robot, vol. 35(4), pp. 324-336, 2008.

Haidegger, T., Sándor, J. and Benyó, Z. “Surgery in space: the future of robotic telesurgery”. Surgical endoscopy, vol. 25(3), pp. 681-690, 2011.

Halim I., Tavakkolizadeh A. "NOTES: The next surgical revolution?". International Journal of Surgery, vol. 6, pp. 273-276, 2008.

Intuitive Surgical website FAQ: http://www.davincisurgery.com/da-vinci-surgery/frequently-asked-questions.php

Huang P., Gu L., Zhang J., Yu X. et al. “Virtual Surgery Deformable Modelling Employing GPU Based Computation”. 17th International Conference on Artificial Reality and Telexistence. Aalborg University Esbjerg, Denmark, pp. 221-227, 2007.

Jakel R., Schmidt-Rohr S.R., Losch M. and Dillmann R. “Representation and constrained planning of manipulation strategies in the context of programming by demonstration”. IEEE International Conference on Robotics and Automation (ICRA), pp. 162-169, 2010.

Jakopec, M., Rodriguez y Baena, F., Harris, S. J., Gomes, P., Cobb, J., & Davies, B. L. “The hands-on orthopaedic robot Acrobot: Early clinical trials of total knee replacement surgery”. IEEE Transactions on Robotics and Automation, vol. 19(5), pp. 902-911, 2003.

Kahnamoui K., Cadeddu M., Farrokhyar F., Anvari M., “Laparoscopic surgery for colon cancer: a systematic review”. Canadian Journal of Surgery, vol. 50, pp. 48-57, 2007.

Kazanzides, P., Fichtinger G., Hager G.D., Okamura A.M., Whitcomb L.L., and Taylor R.H. “Surgical and Interventional Robotics – Core Concepts, Technology, and Design”, IEEE Robotics & Automation Magazine, vol. 15(2), pp. 122-130, 2008.

King H., Low T., Hufford K., and Broderick T., “Acceleration compensation for vehicle based telesurgery on earth or in space,” IEEE/RSJ Int’l Conf. on Intelligent Robots and Systems (IROS), pp. 1459–1464, 2008.

Ko, S.Y., Lee, W.J. and Kwon D.S. “Intelligent interaction based on a surgery task model for a surgical assistant robot: Awareness of current surgical stages based on a surgical procedure model”. International Journal of Control, Automation and Systems, vol. 8(4), pp. 782-792, 2010.

Kroh, M., El-Hayek, K., Rosenblatt, S., Chand, B., Escobar, P., Kaouk, J. and Chalikonda, S. “First human surgery with a novel single-port robotic system: cholecystectomy using the da Vinci Single-Site platform”, Surgical endoscopy, vol. 25(11), pp. 3566-3573, 2011.

Krupa A., Morel G., Mathelin M. “Achieving high-precision laparoscopic manipulation through adaptive force control”, Advanced Robotics, vol. 18(9), pp. 905-926, 2004.

Kumar, R., et al. “Performance of robotic augmentation in microsurgery-scale motions”. 2nd Intl. Symp. On Medical Image Computing and Comp.-Assisted Surgery, pp. 1108-1115, 1999.

Kwok K.W., Tsoi K.H., Vitiello V., Clark J., Chow G.C., Luk W. and Yang G Z. “Dimensionality reduction in controlling articulated snake robot for endoscopy under dynamic active constraints”, IEEE Transactions on Robotics, vol. 29(1), pp. 15-31, 2013.

Lin H.C., Shafran I., Murphy T.E., Okamura A.M., Yuh D.D. and Hager G.D. “Automatic detection and segmentation of robot-assisted surgical motions”. In Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 802-810, 2005.

Lloyd S. “Least squares quantization in PCM”. IEEE Transactions on Information Theory, vol. 28(2), pp. 129-137, 1982.

Locke R., Patel, Rajni V. “Optimal remote center-of-motion location for robotics-assisted minimally-invasive surgery”. IEEE International Conference on Robotics and Automation (ICRA), pp. 1900-1905, 2007.

Lum M.J., Friedman D.C., Sankaranarayanan G., King H., Fodero K., Leuschke R., ... and Sinanan M.N. “The RAVEN: Design and validation of a telesurgery system”. The International Journal of Robotics Research, vol. 28(9), pp. 1183-1197, 2009.

Malik A. et al. “Endoluminal and transluminal surgery: current status and future possibilities”. Surgical Endoscopy, vol. 20(8), pp. 1179–1192, 2006.

Merchant A.M. et al. “Transumbilical Gelport Access Technique for Performing Single Incision Laparoscopic Surgery (SILS)”. Journal of Gastrointestinal Surgery, vol. 13(1), pp. 159-162, 2008.

Mettler, L., Ibrahim, M. and Jonat, W. “One year of experience working with the aid of a robotic assistant (the voice-controlled optic holder AESOP) in gynaecological endoscopic surgery”. Human Reproduction, vol. 13(10), pp. 2748-2750, 1998.

Michelin M., Poignet P., Dombre E. “Dynamic Task/Posture Decoupling for Minimally Invasive Surgery Motions: Simulation Results”, IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 3625-3630, 2004.

Michelin M., Poignet P., Dombre E., “Experimental Study of Dynamic Task/Posture Decoupling in Minimally Invasive Surgery Motions”, Experimental Robotics IX, Springer Tracts in Advanced Robotics, vol. 21, pp. 217-226, 2006.

Mitra, S. and Acharya, T. “Gesture recognition: A survey”. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 37(3), pp. 311-324, 2007.

Moreira P., Zemiti N., Liu C. and Poignet P. “Viscoelastic model based force control for soft tissue interaction and its application in physiological motion compensation”. Computer methods and programs in biomedicine, vol. 16(2), pp. 52-67, 2014.

Muñoz V.F., Garcia-Morales I., Perez del Pulgar C.J., Gomez-De-Gabriel J.M., Fernandez-Lozano J., Garcia-Cerezo, et al. “Control movement scheme based on manipulability concept for a surgical robotic assistant”. IEEE International Conference on Robotics and Automation (ICRA), pp. 245-250, 2006.

Muradore R., Bresolin D., Geretti L., Fiorini P. and Villa T. “Robot Surgery”, IEEE Robotics & Automation Magazine, vol. 18(3), pp. 24-32, 2011.

Niccolini M., Petroni G., Menciassi A., and Dario P. “Real-time control architecture of a novel Single-Port lapaRoscopy bimaNual roboT (SPRINT)”. IEEE International Conference on Robotics and Automation (ICRA), pp. 3395-3400, 2012.

Nuno, E., Rodríguez, A. and Basañez, L. “Force reflecting teleoperation via ipv6 protocol with geometric constraints haptic guidance”. In Advances in Telerobotics, pp. 445-458, 2007.

O’Malley M.K., Gupta A., Gen M., and Li, Y. “Shared control in haptic systems for performance enhancement and training”. Journal of Dynamic Systems, Measurement, and Control, vol. 128(1), pp. 75-85, 2005.

Oosterhout J., Wildenbeest J., Boessenkool H., Heemskerk C., de Baar M., van der Helm F., Abbink D. “Haptic Shared Control in Tele-manipulation: Effects of Inaccuracies in Guidance on Task Execution” IEEE Transactions on Haptics, in press, 2015.

Osa T., Harada K., Sugita N. and Mitsuishi M. “Trajectory planning under different initial conditions for surgical task automation by learning from demonstration”. IEEE International Conference on Robotics and Automation (ICRA), pp. 6507-6513, 2014.

Padoy, N. et al. “Human-Machine Collaborative surgery using learned models”. IEEE International Conference on Robotics and Automation (ICRA) pp. 5285–5292, 2011.

Park S., Howe R.D. and Torchiana D.F. “Virtual fixtures for robotic cardiac surgery”. In Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 1419-1420, 2001.

Petroni G., Niccolini M., Caccavaro S., Quaglia C., Menciassi A., Schostek S.and Dario P. “A novel robotic system for single-port laparoscopic surgery: preliminary experience”. Surgical endoscopy, vol. 27(6), pp. 1932-1937, 2013.

Perino, A., et al. “Total laparoscopic hysterectomy versus total abdominal hysterectomy: an assessment of the learning curve in a prospective randomized study”. Human Reproduction, vol. 14(12), p. 2996-2999, 1999.

Piccigallo M., Scarfogliero U., Quaglia C., Petroni G., Valdastri P., Menciassi A. and Dario P. “Design of a novel bimanual robotic system for single-port laparoscopy”. IEEE/ASME Transactions on Mechatronics, vol. 15(6), pp. 871-878, 2010.

Puangmali P., Althoefer K., Seneviratne L. D., Murphy D. and Dasgupta P. "State-of-the-art in force and tactile sensing for minimally invasive surgery", IEEE Sensors Journal, vol. 8(4), pp. 371-381, 2008.

Rabiner L.R. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE, vol. 77, pp. 257-286, 1989.

Rabiner L.R., Levinson S.E. and Sondhi M.M. "On the application of vector quantization and hidden Markov models to speaker-independent, isolated word recognition". The Bell System Technical Journal, vol. 62(4), pp. 1075-1105, 1983.

Reza M.M., Blasco J.A., Andradas E., Cantero R., Mayol J. "Systematic review of laparoscopic versus open surgery for colorectal cancer". The British journal of surgery. Vol. 93, pp. 921-928, 2006.

Richards C., Rosen J., Hannaford B., Pellegrini C. and Sinanan M. "Skills evaluation in minimally invasive surgery using force/torque signatures". Surgical endoscopy, vol. 14(9), pp. 791-798, 2000.

Rivas-Blanco I., Estebanez B., Cuevas-Rodriguez M., Bauzano E. and Munoz V.F. "Towards a cognitive camera robotic assistant", 5th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics, pp. 739-744, 2014.

Ryden F. and Chizeck H.J. "Forbidden-region virtual fixtures from streaming point clouds: Remotely touching and protecting a beating heart". IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3308-3313, 2012.

Reichenspurner H. et al. "Use of the Voice Controlled and Computerassisted Surgical System Zeus for Endoscopic Coronary Artery Surgery Bypass Grafting" Journal of Thoracic Cardiovascular Surgery, vol. 118, pp. 11-16, 1999.

Richards C. et al. "Skills evaluation in minimally invasive surgery using force/torque signatures". Surgical Endoscopy, vol. 14(9), pp. 791-798, 2000.

Rodriguez and Chitwood. "Robotics in Cardiac Surgery". Scandinavian Journal of Surgery, vol. 98, pp. 120-124, 2009.

Rosen J., Hannaford B., Richards C.G. and Sinanan M.N. "Markov modeling of minimally invasive surgery based on tool/tissue interaction and force/torque signatures for evaluating surgical skills". IEEE Transactions on Biomedical Engineering, vol. 48(5), pp. 579-591, 2001.

Rozo L., Jiménez P. and Torras C. “A robot learning from demonstration framework to perform force-based manipulation tasks”. Intelligent Service Robotics, vol. 6(1), pp. 33-51, 2013.

Rozo L., Calinon S. and Caldwell D.G. “Learning force and position constraints in human-robot cooperative transportation”. The 23rd IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 619-624, 2014.

Ruiz Morales and Correcher, “Force estimation for a minimally invasive robotic surgery system”, European Patent, publication number: EP2491884 A1, 2012.

Sakoe H. and Chiba S. “Dynamic programming algorithm optimization for spoken word recognition,” IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 26, pp. 43–49, 1978.

Schlachta C.M., Mamazza J., Seshadri P.A. Cadeddy M., Gregoire R., Poulin E.C. “Defining a learning curve for laparoscopic colorectal resection”. Diseases of the colon & rectum, vol. 44(2), pp. 217-22, 2001.

Schwarz G. “Estimating the dimension of a model”. The annals of statistics, vol. 6(2), pp. 461-464, 1978.

Segre A.M. “Machine learning of robot assembly plans”. Kluwer Academic Publishers, 1988.

Seibold U., Kuebler B., Hirzinger G. “Prototypic force feedback instrument for minimally invasive robotic surgery”. In: Bozovic V (ed) Medical robotics. I-Tech Education and Publishing, pp. 377–400, 2008.

Senapati S. and Advincula A.P. “Telemedicine and robotics: paving the way to the globalization of surgery”. International Journal of Gynecology & Obstetrics, vol. 91(3), pp. 210-216, 2005.

Shang J., Payne C.J., Clark J., Noonan D.P., Kwok K.W., Darzi, A. and Yang G.Z. “Design of a multitasking robotic platform with flexible arms and articulated head for minimally invasive surgery”. International Conference on Intelligent Robots and Systems (IROS), pp. 1988-1993, 2012.

Shon A.P., Grochow K. and Rao R.P. “Robotic imitation from human motion capture using gaussian processes”. 5th IEEE-RAS International Conference on Humanoid Robots, pp. 129-134, 2005.

N. Shussman, A. Schlager, et al. “Single-incision laparoscopic cholecystectomy: lesson learned for success”, Surgical Endoscopy, vol. 25, pp. 404-407, 2011.

Siciliano B., Sciavicco L., “Robotics: Modelling, Planning and Control”, Advanced Textbooks in Control and Signal Processing Series, Springer-Verlag, 2010.

Taylor R. and Stoianovici D. “Medical Robotics in Computer-Integrated Surgery”, IEEE Transactions on Robotics and Automation, vol. 19(5), pp. 765-781, 2003.

Taylor R.H. et al. “A Telerobotic Assistant for Laparoscopic Surgery”, IEEE Engineering Medical Biology Magazine, pp. 279-287, 1995.

Tholey G., Desai J.P. and Castellanos A.E. “Force feedback plays a significant role in minimally invasive surgery: results and analysis”. Annals of surgery, vol. 241(1), p. 102, 2005.

Titan Medical Inc. website: http://www.titanmedicalinc.com/product/

Tobergte A. et al. "The sigma. 7 haptic interface for MiroSurge: A new bi-manual surgical console" 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2023-3030, 2011.

Unger B.J., Nicolaidis A., Berkelman P.J., Thompson A., Klatzky R.L. and Hollis R.L. “Comparison of 3-D haptic peg-in-hole tasks in real and virtual environments”. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), vol. 3, pp. 1751-1756, 2001.

Viterbi A.J. “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm”. IEEE Transactions on Information Theory, vol. 13(2), pp. 260-269, 1967.

Weede O., Dittrich F., Worn H., “Workflow analysis and surgical phase recognition in minimally invasive surgery”, Proc. Of the 2012 IEEE Int. Conf. on Robotics and Biomimentics, pp. 1068-1074, 2012.

Xu K., Goldman R.E., Ding J., Allen P.K., Fowler D.L. and Simaan N. "System Design of an Insertable Robotic Effector Platform for Single Port Access (SPA) Surgery". IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5546-5552, 2009.

Resumen de la Tesis Doctoral

Navegación Inteligente en Robótica Quirúrgica

En las últimas dos décadas, los procedimientos quirúrgicos han evolucionado desde el uso de técnicas como la laparotomía (cirugía abierta) hacia el uso de la cirugía mínimamente invasiva siguiendo la hipótesis de que reduciendo el número y tamaño de las incisiones realizadas sobre el paciente, se reduciría el tiempo de recuperación y las complicaciones posoperatorias. La cirugía laparoscópica se basa en el empleo de instrumentos especiales y una cámara, los cuales son introducidos dentro de la pared abdominal mediante, al menos, tres incisiones pequeñas, permitiendo esto que el cirujano pueda manipular los órganos internos del paciente. Sin embargo, la tendencia actual se mueve hacia el desarrollo de nuevos métodos. Uno de ellos es conocido como cirugía de un solo puerto (SPAS en inglés), éste es un procedimiento quirúrgico en el que se realiza una única incisión, a través de la cual todos los instrumentos y la cámara son introducidos mediante el empleo de un trocar multipuerto. De esta manera, la reducción del número de incisiones proporciona ciertos beneficios, tales como: mejores resultados cosméticos, reducción del dolor posoperatorio así como un acortamiento de la estancia hospitalaria. A pesar de estas ventajas, esta nueva técnica proporciona algunos inconvenientes para los cirujanos ya que los instrumentos están introducidos a través del mismo trocar y estos pueden colisionar dentro y fuera del abdomen. Además, la proximidad de los instrumentos a la cámara endoscópica, introducido también por el mismo trocar, provoca una pérdida de triangulación y una reducción del campo de visión dentro del abdomen. Estos inconvenientes implican una restricción de los movimientos que el cirujano puede realizar con

los instrumentos, lo cual requiere una mayor habilidad del cirujano. En este sentido, las plataformas robóticas teleoperadas surgen como una herramienta útil para los cirujanos, en las que un sistema de navegación inteligente puede ser integrado, mejorando la habilidad del cirujano mediante el empleo de un sistema de guiado que lo asista durante la realización de una tarea quirúrgica. Además, el sistema de navegación podría ser capaz de reconocer los gestos que realiza el cirujano, realizar movimientos teniendo en cuentas las restricciones de SPAS, y guiar de forma correcta al cirujano durante la realización del gesto reconocido. Estas características requieren un conocimiento previo del procedimiento quirúrgico, y por lo tanto, se necesita un método que sea capaz de aprender este conocimiento.

Para alcanzar este sistema de navegación inteligente, varios ámbitos tecnológicos debe ser extendidos e integrados:

Navegación

Tal y como se ha comentado previamente, las propias restricciones de SPAS deberán ser consideradas cuando los instrumentos son manejados por manipuladores. La restricción más importante es el punto de fulcro, el cuál es el punto por el que los instrumentos se introducen dentro del abdomen del paciente. Esta restricción es comúnmente resuelta mediante el empleo de un centro de rotación remoto (RCM en inglés) el cuál se hace coincidir con el punto de fulcro. El RCM reduce el número de grados de libertad de movimiento del instrumento al cuatro: tres rotacionales y uno longitudinal a lo largo del instrumento. La plataforma robótica da Vinci y otras usan un RCM mecánico basado en una configuración cinemática específica que es ajustada antes de realizar la cirugía, y permanece fijo durante toda la operación. Esta solución se emplea principalmente en aplicaciones clínicas debido a su robustez y seguridad, un fallo en el controlador del robot no dañaría al paciente. Sin embargo, esta solución requiere una estructura cinemática específica, voluminosa y compleja, la cual no resulta útil para SPAS debido principalmente a su volumen. Por ejemplo, el sistema da Vinci emplea cuatro brazos que están adheridos a una estructura principal con una configuración cinemática que desarrolla un centro de rotación remoto basado en un paralelogramo. Aunque este sistema ha sido empleado en SPAS, existen grandes limitaciones en su empleo debido a su volumen, el rango de movimientos dentro del paciente es muy reducido y los brazos pueden colisionar durante el movimiento. Por otro lado, sistemas robóticos más ligeros, basados en configuraciones cinemáticas genéricas, usan un RCM virtual o por software, en el que el punto de fulcro es estimado en línea y puede cambiar durante la cirugía. Esta solución es útil cuando el punto de fulcro se mueve, como lo hace la pared abdominal durante la respiración. No obstante, cuando se emplea un RCM virtual, la ubicación del punto de fulcro debe ser conocida. Para solucionar esto, el uso de un estimador en línea que use la información proporcionada por sensores que midan la fuerza ejercida a lo largo de un instrumento quirúrgico aparece como una solución. La principal ventaja de este método es que este sensor se puede emplear para diferentes propósitos: teleoperación bilateral, estimación del punto de fulcro y/o control de esfuerzos. En este sentido, varios autores han realizado un extenso estudio del empleo de sensores de esfuerzos para estimar el punto de fulcro mediante el empleo de varias estrategias. Sin embargo, ninguna de estas contribuciones considera la interacción del instrumento con tejido blando dentro del abdomen. Este efecto debe ser considerado ya que lleva a errores en la estimación del punto de fulcro, y consecuentemente, a la aplicación de fuerzas no deseadas en el abdomen del paciente debido a este error. Además, ninguno de estos estudios ha sido adaptado para la técnica SPAS, en la que los instrumentos están introducidos por el mismo trocar, lo cual podría ser empleado para mejorar la estimación del punto de fulcro a través de la información redundante que se obtendría al estimar el punto de fulcro a través de las fuerzas ejercidas en cada instrumento.

Interacción Humano-Robot

Durante la teleoperación, el cirujano lleva a cabo tareas mediante el control de una plataforma robótica. Para poder realizar estas tareas de forma eficiente, el cirujano necesita recibir información sensorial de los robots. Teniendo esto en cuenta, estas plataformas robóticas se pueden clasificar así:

- Teleoperación directa: este es el sistema de teleoperación básico. El cirujano proporciona los movimientos a la plataforma manejando un dispositivo maestro durante la cirugía, y el robot (dispositivo esclavo) reproduce estos movimientos en tiempo real. Generalmente, el cirujano teleopera el robot mediante una consola que contiene los dispositivos maestros que son usados para mover los manipuladores y una o varias pantallas que proporcionan información visual. Este tipo de teleoperación ha evolucionado hacia lo denominado *telecirugía*, en la que un cirujano puede llevar cabo una operación sin tener que estar físicamente presente. Esta técnica no sólo engloba una consola maestra y robots esclavos sino que necesita de tecnologías de comunicaciones e información.
- Teleoperación bilateral: ésta es una evolución de la teleoperación directa. En este caso, el cirujano también es capaz de sentir las fuerzas que son ejercidas por los instrumentos durante la cirugía. Así, se colocan sensores de esfuerzos en cada robot esclavo, y se emplean dispositivos hápticos que proporcionan realimentación de esfuerzos al cirujano. Estos sensores de esfuerzos se pueden clasificar dependiente de proximidad a la punta del instrumento y su sensibilidad. El primer tipo engloba los sensores de esfuerzos que son colocados en la muñeca del manipulador, es decir, entre el efector final del manipulador y el instrumento

quirúrgico. Este tipo de sensores pueden medir las fuerzas y pares ejercidos a lo largo del instrumento y no necesitan ser esterilizados ya que no son introducidos en el abdomen. Sin embargo, estos sensores no proporcionan suficiente sensibilidad para diferenciar entre diferentes tipos de tejidos e incluso para diferenciar si las fuerzas obtenidas provienen de la punta del instrumento o de la interacción de éste con el abdomen. Por otro lado, el segundo tipo de sensores lo conforman aquellos que son instalados en la punta del instrumento, usados para sentir el tejido blando dentro del abdomen, haciendo que el cirujano sienta las mismas fuerzas que si realizara una laparotomía. Sin embargo, hay dos inconvenientes muy importantes cuando se usa este tipo de sensores: es necesario que estos se puedan esterilizar ya que se introducen dentro del abdomen, y por otro lado, cambios en la temperatura del sensor afectan las medidas de fuerza cuando se emplean transductores.

Como se ha descrito, estos tipos de teleoperación permiten la reproducción de los movimientos que realiza el cirujano con sus manos cuando éste usa instrumentos. Sin embargo, sería muy útil un método que proporcionara un apoyo adicional al cirujano. En este sentido, el guiado háptico es una evolución de la teleoperación bilateral. Esto se emplea para proporcionar fuerzas de guiado que permitan limitar el espacio del trabajo del cirujano o asistirlo durante la ejecución de una trayectoria predefinida que resuelva una tarea concreta. Esta técnica es comúnmente llamada *fijaciones virtuales* (*virtual fixtures* en inglés). Se ha demostrado que mediante el empleo de esta técnica se mejora la precisión y seguridad de la cirugía, y se reduce el tiempo necesario para realizar tareas quirúrgicas ya que combina la precisión de un robot con la inteligencia del ser humano. El uso de fijaciones virtuales se basa en la premisa de que las trayectorias de referencia y las regiones prohibidas son previamente conocidas, normalmente mediante el procesado de imágenes médicas o marcas realizadas en el paciente. Sin embargo, hay tareas en las que estas posiciones pueden verse afectadas por errores de estimación o no pueden ser estimadas. Por lo tanto, se necesitan nuevos métodos que generen guiado háptico.

Aprendizaje Computacional

El aprendizaje computacional ha sido ampliamente empleado en robótica. Esto se ha empleado en robótica quirúrgica para codificar movimientos realizados por el cirujano en un modelo que posteriormente es empleado para reconocer estos movimientos o proporcionar trayectorias que han sido previamente entrenadas.

En el caso del reconocimiento de movimientos, el aprendizaje computacional se ha empleado para detectar y clasificar movimientos que son realizados por cirujano, de forma que se pueda identificar la fase del procedimiento quirúrgico. Además, esto se ha empleado para implementar colaboración entre robots y cirujanos, en el que un manipulador es capaz de asistir al cirujano dependiente de la fase del procedimiento quirúrgico en el que éste se encuentre. Sin embargo, el uso de estos algoritmos para detectar si los instrumentos interaccionan con tejido blando dentro del abdomen no ha sido investigado aún. Esto sería muy útil para la estimación del punto de fulcro, ya que ayudaría a determinar si el instrumento está interaccionando con tejido blando dentro del abdomen, lo que significa que no se podría estimar el punto de fulcro empleando las fuerzas medidas a lo largo del instrumento.

El aprendizaje computacional también ha sido ampliamente empleado para generar trayectorias continuas en el tiempo que se basan en posiciones del manipulador o medidas de fuerza realizadas por éste. Así, el método de aprendizaje por demostración (LfD) ha permitido que los robots puedan llevar a cabo tareas que han sido previamente entrenadas por un humano. Por ejemplo, se ha podido llenar un vaso de agua, golpear pelotas de ping pong, alimentar una muñeca o situar una bola en un agujero dentro de una caja. Sin embargo, aún no se ha estudiado en profundidad el uso de LfD para guiado háptico. La principal dificultad está en generar las fuerzas de guiado apropiadas para ayudar a resolver una tarea en tiempo real, teniendo en cuenta la interacción del operador con el sistema, es decir, las fuerzas de guiado háptico no deberían llevar al operador a seguir una trayectoria temporal, sino que debería proporcionar un guiado que dependa de los movimientos que realiza el operador.

Estas tecnologías, una vez extendidas e integradas, podrían ayudar a resolver los problemas inherentes de SPAS, quedando todo esto integrado en un sistema de navegación inteligente que asista a los cirujanos durante la ejecución de procedimientos quirúrgicos basados en SPAS.

Contribuciones

Esta tesis proporciona resultados teóricos y experimentales relacionados con la navegación inteligente de instrumentos quirúrgicos que son manejados por robots en técnicas SPAS. En este sentido, la navegación del instrumento tiene que considerar las restricciones inherentes de SPAS para evitar fuerzas indeseables en el paciente y para llevar a cabo los movimientos del instrumento de una forma precisa. Además, el empleo de guiado háptico para asistir al cirujano durante la teleoperación aparece como un método prometedor que aún no ha sido investigado. Por lo tanto, esta tesis propone un método de navegación inteligente que integra una solución a cada uno de estos problemas dentro de una arquitectura global. Por lo tanto, las principales contribuciones de esta tesis son:

- **Algoritmo de control de posición para manejar instrumentos quirúrgicos en SPAS**

Se propone un método de navegación basado en un control paralelo de fuerza-posición que ha sido adaptado a SPAS. Este método usa un control paralelo de fuerza-posición para levar a cabo los movimientos del instrumental teniendo en cuenta la restricción del punto de fulcro y minimizando las fuerzas ejercidas en la pared abdominal del paciente. Además, aprovechando SPAS, lo que significa que los instrumentos están introducidos por el mismo trocar, el punto de fulcro es estimado empleando un detector de interación con tejido blando y un método de fusión de medidas.

- **Detector de interacción con tejido blando**

Se necesita conocer cuando los instrumentos interaccionan con tejido blando dentro del abdomen para reducir errores en la estimación del punto de fulcro. Por lo tanto, se propone el uso de algoritmos basados en aprendizaje computacional para aprender los movimientos del cirujano y detectar esta interacción. Para ello, cada movimiento es dividido en una serie de estados que proporcionan información sobre la interacción del instrumento con el tejido blando. Así, el propósito de este algoritmo es reconocer los movimientos del cirujano y el estado en el que éste se encuentra, de forma que se pueda conocer si existe interacción con tejido blando en cada instante de tiempo.

- **Método de guiado háptico basado en aprendizaje por demostración**

Para extender la habilidad del cirujano durante la teleoperación, se ha definido un método génerico para guiado háptico que se basa en aprendizaje por demostración. Este método tiene en cuenta los movimientos que han sido previamente entrenados junto con su relación con la información sensorial obtenida durante el movimiento. Usando esta información se generan fuerzas de guiado háptico que son transmitidas al cirujano con el fin de asistirlo durante la teleoperación.

- **Implementación del método de navegación inteligente propuesto y sus resultados experimentales**

Los métodos propuestos han sido implantados en dos plataformas robótics de teleoperación. Para validar experimentalmente el método de navegación inteligente para SPAS se empleó la plataforma CISOBOT, la cual ha sido desarrollada por la Universidad de Málaga. Los resultados experimentales obtenidos han demostrado el comportamiento del método propuesto. Por otro lado, el método de guiado háptico fue implantado en la plataforma “LWR Taskboard” desarrollada por el laboratorio de háptica y telerrobótica de la Agencia Espacial

Europea. Esta plataforma se empleó para resolver la inserción de una clavija (peg-in-hole en inglés) y así validar el método propuesto de guiado háptico.

Contexto y Motivación

Esta tesis se encuentra enmarcada dentro de los trabajos de investigación en robótica quirúrgica que se llevan a cabo en el grupo de investigación en Ingeniería de Sistemas y Automática de la Universidad de Málaga. Además, durante la realización de esta tesis doctoral se realizó una estancia de tres meses en el laboratorio telerrobótica y háptica de la Agencia Espacial Europea en la que se realizaron trabajos de investigación relacionados con el guiado háptico en teleoperación espacial.

Los resultados de esta tesis extienden estudio previos relacionados con la teleopeación de plataforma robóticas quirúrgicas y la detección de gestos realizados por cirujanos. Estos estudios formaron parte de las actividades de investigación que fueron llevadas a cabo en la Universidad de Málaga, cuyo principal logro fue el diseño e implantación de un asistente robótico que manejaba la cámara laparoscópica, el cual fue exitosamente usado en cirugías con humanos. Actualmente, las actividades de este equipo de investigación se centran en el desarrollo de tecnologías perceptuales, de navegación y cognitivas con aplicación en robótica quirúrgica.

Esta tesis ha sido financiada por el Gobierno de España a través de los proyectos DPI2010-21126-C03-01 y DPI2013-47196-C3-1R. El principal objetivo de estos proyectos ha sido el dotar de autonomía a plataformas robóticas para cirugía mediante el empleo de algoritmos cognitivos y de aprendizaje computacional.

Por otro lado, el laboratorio de háptica y telerrobótica de la Agencia Espacial Europea está implicado en varios proyectos relacionados con la teleoperación de robots mediante el empleo de dispositivos hápticos. Uno de estos es el proyecto METERON, el cual consiste en el desarrollo de tecnologías para la teleoperación de robots desde el espacio. Durante la estancia realizada, se desarrolló un algoritmo para el guiado háptico el cual fue empleado para resolver el problema de inserción de una clavija.

Estructura de la Tesis

Esta tesis está dividida en seis capítulos, seis apéndices y las referencias bibliográficas. Cada capítulo, excepto el primero, comienza con una introducción que presenta el problema que se va a resolver, seguido de la estructura propia de cada capítulo, y finaliza con una serie de conclusiones que resaltan las contribuciones del capítulo así como los resultados obtenidos.

La estructura de esta tesis está enfocada en la solución de diferentes problemas que surgen cuando se emplea una plataforma robótica teleoperada en SPAS: un método de navegación que considera la restricción del punto de fulcro como un centro de rotación remoto virtual, un detector de interacción con tejido blando basado en aprendizaje computacional, empleado para mejorar la estimación del punto de fulcro, y un sistema de guiado háptico, basado en aprendizaje por demostración, que es capaz de asistir al cirujano durante la teleoperación del sistema robótico. Así, el capítulo 2 introduce el pasado y presente de la teleoperación quirúrgica mediante robots, proporcionando una breve descripción de las plataformas robóticas quirúrgicas más importantes. En este mismo capítulo se presenta el estado del arte del uso de aprendizaje computacional en robótica quirúrgica. Esta metodología ha sido empleada para resolver dos problemas en esta tesis: el uso de modelos ocultos de Markov para reconocimiento del gesto y el uso de aprendizaje por demostración para la generación de movimientos del robot que se basan en un entrenamiento previo. Finalmente, este capítulo describe el uso que se ha realizado del guiado háptico en robótica quirúrgica.

El capítulo 3, titulado "Smart Navigation for Single Port Access Surgery", propone un método de navegación para el uso de robots en cirugía de un solo puerto, el cual es capaz de mover dos instrumentos quirúrgicos que se han introducido a través de trocar multipuerto teniendo en cuenta la restricción del punto de fulcro, el cual es estimado mediante la lectura de las fuerzas ejercidas a lo largo del instrumento. Para mejorar la estimación del punto de fulcro, el método de navegación propuesto emplea un método de fusión de medidas que se aprovecha del efecto de tener los dos instrumentos introducidos a través del mismo trocar. Sin embargo, el empleo de estas fuerzas para estimar el punto de fulcro requiere que el instrumento únicamente interaccione con el abdomen del paciente, por lo tanto, es necesario el empleo de un método que sea capaz de detectar cuando la punta del instrumento interacciona con tejido blando. Este método se describe también en este capítulo. Finalmente, se describe el esquema de control paralelo de fuerza-posición que ha sido empleado para minimizar las fuerzas ejercidas en el abdomen del paciente.

Una vez definido el método de navegación para teleoperación en SPAS, el capítulo 4 propone el uso de aprendizaje por demostración para el guiado háptico, presentando en este capítulo un método para resolver tareas genéricas. En concreto, este método se divide en dos fases. La primera consiste en dividir las tareas en gestos y entrenar cada uno de ellos mediante demostraciones, y la segunda fase hace uso de estos entrenamientos para asistir al operador durante la reproducción de la tarea.

El capítulo 5 “Implementation and Experiments” descibe los experimentos que han sido llevados a cabo para validar el sistema de navegación inteligente para SPAS propuesto. El método de navegación propuesto fue probado mediante el análisis de las fuerzas ejercidas en el abdomen, los errores de estimación en el punto de fulcro y el retraso que se obtuvo en la detección de la interacción con el tejido blando. El método de guiado háptico basado en aprendizaje por demostración fue validado empleando la tarea de inserción de clavija, cuyos resultados experimentales demostraron el funcionamiento de las fases de entrenamiento y reproducción.

El capítulo 6 resalta las contribuciones más relevantes de esta tesis y propone una serie de trabajos futuros. Finalmente, los apéndices proporcionan al lector el análisis de estabilidad del esquema de control planteado en el capítulo 3 y la teoría de los modelos matemáticos que han sido empleados en los diferentes capítulos.

Conclusiones

La cirugía minimamente invasiva ha supuesto un gran logro con respecto a la cirugía tradicional. A pesar de que hay varias técnicas en cirugía minimamente invasiva, la cirugía de único puerto es una de las técnicas de cirugía minimamente invasiva más novedosas. Esta técnica consiste en introducir todos los instrumentos, así como la cámara laparoscópica, a través de un trocar multipuerto, con lo que se reduce el número de cicatrices. De esta manera se reduce la estancia hospitalaria y se mejoran los resultados cosméticos. Sin embargo, esta técnica conlleva varios inconvenientes para los cirujanos. En este sentido, el uso de plataformas robóticas teleoperadas emergen como una ventaja para poder realizar este tipo de técnicas. El uso de estas plataformas ayudaría a los cirujanos a manejar los instrumentos quirúrgicos evitando los problemas inherentes a este tipo de técnicas. Uno de los problemas que surgen cuando se emplean estas plataformas robóticas está relacionado con el hecho de que los movimientos están limitados por estar los instrumentos introducidos a través de un trocar. Algunas plataformas robóticas como el sistema da Vinci resuelve este problema mediante el empleo de estructuras cinemáticas voluminosas, las cuales implementan un centro de rotación remoto mecánico que se hace coincidir con el punto de fulcro. Otros autores han resuelto este problema usando un centro de rotación remoto virtual el cual se ha estimado mediante el empleo de las fuerzas de interacción del instrumento con el abdomen. Este método puede ser implementado en robots más ligeros y con configuraciones cinemáticas genéricas, útiles para realizar cirugía de un solo puerto ya que ayudarían a reducir las colisiones entre los diferentes brazos robóticos.

En este sentido, esta tesis propone un método de navegación inteligente para cirugía de un solo puerto, el cual emplea las fuerzas ejercidas a lo largo del instrumento quirúrgico para estimar el punto de fulcro, mejorando su precisión mediante el empleo de un algoritmo de fusión de medidas que se aprovecha del efecto de tener dos instrumentos introducidos por el mismo trocar. Este método de navegación ha sido implantado en la plataforma CISOBOT, la cual ha sido desarrollada por la Universidad de Málaga y cuyos resultados experimentales han demostrado que este método mejora la estimación del punto de fulcro. Así, el error de estimación del punto de fulcro se redujo cuando se empleó el método de fusión de medidas. La estimación del punto de fulcro fue usada para llevar a cabo movimientos de los instrumentos mediante el empleo de un esquema de control paralelo de fuerza-posición basado en la función Jacobiana de los manipuladores empleados, el cual emplea el punto de fulcro estimado como un centro de rotación remoto virtual y además reduce las fuerzas ejercidas en el abdomen del paciente. Como se ha demostrado en los resultados experimentales, el error máximo obtenido durante la reproducción de varias trayectorias fue bastante bajo tanto en trayectorias predefinidas como las proporcionadas por un operador mediante la teleoperación del sistema robótico. Además, las fuerzas abdominales que se ejercieron en el abdomen quedaron reducidas mediante el empleo del esquema de control propuesto.

Como se explica en la sección 3.3, aparecen dos fuerzas de interacción a lo largo del instrumento. La primera de ellas es la fuerza que es ejercida en el instrumento debido a su interacción con el abdomen del paciente, la cual se puede emplear para estimar el punto de fulcro. La segunda interacción se produce cuando la punta del instrumento interacciona con tejido blando dentro del abdomen, la cual se puede emplear para proporcionar realimentación háptica al cirujano. Sin embargo, la estimación del punto de fulcro sólo se puede realizar cuando el instrumento no interacciona con tejido blando dentro del abdomen. Por este motivo, se ha desarrollado un detector de interacción con tejido blando basado en modelos ocultos de Markov, el cual ha sido entrenado para detectar cuatro gestos genéricos: tirar, empujar, levantar y hundir. Cada uno de estos gestos se ha dividido en cinco estados, los cuales proporcionan información acerca de la interacción del instrumento con tejido blando dentro del abdomen. Este método primero detecta el gesto que se está realizando y después estima el estado que se corresponde con ese gesto, proporcionando información sobre la interacción con tejido blando (Figura 3-17). Para validar este método se realizaron dos experimentos. El primero de ellos fue empleado para validar el reconocimiento del gesto, en el que se reconocieron los cuatro gestos definidos exceptuando el gesto “tirar” como se muestra en la Tabla 5.2. El segundo experimento consistió en la comprobación de la estimación del estado una vez el gesto ha sido correctamente reconocido. Para esto, fue medido el retraso entre el instante en el que la interacción con tejido blando ocurre y ésta es detectada, así como el instante en el que la interacción termina y es detectada (Figura 3-15). Como muestra la Figura 5-16, el retraso medio es aceptable para llevar a cabo la estimación del punto de fulcro sin errores.

Una vez definido el método de navegación para cirugía de un solo puerto, la habilidad del cirujano se podría mejorar mediante el empleo de un sistema de guiado háptico que asista al cirujano durante la teleoperación de un sistema robótica quirúgico. En este sentido, en el capítulo 4 se ha propuesto un método que proporcione guiado háptico basado en aprendizaje por demostración. El objetivo de este método es usar gestos que han sido entrenados previamente por un cirujano experto para proporcionar fuerzas de guiado basadas en la información sensorial que proporciona el sistema robótico. Para esto, se emplearon modelos de mezclas Gausianas para codificar la relación entre la información sensorial y las fueras de guiado ideales, obtenidas durante la fase de entrenamiento. Una vez que cada gesto fue codificado en un modelo de mezclas Gausianas, se empleó una regresión de mezclas Gausianas para generar las fuerzas de guiado durante la reproducción del gesto. El método propuesto ha sido empleado para resolver la tarea de inserción de clavija (peg-in-hole), la cual ha demostrado ser una tarea compleja cuando se lleva a cabo usando un sistema robótico teleoperado. De la misma manera que este método se ha empleado para esa tarea, se podría emplear para resolver tareas quirúrgicas tales como la inserción de un instrumento por un trocar, sostener la vesícula con una fuerza que ha sido previamente entrenada, introducir instrumentos por orificios, etc.

Un estudio preliminar sobre el empleo del método propuesto de guiado háptico fue realizado con la plataforma LWR Taskboar Workcell del laboratorio de háptica y telerrobótica de la Agencia Espacial Europea, resolviendo la tarea de inserción de clavija (peg-in-hole). Esta tarea se dividió en dos gestos (Figura 5-17): gesto de contacto con la superficie, que se empleó para colocar la clavija en el agujero; y el gesto de efecto palanca, el cual se empleó para alinear la clavija con el agujero. Cada gesto fue entrenado mediante demostraciones kinestésicas comenzando desde diferentes posiciones iniciales (Figura 5-18), y estos fueron codificados en dos modelos de mezclas Gausianas. Una vez realizado esto, se llevaron a cabo más demostraciones con el fin de validar el error entre estas demostraciones y las fuerzas de guiado háptico obtenidas a través de la regresión de mezclas Gausianas. Como se muestra en la Figura 5-21, el error fue reducido y por lo tanto la fase de entrenamiento fue correcta. Finalmente, se realizaron varias inserciones de clavijas usando el sistema robótico teleoperado. Los resultados obtenidos (Figura 5-22) demostraton que la clavija se introduce con mayor facilidad cuando se emplea guiado háptico.

En resumen, esta tesis propone un método de navegación inteligente para cirugía de único puerto que tiene en cuenta la restricción del punto de fulcro, el cual es estimado mediante el uso de las medidas de fuerzas y pares que se ejercen a lo largo del instrumento debido a su interacción con el abdomen, las cuales pueden ser leídas mediante un sensor de esfuerzos. Debido a que el instrumento también puede interaccionar con tejido blando, se ha incluido en el método de navegación un detector de interacción con tejido blando que permite detectar esa situación. Además, la habilidad del cirujano se ha mejorado mediante el empleo de un método de guiado háptico basado en aprendizaje por demostración con el cual se ha resuelto una tarea genérica y a la vez compleja que es la inserción de una clavija, la cual es similar a otras tareas quirúrgicas.

This thesis is focused on the navigation of surgical instruments by teleoperated surgical robotic platforms in Single Port Access Surgery. In particular, the proposed navigation method is based on a virtual Remote Centre of Motion, which coincides with the insertion point (fulcrum point) that is estimated using abdominal interaction forces along the surgical instruments. Because these instruments also interact with the soft tissue inside the abdomen, which affects the fulcrum point estimation, a method is needed to determine whether the instrument tip interacts with the soft tissue inside the abdomen. To this end, we have used a soft tissue interaction detector based on a Hidden Markov Model. Furthermore, this thesis proposes the use of haptic guidance to improve the surgeon's experience when using teleoperated robotic platforms. Thus, Learning from Demonstration is proposed to generate guidance force references that guide the surgeon during the reproduction of a task.